ECCM^2

Proceedings of the Second European Conference on Cognitive Modelling

Nottingham, UK

1st - 4th April 1998

Proceedings of the Second European Conference on Cognitive Modelling (ECCM-98)

Nottingham, UK

1st - 4th April 1998

Edited by

Frank E Ritter and Richard M Young

NOTTINGHAM
University Press

Nottingham University Press
Manor Farm, Main Street, Thrumpton
Nottingham, NG11 0AX, United Kingdom

NOTTINGHAM

First published 1998

British Library Cataloguing in Publication Data
Proceedings of the Second European Conference on
Cognitive Modelling (ECCM-98)

ISBN 1-897676-67-0

Printed and bound by Quorn Litho, Loughborough, Leics.

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)

2. REPORT DATE

4 April 1998

3. REPORT TYPE AND DATES COVERED

Conference Proceedings

4. TITLE AND SUBTITLE

Poceedings of the Secnd European Conference on Cognitive Modelling

5. FUNDING NUMBERS

F61775-98-WE003

6. AUTHOR(S)

Ritter, Frank E, and Richard M. Young, Ed.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Department of Psychology
University of Nottingham
Nottingham NG7 2RD
United Kingdom

8. PERFORMING ORGANIZATION REPORT NUMBER

N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

EOARD
PSC 802 BOX 14
FPO 09499-0200

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

CSP 98-1046

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

A

13. ABSTRACT (Maximum 200 words)

The Final Proceedings for 2nd European Conference on Cognitive Modeling, 1 April 1998 - 4 April 1998

This interdisciplinary conference covered all areas of cognitive modeling, including artificial intelligence programming; classification; problem solving; reasoning; inference; learning; language processing; human-computer interaction; symbolic and connectionist models; evolutionary computation; artificial neural networks; grammatical inferences; reinforcement learning; and data sets designed to test models.

14. SUBJECT TERMS

Psychology, Computers, Human Factors, Intelligent Tutoring

15. NUMBER OF PAGES

213

16. PRICE CODE

N/A

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19, SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Second European Conference on Cognitive Modelling (ECCM-98)

Nottingham, UK

1st - 4th April 1998

Programme Co-Chairs

Frank E Ritter (Nottingham, UK)
Richard M Young (Hertfordshire, UK)

Programme Committee

Paul Brna (Leeds, UK)
Axel Cleeremans (Brussels, B)
Fernand Gobet (Nottingham, UK)
Daniel Kayser (Paris, F)
Christian Lebière (Carnegie Mellon, Pittsburgh, USA)
Gerhard Strube (Freiburg, D)
Maarten van Someren (Amsterdam, NL)

Local arrangements committee

Frank Ritter (Chair), David Golightly, Gary Jones, George Kuk

Support provided by

Department of Psychology, University of Nottingham
Department of Psychology, University of Hertfordshire
Engineering and Physical Sciences Research Council (UK)
Department of Computer Science, University of Nottingham
European Research Office of the US Army
US Air Force European Office of Aerospace Research and Development
EuroSoar UK

Table of Contents

Invited talks

Papers

Symposia

Poster abstracts

Preface

This document is a record of the papers and other material presented at the Second European Conference on Cognitive Modelling (ECCM-98), which was held on the campus of the University of Nottingham from 1st to 4th April 1998. The conference attempted to build on the success of the first meeting in the series, which had been held in Berlin in November 1996. As well as presented papers, the conference included tutorials (on ACT-R, Soar, and COGENT), invited addresses, symposia, posters, and demonstrations of models and modelling software.

In the call for papers, we tried to encourage the submission of papers which report both a running (i.e. implemented) computer model *and* some empirical data against which the model can be compared. We were pleased with the results. Almost all the papers submitted included both those components, the only real exceptions being papers where such a criterion was not appropriate, such as those dealing with tools or methodology.

We were also pleased by the quality of the papers submitted. The quantity and the quality were high enough that we were able to be selective, while still having enough papers for a full conference programme. Within the constraints of preparing for a conference — where a large number of papers have to be assessed in a short interval of time, and where decisions about acceptance or rejection have to be made on the basis of a paper as it stands — we attempted some serious refereeing. Of course, the review process could not be as thorough as it is for journal publication, but each paper was read and commented on by at least two members of the programme committee, and we tried hard to make the feedback given to authors clear and informative, especially in cases where changes were suggested or where reasons for rejecting a paper (or accepting it as a poster) were offered.

Of the 40 papers submitted, we accepted 20, and invited a further 10 to be presented as posters (6 of which took up the invitation). We also accepted 5 of the 6 contributions submitted as posters. Our main criterion for posters was that they should be of relevance to the cognitive modelling research community, but possibly reporting work that is too preliminary to be presented as a main paper, or possibly focused on a model without as yet including the comparison to data.

As well as having representation from a wide range of areas of cognitive modelling, the conference is a truly international event. Contributions to the programme came from 14 different countries: the UK (11), USA (9), France (8), Germany (7), Italy (3), Belgium (2), Finland (2), The Netherlands (2), Australia, Bulgaria, Greece, Japan, Sweden, and Switzerland (1 each). It should be noted that the author index to these Proceedings lists no fewer than 80 authors who have contributed to the conference.

It is appropriate to end this introduction with some thoughts about the nature of the ECCMs and how they relate to other meetings. Many of us tend to think of cognitive modelling as a research activity dominated by the USA. Yet even in the USA, the publication of descriptions of running computer models and their detailed comparison with empirical data is comparatively rare, and there seem to be no meetings attempting what ECCM is trying to do. The closest that comes to mind is the annual meeting of the Cognitive Science Society. Yet the feel of that meeting is entirely different to ECCM, in part because it is indeed a meeting of a particular scientific society (which ECCM is not), and in part because Cognitive Science (as viewed by the Society) is a broad field, of which cognitive modelling is seen as just a small part. Mainly, what makes ECCM distinctive is the point we stressed above, namely our emphasis on the presentation of both an implemented model and its comparison against empirical data, and on keeping a reasonable balance between the two.

At the time of writing, nothing has been decided about the location and timing of any third ECCM. There are some uncertainties about future meetings, and especially about our relationship to the ongoing series of European Conferences on Cognitive Science (ECCS: St Malo, 1995; Manchester, 1997; Sienna, 1999). These matters are to be discussed at a special session during the conference. We certainly hope that something recognisably similar to the first two ECCMs continues, though perhaps still more international in flavour. To judge from the papers at this conference, cognitive modelling in Europe is in a comparatively healthy state.

Richard M Young and Frank E Ritter
Hatfield and Nottingham
March 1998

Invited Talks

Modeling Neural Function and High Level Cognition

Marcel Just
Center for Cognitive Brain Imaging
Department of Psychology
Carnegie Mellon University
Pittsburgh, PA 15213, USA
+1 (412) 268-2791
just+@cmu.edu

ABSTRACT

Recent brain imaging findings suggest several new assumptions concerning the architectural properties of the neural systems that underlie high level cognition, such as language, comprehension, visual cognition, and problem solving. Some of these assumptions have to do with

1. resource-constrained processing and task assignment;
2. dynamic configuration and resource recruitment;
3. functional embedding, self-similarity, and interaction among the components of the cognitive system;
4. a preference ordering for the types of processing that each cognitive component can perform (graded specialization).

The 4CAPS computational modeling system implements these assumptions, with the goal of accounting not only for processing times and error probabilities, but also for the amount of brain activation observed in each of the activated component neural systems. 4CAPS consists of several component processing modules, each of which is a parallel production system with some connectionist properties, and each of which is intended to correspond to the function of an underlying large-scale neural network. The component production systems are highly interactive with each other, operate in parallel, and have a task allocation regimen based on graded specialization and resource availability.

Mechanisms and Implications of Pervasive Episodic Memory

Erik M. Altmann
Psychology Department and Krasnow Institute
George Mason University
Fairfax, VA 22030 USA
+1 703-993-1326
altmann@gmu.edu

ABSTRACT

This paper investigates the memory phenomena underlying directed access to hidden objects. A computational cognitive model is described that encodes long-term episodic traces automatically whenever it attends to an object in its environment. Later, if an object of interest is hidden from view, the model can try to remember seeing it. This involves generating appropriate cues from memory to try to trigger episodic traces encoded while attending to that object. The underlying cognitive architecture (Soar) constrains the nature of these cues and the processes required to generate them. These constraints lead to a theory of episodic indexing, which is that people store simple information about attention events in large amounts, but make use of it only to the extent that they are able to generate appropriate images from memory. Episodic indexing helps characterize the cognitive cost of a cluttered interface.

Keywords

Cognitive simulation, episodic memory, human-computer interaction, Soar

INTRODUCTION

Our surroundings are filled with information. Most of this is hidden to us at any given time, being out of our field of view, yet we manage to gain access to it when we need to. For example, we might recall seeing a figure in a book, or a key phrase, and then return to that area in the book to refresh our memory, or to examine the context more carefully.

This paper investigates the memory phenomena underlying such access to external information. What do people encode about something they see, such that they can remember later that it exists? We would like to know both what information is stored, and under what circumstances. Second, what causes the retrieval of these memories? People typically navigate their environment for a purpose rather than haphazardly, implying some knowledge of a target to be visited. We would like to understand the role of domain knowledge in mediating access to what we know exists in our environment.

Our approach to these questions is to represent the phenomena explicitly using a cognitive architecture, Soar (Newell, 1990; Rosenbloom, Laird, & Newell, 1992). Soar includes mechanisms grounded in psychological theory and data that impose constraints on the representation of behavior. Applied to hidden-object access, these constraints imply that people store large amounts of information about their environment, but retrieve it only occasionally and with requisite knowledge and cognitive effort.

The paper is organized as follows. We first characterize the kind of task that requires the fine-grain episodic memory for efficient performance, and introduce the model using simple hypothetical examples to illustrate its encoding and retrieval processes. We then offer an accounting of the memory bandwidth implied by pervasive episodic encoding. Finally, we examine the theory for consistency with other findings on episodic memory, and for design of interfaces to extensive information environments.

THE MODEL

The kind of hidden-information access we are interested in studying is illustrated by the following scenario. A computer user is working with an application that generates much more information than fits on the screen at once. Most of this information is hidden, scrolled out of the way by the application to make room for the new information that it generates continually. This old information remains accessible, and the user occasionally scrolls some of it back into view. Thus, the user appears to have a memory that functions as an index to the environment. Much as the index in a book supports looking up a term of interest, the episodic index stored in memory supports "looking up" objects of interest in the environment. We are interested in how this index is created in memory, and how it is later accessed. In the following, we use examples from a hypothetical database programming task. The real task simulated by the model is described elsewhere (Altmann & John, in press; Altmann, 1996; Altmann, Larkin, & John, 1995).

The model's main mode of performance is a kind of comprehension in which it tries to gather information about objects in its environment. This is a generalized and simplified representation of interaction with an information-rich environment. In particular, it is simplified in that the model does not construct the complex mental structures generally associated with comprehension of text (e.g., Lewis, 1993; Kintsch, 1998).

The model selects goals to comprehend objects and issues commands to change the display. Some commands generate new information, and some scroll to old information. The model uses this external information as it tries to comprehend objects.

To comprehend a particular object, the model selects subgoals that retrieve information about that object. Information can come either from the display (an external source) or from LTM (an internal source). For example, suppose the model is comprehending a data structure that represents a student record. The student record contains a field for the student's Social Security Number (SSN), which is displayed on the screen. To retrieve information about this field, the model selects an attend subgoal. Suppose (for simplicity) that the model attends only to the field and not to the actual number stored there. This act of attention would add the following attribute-value pair to WM.

```
(^field ssn)            From attending to SSN field.
```

Alternatively, if this information is not available externally but the model has the appropriate domain knowledge, the same information can be recalled from LTM. To do this, the model selects a probe subgoal. For example, the model might probe with the SSN field, perhaps to see if this activates any other information relevant to the student record. Probing and attention are symmetrical in that a probe can look exactly like the output of attention.

```
(^field ssn)            From probing with SSN field.
```

Under episodic indexing, attention and probing process another kind of element, one which represents the actual event of attending to an object. Attention automatically adds this element to WM as a side effect of attending to an object. Thus the full outcome of attending to an SSN field would be the following.

```
(^field ssn)            From attending to SSN field.
(^attended-to ssn)      From attending to SSN field.
```

The same representation could also be produced by a probe, consistent with the attention-probing symmetry noted above. The probe below represents the model asking itself, "What do I know about the event of attending to an SSN field?"

```
(^field ssn)            From probing with SSN field.
(^attended-to ssn)      From probing with SSN field.
```

We refer to an attribute-value pair like attended-to ssn, when generated by a probe, as an image of attending to an object. The term image is meant to suggest a code like that produced by attention, namely more like a percept than an abstraction or a concept. Beyond this, we do not attempt to interpret the model's images phenomenologically, or psychologically in terms other than how they function in the model. For example, their symbolic nature reflects Soar's representation language and is not intended as a statement in the debate over propositional vs. analog spatial codes. In general, LTM contains many kinds of codes (Bower, 1975), and in particular expert programmers often use vivid imagery to understand programs, including color, sound, and dancing symbols (Petre & Blackwell, 1997). Amidst this diversity it seems reasonable to posit a code representing the event of attending to an object.

Thus the model can imagine attending to an object, providing it has the knowledge to do so. Such imagining, and hence the requisite store of images, is the basis of the retrieval processes of episodic indexing.

Learning in Soar

Encoding information about the environment is a form of learning, and requires that the model modify its long-term knowledge representation. In Soar, all long-term knowledge is represented productions. These are condition-action rules like the one below. If the condition part (above the arrow) matches a structure in WM, then the action part (below the arrow) adds new elements to WM. The production below acts as a declarative memory, because it associates an object (a student record) with facts about that object (that it has an SSN field). In general, all the model's operations, like attending to objects, generating probes, and recalling facts, depend on knowledge represented as productions.

```
      (^structure student-record)       Condition:
                                  Student record in WM.
  -->
     (^field ssn)            Action:  Put SSN field in
                                                   WM.
```

The model learns by acquiring new productions. The learning mechanism is part of Soar. It is unified with Soar's knowledge-representation language (productions) and control structure (goals) in that Soar acquires new productions in response to achieving goals (Laird, Rosenbloom, & Newell, 1986). A new production, or chunk, represents an inference that may have taken several steps to make. The chunk is added to LTM, making the inference available in a single step from then on.

For example, suppose the model's goal were to find the sum of two numbers (4 and 7) and that although it could not retrieve the sum directly from memory, it knew a procedure for adding by counting up from one of the addends. The goal to find the sum would be implemented by subgoals that might involve initializing a running sum to the value of one addend, invoking the counting procedure, and recognizing when the count equaled the other addend. The result (11) would represent achieving the goal, and Soar would encode a chunk associating the relevant inputs to the counting procedure with the new result. In the future, this chunk will compute 4 + 7 = 11 without subgoals, bypassing the counting procedure.

In general terms, a chunk encodes an association between an inferred result (e.g., the sum) and the WM elements on which the inference is based, which we refer to here as premises (e.g., the addends). The premises have either already contributed to achieving the current goal or were in WM when the goal was selected. The result is inferred from the premises through a sequence of intermediate production firings. A chunk will fire immediately in the future if WM contains the same premises.

The chunking process does very little induction or generalization. The result essentially becomes the chunk's action and the premises become the chunk's conditions, though there is some variabilization (Laird, Rosenbloom, & Newell, 1986). This makes a chunk specific to its encoding context, consistent with the encoding specificity principle (Tulving, 1983). This specificity acts as a hard constraint on the nature of the process for retrieving learned knowledge (Howes & Young, 1997).

Encoding the Episodic Index

The model contains two key assumptions about the process of attending to an object. Both assumptions are related to the event of attending. The first assumption is that the event itself is worth representing in WM, apart from the object of attention. The second assumption is that all attention events are goal-directed. This assumption says that the model is always looking for new information about the object it is trying to comprehend, and therefore automatically takes any attention event to contribute to the current goal. The two assumptions together operationalize what we might think of informally as "paying attention to" or "concentrating on" what we are doing. The important implication is that if the model "pays attention" to an event, this enables remembering the event because it causes chunks to be acquired.

The first assumption (that attention events are noteworthy) is implemented as follows. When the model attends to an object, it records the event using its internal clock. That is, it associates the WM code for the attention event with the current value of an internal variable that is updated periodically. For example, when the model attends to the SSN field of a student record, the complete representation created in WM is something like the following.

```
(^attended-to ssn)          From attending to SSN
field.
(^event ssn ^time t42)      From attending to SSN
field.
```

The model's internal clock ticks when it selects a new object to comprehend (meaning that the model's sense of time is keyed to its train of thought). All objects attended while comprehending that object are encoded in LTM with the current time symbol.

The second assumption (that episodic processing contributes to the current goal) is implemented by associating the time symbol with the current goal in WM. This causes Soar to build a chunk, as described in the previous section. The premise of the chunk is the attribute-value pair representing attention to the SSN field, and the result is the time symbol. The two are linked by the inference that the SSN field was attended now. The chunk is shown below (named attended-ssn for reference later).

```
chunk: attended-ssn     Chunk for an attention event.
 (^attended-to ssn)
 -->
 (^event ssn ^time t42)
```

Attended-ssn represents an attention event. This makes it an episodic trace, as distinct from a semantic trace with no temporal content (Tulving, 1983). It functions as one entry in an index of objects encountered in the environment. In the future, if no SSN field is visible, the model can look up the SSN field in this index by attempting to cause this chunk to fire. If the lookup is successful, then the model can infer that it attended to an SSN field in the past, even though no such field is currently visible. The lookup and inference processes are described in the next section.

Retrieval from the Episodic Endex

The episodic index consists of a set of chunks, each of which associates an attention event with a time symbol. Suppose that a particular attention event occurred long enough in the past that it is no longer active in WM and that the corresponding object is no longer in view. The model can use its episodic index to see if the object exists somewhere in the environment. This requires two steps. The first is to generate the cue necessary to get an episodic chunk to fire. We can think of this as "looking up" the object. The second is to make the appropriate inferences based on any recalled time symbols. We can think of this as acting on the information retrieved from the lookup.

To look up an object, the model must add to WM an image of attending to that object, as a cue for triggering episodic chunks. As discussed previously, an image can appear in WM either through attention, which generates the image from an external stimulus, or through probing, which generates the image from memory. In either case, an image appearing in WM will activate all episodic chunks acquired whenever the corresponding object was attended in the past. Production imagine-ssn, below, generates the necessary probe for the SSN field.

```
production: imagine-ssn
  Conditions testing that it's relevant to know that
  an SSN field was seen.
 -->
 (^attended-to ssn)                       A1
 (^imagined ssn)                          A2
```

Imagine-ssn will fire in a situation in which it would be useful to remember seeing an SSN field. For instance, suppose (as we did previously) that the model were asked whether a given database record contained confidential information. The model might try to recall seeing an SSN field by firing imagine-ssn. When imagine-ssn fires, A1 adds to WM an image of attending to the SSN field, providing an opportunity for a chunk like attended-ssn to fire. A2 tags this image as generated from memory rather than from a stimulus. In general, there could be many situations in which it might be useful to imagine an SSN field. Each would be represented in a production like imagine-ssn (with different conditions).

If we suppose that attended-ssn fires in response to imagine-ssn, then WM will contain the following elements.

```
(^attended-to ssn)           From imagine-ssn .
(^imagined ssn)              From imagine-ssn.
(^event ssn ^time t42)       From attended-ssn.
```

From these elements the model can infer that an SSN field exists in the environment. The production that makes this inference is recall-seeing-object, below. This production belongs to the set of generic mechanisms that form part of the model's static knowledge.

```
production: recall-seeing-object
 (^attended-to <o>)                      C1
 (^imagined <o>)                         C2
 (^event <o> ^time <then>)               C3
 (^time <now> != <then>)                 C4
 -->
 (^recall-seeing <o>)
```

Recall-seeing-object's conditions, numbered on the right, are as follows. Conditions C1 and C2 test that there is an image in WM that was generated internally rather than from an external stimulus.[1] C3 and C4 test that the image was attended in the past. The single action summarizes

[1] Angle brackets around a letter (e.g., "<o>") indicate a variable. If the same variable occurs in multiple conditions, it must have the same value in each condition for the production to fire. Thus, for example, C1 and C2 test that the object bound to <o> is both attended-to and imagined.

what is expressed by the conditions. It adds to WM the recollection of having seen the object.

The identity comparison in C4 is the only operation afforded by time symbols. Thus time is categorical, rather than ordinal or interval, and the only categories are present (the current comprehension goal) and past (any previous goal). The model cannot compute, for example, the interval between two events. This information-leanness is consistent with qualitative aspects of the rapid decay of unelaborated temporal codes in people (Underwood, 1977).

The nature and use of the episodic index is shaped by Soar's constraints on learning. Because Soar makes a chunk specific to its encoding context, attended-ssn's conditions are tied to the object code that appeared in WM during the attending event. This specificity implies that recalling the existence of an object must be preceded by imagery involving the object.

Summary of Assumptions

There are four theoretical assumptions that shape how the model acquires and retrieves memories for attention events. The first assumption is that the attention event itself is worth symbolizing in WM, in addition to the attended object. The second assumption is that attention is an integral part of comprehension and thus contributes to every comprehension goal. These two assumptions are hypotheses that we have embodied in the model.

The third and fourth assumptions come with Soar. The third is that all knowledge that contributes to achieving a goal is stored permanently in chunks. The fourth is that chunks are specific to their encoding context.

Together, these assumptions imply that chunk acquisition in the model will be pervasive and automatic, and that retrieval will be effortful. Learning will be pervasive because the model will encode a new episodic chunk for every object it attends to. This learning is automatic, in that the model exercises no control over whether or not to learn, and in that learning is a side effect of attentional processing rather than an end in itself. Retrieval will be effortful because learning involves little induction. To get chunks to fire, cues describing the original encoding context will have to be generated from memory.

Knowledge Distinguished by Dperation

Episodic indexing encodes information about dynamic information arising during task performance. It also allows us to make distinctions among the different operations facilitated by the domain knowledge that one brings to a task. Domain knowledge is involved in three operations:

- Attention. During the acquisition episode, the model must know what to attend to in the first place. Thus the model must be able to identify objects and understand them to be relevant to the task at hand.
- Retrieval. During the retrieval episode, the model must (a) be able to generate an image, and (b) do this when the results of a successful probe on that image would be useful. Thus retrieval depends on both visual familiarity and semantic understanding specific to the particular domain.
- Action. The decision to revisit a hidden object is distinct from recalling that it exists. There might be other means for acquiring the information that the object could provide, and there might be no reason to act on the recollection.

Thus the model points to several operations by which relatively static domain knowledge helps us gain access to the relatively dynamic information around us.

PRODUCTIONS ENCODED AND FIRED

The model simulates 10.5 continuous minutes of problem solving, spanning the encoding and retrieval episodes of a number of scrolling events (Altmann, 1996). This extended lifetime served as a form of methodological control during our analysis. A sufficiently close examination of the data to construct the model was the best way to avoid missing events between acquisition and retrieval that might have recoded or otherwise affected the nature of the participant's episodic index. This extended lifetime also serves to illustrate the implications of pervasive episodic encoding for the bandwidth of the model's memory system in terms of the number of chunks acquired and fired.

Figure 1 tabulates productions and firing counts according to four categories of knowledge (arrayed horizontally). The top bar indicates the number of productions in each category when the model stops, including all preloaded productions and all chunks acquired as the model runs. The bottom bar indicates the total number of production firings in each category during the model's run.

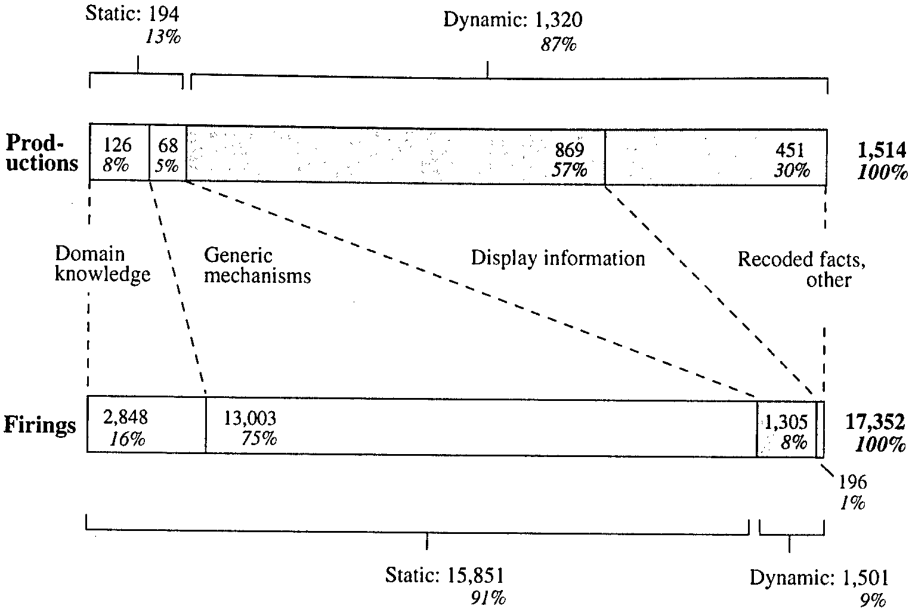


Figure 1: Production and firing counts.

The right half of the picture (shaded) shows that most of the model's productions are acquired by learning, but fire seldom because they are specific to their encoding context. The large number of chunks at the end of the run (1320) indicates the extent to which learning is pervasive. In terms of real time, the model is encoding roughly two chunks per second.

Few of these chunks fire, but some do. In particular, chunks encoding some aspect of the display account for 8% of firings. Thus the model's behavior depends in part on a memory for specific external situations that arise during task performance.

The left half of the picture (unshaded) shows that the model begins with a small number of preloaded productions that account for most of its processing. Preloaded productions number 194 (13%), of which 126 (8%) represent domain knowledge that we attribute to the programmer. This knowledge lets the model attend to external objects, generate cues, and recall facts about objects. It also tells the model what commands it can issue and what objects are important to comprehend and therefore select as comprehension goals.

Expertise should be flexible, in that it should guide behavior under a variety of appropriate circumstances. In our model, a large number of static domain-knowledge productions (93 out of the 126 indicated in Figure 1) represent either comprehension goals or attend or probe subgoals. These 93 productions account for all 499 goal and subgoal selections that occur as the model runs, for a mean of 5.4 goals per production. They also account for 2,518 out of the 2,848 firings of domain-knowledge productions. These measures indicate that to a large extent the model's goal and subgoal productions transfer among situations rather than being hardwired to a particular one.

The category of preloaded productions labeled generic mechanisms accounts for 75% of total production firings, despite being only 5% of the total number of productions. These are domain-independent productions like recall-seeing-object, discussed earlier, which infers the existence of a hidden object from an episodic trace. The high firing rate of mechanistic productions is consistent with their being the most general productions in the model and potentially general across many domains.

The production and firing counts over the model's lifetime illustrate the implications of pervasive episodic learning. In a few minutes of simulated time, the model acquires a great deal of dynamic information about its environment and stores it permanently in LTM. Some of these chunks transfer in the near term, firing seconds to minutes (of simulated time) after being created. The fast rate of learning -- 1,320 productions over 10.5 minutes – suggests that Brooks's (1977) estimate of tens or hundreds of thousands of rules making up a programmer's static domain knowledge may account for only part of what generates expert performance. There may in addition be a vast and constantly growing store of rules capturing dynamic knowledge.

DISCUSSION

Below we discuss the relationship of episodic indexing to previous conceptions of episodic memory in Soar and to a related theory advanced to account for expanded working memory for domain experts. We then speculate on episodic indexing and the cognitive cost of clutter.

Episodic Memory in Soar

Episodic memory is a natural construct to study in Soar. Learning is closely integrated with performance, meaning that events are easy to capture and store in LTM. Moreover, chunk conditions are determined by a process that gives chunks an inherently episodic quality. The chunking mechanism traces from a result back to the WM elements from which the result was generated, encoding an association between the result and important elements of context in which it was encoded. Thus the simple existence of a chunk represents some episodic information. A model can gain access to this information by generating cues that would cause the chunk to fire if it existed, then by monitoring WM for the appearance of the chunk's result. Several Soar models have addressed episodic memory in these terms (e.g., Rieman, Young & Howes, 1996; Rosenbloom, Newell, & Laird 1991).

However, episodic indexing requires richer information to decide whether an object was actually attended at some time in the past. Below we examine the constraints met by the model's time symbols, and how these constraints arise from the interaction of pervasive episodic encoding (an assumption in our model) with encoding specificity (an architectural constraint inherited from Soar).

Episodic encoding extends to probe events as well as attention events -- that is, the model encodes episodic chunks for both. This follows from assuming that attention is integral to comprehension and thus contributes to every comprehension goal. Probing contributes equally to comprehension, and thus with respect to episodic learning the model treats probing and attention symmetrically.

This symmetry could lead to confusion should the model probe repeatedly with the same image. A particular probe will trigger episodic chunks from all previous probes, potentially leading the model to mistake these past probes as attention events. A kind of reality monitoring (Johnson & Raye, 1981) is necessary to avoid this mistake (and hence to avoid scrolling to imaginary objects). To support this reality monitoring, episodic chunks must contain enough information about the source of a memory (attention vs. probing) to let the model discriminate past attention events from past probe events.

Identifying past attention events must be done indirectly because source information cannot be represented explicitly in episodic chunks, when the source is the environment. This seems a surprising constraint, but it follows from encoding specificity. When building a chunk, Soar traces from the result back to premises existing before the result was generated, and encodes these premises as conditions. Therefore, if source information is a result, it also becomes a condition. Thus if a chunk has an action identifying an object as real, its conditions can never be met by an image alone. However, by the same logic, a chunk can have an action identifying an object as imagined and still be triggered by an image. Thus the model includes an probe tag with each chunk built during a probe event (see Appendix).

At retrieval time, these probe tags provide part of the information necessary to decide if the object of interest was ever attended. To make this decision, the model must identify all episodic chunks triggered by the current probe but built during past probes, and subtract them from the total set of episodic chunks triggered by the current probe. If the resulting set is non-empty, then the object was attended in the past. In terms of predicate calculus, the model tests an existential quantifier ("Did I recall an attention event?") by testing a negated universal quantifier ("Did I recall any event that was not a probe?"). This requires that each episodic chunk be uniquely identifiable. Because chunks are identifiable only by their results, this in turn requires that each episodic chunk have a uniquely identifiable result. This requirement cannot be met by a fixed set of symbols because at most one instance of any particular symbol can be represented in WM at any given time whereas the number of distinct events to represent is effectively unbounded. The requirement is met by the model's time symbols (as illustrated in the Appendix), because each is unique and they are generated anew at regular intervals.

Thus episodic indexing contrasts with previous Soar formulations of episodic memory in which multiple chunks may have the same result (e.g., Rieman, Young & Howes, 1996; Rosenbloom, Newell & Laird 1990). The episodic representation in our model is implied by theoretical assumptions interacting with task requirements in a way that does not constrain these other models. Our assumptions specify an indiscriminate encoding of episodic chunks, and the task requires that chunks from attention events transfer to probe events. However, this transfer requirement combined with encoding specificity restricts the source information that episodic chunks can represent. To compensate they are made discriminable by their results, allowing the model to partition past events into probe events and all the rest. This shaping of a representation by a complex interaction of constraints illustrates the benefit of taking a comprehensive and integrated approach to modeling cognitive phenomena (Newell, 1973).

A Form of Long-Term Working Memory

Episodic indexing posits that access to dynamic information depends on static information that one brings to the task. In this it is congruent with long-term working memory (LT-WM; Ericsson & Kintsch, 1995), of which a central claim is that long-term knowledge (as opposed to inherent WM capacity; e.g., Just, Carpenter & Hemphill, 1996) accounts for functionally expanded WM in domains in which one has expertise. Episodic indexing and LT-WM both propose that people store information rapidly in LTM, using domain knowledge to organize it and gain access to it later.

Episodic indexing extends LT-WM in the direction of leaner and more ubiquitous memory structures acquired at encoding time. The most routine application of LT-WM reviewed by Ericsson and Kintsch (1995) is text comprehension, but even this involves online encoding of memory structures that represent potentially intricate semantic mappings. For example, in referent resolution the comprehender must represent the connection between a pronoun and what it stands for, which is a semantic association that is not always straightforward to establish. By contrast, the episodic index is a one-way mapping from semantic to episodic codes which lacks the network structure that typically characterizes semantic memory.

The Cost of Clutter

Episodic indexing suggests that clutter has a cognitive cost, due to the paucity of information encoded with episodic traces and the effect this has at retrieval time. An episodic retrieval indicates the existence of an object of interest but not its whereabouts. This is consistent with the difficulty that even experienced users have in recalling features of interfaces (Mayes, Draper, McGregor, & Oatley, 1988; Payne, 1991), and with findings that spatial and location knowledge is not automatically encoded in real-world task environments (Lansdale, 1991). It is also consistent with the generally reconstructive nature of memory for the source of an item (Johnson, Hashtroudi, & Lindsay, 1993).

One possible strategy for dealing with clutter might be to add spatial information to the episodic information encoded during attention. However, encoding specificity as implemented in Soar predicts that any such information would place a heavy burden on the retrieval process. Location information encoded in the actions of a chunk would also be present in the conditions, thus requiring that location cues be generated at retrieval time. This would not completely defeat the purpose, because the model could use the same kind process it now uses to generate and recognize images at retrieval time. However, more cues would have to be generated, requiring both more cognitive effort and more knowledge from which to generate them.

This shifts the emphasis to alternative strategies. One alternative might be to infer location from the nature of the target item. For example, applications often deposit different kinds of output into different windows. In such environments a reliable and easily-retained mapping from content to location should reduce the cost of clutter. Another strategy might be search, implying that reliable, easily-retained, and flexible searching tools also reduce the cost of clutter. More generally, the implication of episodic indexing is that access to hidden objects requires a reconstructive memory process that becomes more costly the more source information is stored with the target item. Thus users are likely to mitigate clutter by inferring location as needed, implying that interfaces to extensive information environments should support such inferences with direct, structured and learnable item-location mappings.

CONCLUSIONS

We propose that people store simple dynamic information in long-term memory as a matter of course, and use this information to index their environment. Our theory of episodic indexing makes two main claims:

- Pervasive and automatic encoding. People acquire large amounts of recognitional, episodic information about attention events, as a side effect of attention.
- Semantic, image-based retrieval. People retrieve this episodic information as a function of pre-existing knowledge that generates image cues when semantically appropriate.

The generality of these claims rest on the generality of their theoretical underpinnings. Soar's chunking mechanism (which predicts goal-based learning and encoding specificity) has been offered as a universal account of learning (Laird, Rosenbloom, & Newell, 1986; Newell, 1990), and our additional assumptions about the integration of episodic processing, attention, and comprehension are domain-independent. Thus episodic indexing may operate whenever people pay attention to what they are doing, and know the domain well enough to generate the right cues at the right time.

ACKNOWLEDGMENTS

This work was supported in part by a Krasnow Postdoctoral Fellowship, in part by the Wright Laboratory, Aeronautical Systems Center, Air Force Materiel Command, USAF, and ARPA under grant number F33615-93-1-1330, in part by the Office of Naval Research, Cognitive Science Program, Contract Number N00014-89-J-1975N158, and in part by the Advanced Research Projects Agency, DoD, monitored by the Office of Naval Research under contract N00014-93-1-0934. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies or endorsements, either expressed or implied, of Wright Laboratory, the Advanced Research Projects Agency, the Office of Naval Research, or the U.S. Government..

REFERENCES

Altmann, E.M. (1996). *Episodic Memory for External Information.* Doctoral dissertation, School of Computer Science, Carnegie Mellon University, Pittsburgh.

Altmann, E.M. & John, B.E. (in press). Modeling episodic indexing of external information. *Cognitive Science.*

Altmann, E.M., Larkin, J.H. & John, B.E. (1995). Display navigation by an expert programmer: A preliminary model of memory. *CHI 95 Conference Proceedings* (pp. 3-10). New York: ACM Press.

Anderson, J.R. (1990). *The Adaptive Character of Thought.* Hillsdale, NJ: Lawrence Erlbaum.

Bower, G.H. (1975). Cognitive psychology: An introduction. In W. Estes (Ed.), *Handbook of Learning and Cognitive Processes* (Volume 1). Hillsdale, NJ: Lawrence Erlbaum.

Brooks, R.E. (1977). Towards a theory of the cognitive processes in computer programming. *International Journal of Man-Machine Studies*, 9, 737-751.

Ericsson, K.A. & Kintsch, W. (1995). Long-term working memory. *Psychological Review,* 102, 211-245.

Howes, A. & Young, R.M. (1997). The role of cognitive architecture in modeling the user: Soar's learning mechanism. *Human-Computer Interaction*, 12, 311-343.

Johnson, M.K. & Raye, C.L. (1981). Reality monitoring. *Psychological Review*, 88, 67-85.

Johnson, M.K., Hashtroudi, S. & Lindsay, D.S. (1993). Source monitoring. *Psychological Bulletin*, 114, 3-28.

Just, M.A. & Carpenter, P.A. (1992). A capacity theory of comprehension: Individual differences in working memory. *Psychological Review*, 99, 122-149.

Kintsch, W. (1998) *Comprehension: A paradigm for cognition.* New York: Cambridge University Press.

Laird, J.E. Rosenbloom, P.S. & Newell, A. (1986). Chunking in Soar: The anatomy of a general learning mechanism. *Machine Learning*, 1, 11-46.

Lansdale, M.W. (1991). Remembering about documents: Memory for appearance, format, and location. *Ergonomics*, 34, 1161-1178.

Lewis, R.L. (1993). *An Architecturally-Based Theory of Human Sentence Comprehension.* Doctoral dissertation, School of Computer Science, Carnegie Mellon University, Pittsburgh.

Newell, A. (1990). *Unified Theories of Cognition.* Cambridge: Harvard University Press.

Newell, A. (1973). You can't play 20 questions with nature and win: Projective comments on the papers of this symposium. In W.G. Chase (Ed.), *Visual Information Processing*. New York: Academic Press

Petre, M. & Blackwell, A.F. (1997). A glimpse of expert programmers' mental imagery. *Empirical Studies of Programmers: 7th workshop*. New York: ACM Press.

Rieman, J., Young, R.M. & Howes, A. (1996). A dual-space model of iteratively deepening exploratory learning. *International Journal of Human-Computer Studies*, 44, 743-775.

Rosenbloom, P.S., Laird, J.E., & Newell, A., Eds. (1992). *The Soar Papers: Research on integrated intelligence*. Cambridge: MIT Press.

Rosenbloom, P.S., Newell, A. & Laird, J.E. (1991). Towards the Knowledge Level in Soar: The role of the architecture in the use of knowledge. In K. VanLehn (Ed.), *Architectures for Intelligence*. Hillsdale, NJ: Lawrence Erlbaum.

Tulving, E. (1983). *Elements of Episodic Memory*. New York: Oxford University.

Underwood, B.J. (1977). *Temporal Codes for Memories: Issues and problems*. Hillsdale, NJ: Lawrence Erlbaum.

APPENDIX

Below we present a complete picture of the processing that occurs when the model probes with an image and retrieves episodic chunks. (This elaborates on the process described in the section, Encoding the Episodic Index.) In the general case, the episodic chunks retrieved by a probe will be of two kinds: those encoded during attention events and those encoded during (past) probe events. Only those acquired during probe events will contain source information in their actions (as discussed in the section, Episodic Memory in Soar). To determine whether the object of interest was actually attended in the past, the model computes the difference between the total set of episodic chunks retrieved and those representing probe events. The scenario below supposes that the model first probes for information about the SSN field, then actually attends to the field, then probes again.

At time t42, the model probes by placing an image in WM (A1) together with source information identifying the image as an image (A2).

```
production: imagine-ssn
  Conditions testing that it's relevant to know that
  an SSN field was seen.
 -->
 (^attended-to ssn)          A1
 (^imagined ssn)             A2
```

The model encodes an episodic chunk during the probe event, under the assumption of pervasive episodic encoding. Source information is included as a chunk action (A2) and hence also as a chunk condition (C2).

```
chunk: imagined-ssn     Chunk capturing a probe
event.
 (^attended-to ssn)          C1
 (^imagined ssn)             C2
 -->
 (^event ssn ^time t42)      A1
 (^probe t42)                A2
```

At time t43, the model actually attends to the SSN object, resulting in another episodic chunk.

```
chunk: attended-ssn     Chunk capturing attention
event.
 (^attended-to ssn)
 -->
 (^event ssn ^time t43)
```

Finally, at time t44, the model probes a second time (by firing imagine-ssn). This triggers the two episodic chunks described above, causing the following elements to enter WM.

```
 (^attended-to ssn)          From imagine-ssn.
 (^imagined ssn)             From imagine-ssn.
 (^event ssn ^time t42)      From imagined-ssn.
 (^probe t42)                From imagined-ssn.
 (^event ssn ^time t43)      From attended-ssn.
```

From these elements the model can infer that an SSN field exists in the environment. The production that makes this inference is recall-seeing-object, below. Condition C5 (not reported in the section, Using the Episodic Index) effectively subtracts the set of probe events (containing t42) from the set of probe plus attention events (containing t42 and t43). The leading minus sign ("-") negates the subsequent condition, meaning that WM cannot contain an element matching that condition. In our scenario, this negated condition holds for at least one past event (t43). Thus the production matches, inferring that SSN was attended at some point in the past (A1).

```
production: recall-seeing-object
 (^attended-to <o>)          C1, <o> = ssn
 (^imagined <o>)             C2
 (^event <o> ^time <then>)   C3, <then>=t43
 (^time <now> != <then>)     C4, <now>=t44
-(^probe <then>)             C5
 -->
 (^recall-seeing <o>)        A1
```

Why Operators' Cognitive Models are Hard to Incorporate into Design: The Case of Human Reliability Models

René Amalberti
(Institute of Aerospace Medicine, Department of Defence, France)
11 Boulevard Hotel de Ville
93600 Aulnay-sous-bois, France
+33 (0)1 48 66 85 52
rene-a@mail.imaginet.fr

ABSTRACT

Operators' models, or equivalent end-user models, have became a standard prerequisite for most man-machine system design. Nowadays, the designer can chose among a great variety of models: behavioral models of performance, running competence models, and cognitive models are available in a large range of granularity from quasi-neuropsychological models of memory to framework models of dynamic cognition. However, despite -- or maybe because of -- that variety, modelling the operator is still an area of uncertainty within the industry, with multiple forms and meanings, and with a persistent feeling that these models, whereas they should be useful, are hard to incorporate into the design process.

This paper focuses on the development and use of cognitive models of human reliability for the design of complex systems, and tries to understand biases and limitations of their use within the industry. In that sense, the paper is more industry-oriented than research oriented. It is divided into three sections. The first section details the range of existing cognitive models of human reliability and proposes a classification of these models into four main categories: error production models, error detection and recovery models, systemic models, and integrated safety ecological models. The example of the Aviation Industry shows how difficult it has been in the recent past to incorporate the most advanced of these models into design, whereas the same Industry had long complained about the lack of availabilily of cognitive operators' models.

The second section tries to explain the reason for the relative failure. It shows the inter-dependency existing between the category of cognitive model, the safety paradigm, and the strategy for design. Severe drawbacks may occur each time a model is used with the wrong safety paradigm or the wrong strategy for design. It also shows that the more cognitively-based the model is, the less it is incorporated into design. The lack of education in psychology of designers, as well as the lack of a clear procedure for incorporating such models into design, are among the most important factors explaining this lack of success.

The third and last section points to new directions in cognitive modelling to improve the fit between operator modelling and design requirements.

Papers

Acquisition and Transfer of Declarative and Procedural Knowledge

Todd R. Johnson
Department of Pathology
The Ohio State University
Columbus, OH 43228
614-292-3284
johnson.25@osu.edu

ABSTRACT

Recent results in cognitive skill acquisition suggest that task speed-up can be due to either speed-up of procedural knowledge or speed-up of the retrieval of declarative knowledge. This paper presents a single Act-R model that closely fits the data of two learning and transfer experiments conducted by Rabinowitz and Goldberg (1995). These experiments test three main hypotheses: 1) access to procedural and declarative knowledge speeds up as separate power laws of practice; 2) training on a large variety of problems leads to strengthening of procedural knowledge, whereas training on a small set of problems leads to the acquisition and strengthening of declarative knowledge; and 3) procedural knowledge operates in one direction only—from condition to action—whereas declarative knowledge can be cued by any of its elements. The model provides a good fit to the data, further validating Act-R as a model of the human cognitive architecture

Keywords

Declarative memory, procedural memory, learning, transfer, knowledge compilation, Act-R, Soar.

INTRODUCTION

One common view of cognitive skill acquisition is that it progresses from an interpretive stage to a procedural stage using some kind of knowledge compilation mechanism (Stillings et al., 1995; VanLehn, 1989). Such a mechanism produces procedural knowledge from the results of more deliberate, interpretive problem solving. This view has received a lot of empirical support. Several researchers have shown that knowledge compilation can model the transition from novice to expert behavior (Larkin, 1981; Newell & Rosenbloom, 1981). One major research effort, the Soar architecture, even asserts that knowledge compilation is the only mechanism required to account for all human learning (Newell, 1990). Researchers using Soar have been able to model a wide range of learning strategies (Miller, 1993; Rosenbloom & Aasman, 1990; Steier et al., 1987). Knowledge compilation mechanisms can also sometimes account for the ubiquitous power law of learning (Newell & Rosenbloom, 1981).

Recent results on the characteristics of declarative and procedural knowledge, however, threaten the simplicity of this view of skill acquisition, because they suggest that cognitive skill can also improve through the acquisition and strengthening of declarative memory elements (for a review see (VanLehn, 1996)). A number of experiments have suggested that the retrieval of declarative knowledge and the application of procedural knowledge speed up as separate power laws of practice. In other words, the time to retrieve a declarative memory speeds up as a power function of the number of retrievals, whereas the time to apply a procedure speeds up as a power function of the number of applications. This implies that cognitive skill can improve by acquiring and strengthening procedural or declarative knowledge, or some combination of the two.

Despite the intuitive nature of the distinction between declarative and procedural knowledge, the hypothesis that there are separate long-term memory stores for declarative and procedural knowledge remains a controversial issue in cognitive science. The controversy arises because, in theory, anything that can be modeled with two distinct long-term stores can also be modeled using only a procedural long-term store. For example, long-term procedural knowledge might add "Washington, DC" to working memory whenever working memory encodes a goal to determine the capitol of the United States. Working memory is widely thought to be a declarative store, so the declarative-procedural distinction applies only to long-term memory.

There is, however, mounting evidence in favor of the distinction. Cognitive neuroscientists have found a double dissociation between declarative and procedural knowledge—some patients can acquire new declarative knowledge, but not procedural, whereas other patients can acquire procedural, but not declarative. There is also evidence that the two kinds of knowledge have different retrieval characteristics: declarative knowledge can be primed by any of its components, but procedural knowledge only works in one direction: from a specific set of cues to an action. A review of these issues can be found in (Anderson, 1993).

Rabinowitz and Goldberg (1995) conducted two experiments that nicely illustrate many of the recent phenomena concerning skill acquisition and the distinction between declarative and procedural knowledge. These experiments use a learning and transfer paradigm to examine learning of declarative and procedural knowledge, and their different retrieval characteristics.

This paper presents a single Act-R model that accounts for the data in the two Rabinowitz and Goldberg experiments. In addition, the paper presents protocol results from a newly conducted experiment designed to

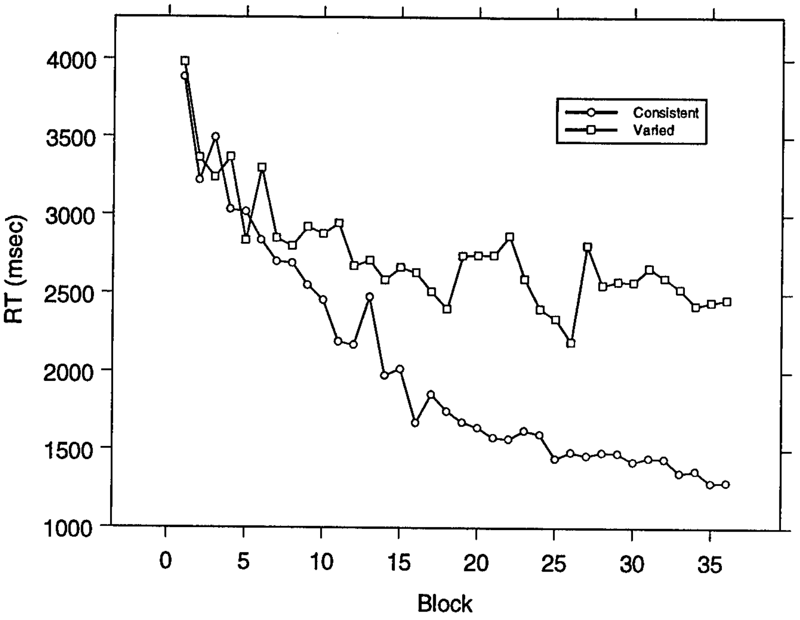


Figure 1: Mean response times during alphabet arithmetic training as a function of training group and practice block. Data plotted from original data by Rabinowitz and Goldberg (1995).

further test the assumptions of the experiments and the model.

THE RABINOWITZ AND GOLDBERG EXPERIMENTS

Both experiments used an alphabet arithmetic task, which consists of problems of the form *letter1 + number = letter2*, where *letter2* is *number* letters after *letter1*. For example, A+2=C, because C is 2 letters after A.

In Experiment 1, one group of participants (the consistent group) received training on 36 blocks of problems, where each block consisted of the same 12 problems. Another group of participants (the varied group) received training on 6 blocks of problems, where each block consisted of the same 72 problems. Thus, both groups received 432 training trials, but the consistent group practiced each problem 36 times, whereas the varied group practiced each problem only 6 times. The problems used addends from 1 to 6. Consistent problems had two occurrences of each addend, whereas varied problems had 12 occurrences.

In the transfer phase, both groups received 12 new addition problems, repeated 3 times. Rabinowitz and Goldberg reasoned that during training the consistent group would quickly acquire declarative knowledge of the answers and switch to retrieval, whereas the varied group would continue to count up the alphabet. Thus the consistent group would get a lot of practice at retrieving the answers to the same 12 problems, but relatively little practice on the procedural knowledge needed to count up the alphabet. In contrast, the varied group would receive little or no practice retrieving declarative knowledge, but a great deal of practice counting up the alphabet. When transferred to the 12 new addition problems, the consistent group should revert to counting up the alphabet, resulting in a dramatic decrease in speed. However, the varied group should show perfect transfer from the training problems to the new problems.

The training results are shown in Figure 1. Each point on the graph is the mean of the median response times for all subjects on a block of 12 problems. The different asymptotes support the assertion that varied participants practice procedural knowledge, while consistent participants switch to and then practice retrieval.

The transfer results, shown in Figure 2, support the predictions: the varied group shows perfect transfer, but the consistent group shows considerable slow-down.

Although Experiment 1 supports the predictions, it is also consistent with a procedural-only long-term store. The consistent subjects might have acquired problem-specific procedural knowledge that directly produces the answer to each problem. For example, knowledge of the form "If problem is A+2, then type C." Since this knowledge is specific to the 12 training problems, it would not have helped the participants during the transfer phase. This issue is examined in Rabinowitz and Goldberg's second experiment.

The second experiment attempts to determine whether consistent training leads to specific procedural knowledge, or to declarative knowledge. It is based on the hypothesis that declarative and procedural knowledge have different retrieval characteristics. Declarative knowledge is thought to be subject to symmetric retrieval, meaning that any part of a declarative memory element can act as a cue for the retrieval of that element. Procedural knowledge is thought to be subject to symmetric access, meaning that a procedure operates in only one direction: from condition to action.

Training in Experiment 2 was identical to Experiment 1, however, in the transfer phase, both groups were given 12 subtraction problems repeated 3 times. A subtraction

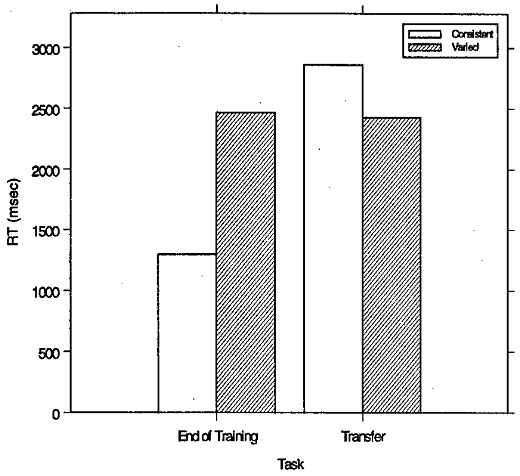


Figure 2: Mean response times for Experiment 1 as a function of task and group.

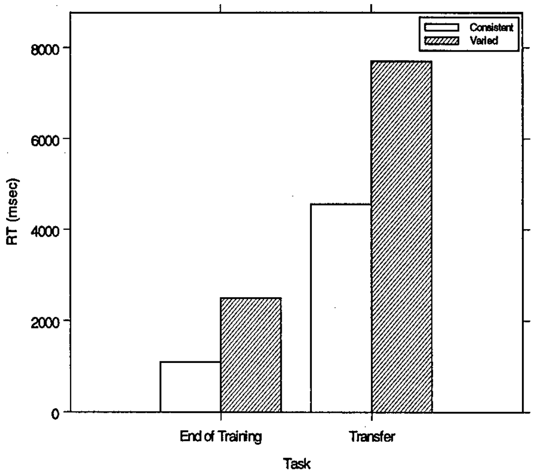


Figure 3: Mean response time for Experiment 2 as a function of task and group.

problem is of the form *letter1 - number = letter2*. For example, C-2=A. The 12 subtraction problems were inverted versions of the addition problems that both groups had seen during training. If the consistent group acquires declarative knowledge of the addition problems, the participants in this group should be able to solve the subtraction problems by retrieving and inverting addition problems. However, if this group has acquired problem-specific procedural knowledge, they will need to develop a new procedural for counting down the alphabet, as will the varied participants—who presumably strengthen their procedural knowledge during training.

Training results are similar to those for Experiment 1, so they are not reproduced here. Figure 3 shows that the transfer results are consistent with the predictions: the varied group requires considerably more time than the consistent group.

Taken together, Experiments 1 and 2 support the speed-up of both declarative knowledge retrieval and procedural knowledge application, as well as symmetric access to declarative knowledge and asymmetric access to procedural knowledge.

AN ACT-R MODEL

Act-R (Anderson, 1993) seems well suited for modeling these results, because it contains procedural and declarative long-term stores, along with learning mechanisms that alter the speed of elements in the two stores as a function of experience. Trafton (1996) has described an Act-R model for Experiment 1, but a bigger challenge is to construct a single Act-R model that can account for the results from both experiments. Such a model will serve three purposes. First, it will act as an additional test for several of Act-R's theoretical assumptions. Second, although each of Act-R's mechanisms has been tested in isolation, this model will test the interaction of several mechanisms. Third, the model will provide an explicit account of declarative and procedural learning and transfer that might then be used to analyze a wide range of more complex cognitive tasks.

The model presented here uses Act-R 4.0 (Anderson & Lebiere, in press).

Act-R is a parallel matching, serial firing rule-based system. It contains two long-term stores: procedural memory, represented by production rules, and declarative memory, represented by an associative network of declarative memory elements (DMEs). Working memory is viewed as the highly active portion of long-term declarative memory.

The alphabet arithmetic model has six production rules for the main goal. These are described in Table 1. READ-DISPLAY and ENCODE-DISPLAY simply read and look up the meaning of the textual symbols in the problem. REPORT-ANSWER reports the answer and signals that the goal has been achieved.

The remaining three rules—RETRIEVE-PLUS-RESULT, RETRIEVE-MINUS-RESULT, AND SUBGOAL-COUNT—are the most important rules in the model. RETRIEVE-PLUS-RESULT attempts to solve an addition problem by retrieving a fact from declarative memory that matches the problem, but also contains the answer. If successful, it uses the retrieved answer as the solution. RETRIEVE-MINUS-RESULT attempts to solve a subtraction problem by retrieving an addition DME that is the inverse of the subtraction problem. In other words, if the current problem is C-2=?, this rule will attempt to retrieve a fact of the form *letter* + 2 = C. SUBGOAL-COUNT creates a subgoal to solve the current problem by counting up or down the alphabet.

The model is designed so that Act-R will first try to retrieve an answer by using one of the retrieve rules. If the retrieval fails, then SUBGOAL-COUNT will fire to create the computation subgoal.

The model switches from computation to retrieval by acquiring declarative representations of problems that it has solved. When the model begins to solve problems it does not have any DMEs of past problems to retrieve, so it always uses SUBGOAL-COUNT. However, each time it solves a problem, it automatically remembers the problem and solution as a DME. These DMEs are then available for recall in future trials. Details of this memorization process are given below following the description of the computation subgoal.

The computation subgoal works by counting either up or down the alphabet. It uses a set of declarative memory elements that represent the alphabet using chunks thought to be common to people raised in United States:

ABCD EFG HIJK LMNOP QRS TUV WXYZ

Each chunk is a DME containing up to five letters and a pointer to the next chunk. For example, the second chunk in the alphabet (named alpha2) is represented as:

```
alpha2
      ISA item
      FIRST e
      SECOND f
      THIRD g
      NEXT alpha3
```

The subgoal contains 26 rules that implement counting forward and backward through the alphabet. To do this, it must first retrieve the alphabet chunk that contains the starting letter. Next it steps forward along the chunk until it finds the starting letter. Finally, it counts along the alphabet (either forward or backward) the required number of letters. If it reaches a chunk boundary, it must retrieve either the next or previous chunk before continuing the count.

The subgoal automatically produces a declarative memory trace of the problem and its solution. Goals in Act-R are DMEs that have been pushed onto the goal stack. You can think of a goal as a kind of goal-specific working memory, because it encodes the problem, the solution, and any partial results. When the subgoal has computed an answer, a rule pops the goal off of Act-R's goal stack. This removes the goal from the stack, but it remains in declarative memory as a DME representing the problem and its solution. For example, the DME representing A+2=C is:

```
Add-fact-10
      ISA problem
      ARG1 a
      OP plus
      ARG2 2
      COUNT 2
      RESULT c
```

Here, Add-fact-10 is an arbitrary name for the DME, and COUNT is used during processing to keep track of how many letters were counted.

Every time the subgoal solves a new problem, it leads to a new DME representing the problem and its solution. These DMEs are then available for retrieval by the two retrieval rules described above.

The model accounts for the experimental data by using three of Act-R's mechanisms: base-level learning, which speeds up access to commonly retrieved DMEs, strength learning, which speeds up rules that are commonly used, and the memory retrieval threshold, which prevents the retrieval of DMEs below a specified activation.

To understand how these mechanisms produce the speed-up and transfer shown in the data, you must first understand how Act-R predicts latencies. The total time for a trial in Act-R is the sum of the times needed to fire each production rule during that trial. The time to fire a rule is the sum of the time needed to retrieve the DMEs it matches plus the time to execute the rule's action. The time to retrieve a DME depends on its activation and the strength of the production rule that is retrieving it. Intuitively, latency of retrieval is inversely proportional to production strength and DME activation. The time to match DME *i* is given by Equation 1:

$$t_i = Fe^{-f(A_i+S_p)} \qquad \text{Equation 1}$$

Here, F and f are constants. *Ai* is the activation of DME *i*, and *Sp* is the strength of production *p*.

The activation of a DME is the sum of its base level activation and the spreading activation from other DMEs:

$$A_i = B_i + \sum_j W_j S_{ji} \qquad \text{Equation 2}$$

where *Bi* is the base level activation, *Wj* is the source activation of DME *j*, and *Sji* is the strength of association from *j* to *i*. A single unit of source activation is divided among all DMEs that fill slots of the current goal. For the present model, this means that elements of the current problem (i.e., the letter, operator, and number) will spread activation to DMEs representing past solutions.

Read-Display

IF the goal is to do an alphabet arithmetic problem, but the problem text has not yet been read
THEN read the problem text from the display

Encode-Display

IF the goal is to do an alphabet arithmetic problem, and the problem text has been read, but its meaning has not been determined
THEN encode the meaning of each textual symbol

Retrieve-Plus-Result

IF the goal is to do an alphabet ADDITION arithmetic problem of the form letter1 + number =, but the answer has not been determined, and there is a fact in memory stating that letter1 + number = letter2
THEN note letter2 as the answer

Retrieve-Minus-Result

IF the goal is to do an alphabet SUBTRACTION arithmetic problem of the form letter1 - number =, but the answer has not been determined, and there is a fact in memory stating that letter2 + number = letter1
THEN note letter2 as the answer

Subgoal-Count

IF the goal is to do an alphabet arithmetic problem, but the answer has not been determined
THEN set a subgoal to compute the answer by counting

Report-Answer

IF the goal is to do an alphabet arithmetic problem, and the answer has been determined
THEN report the answer and pop the goal

Table 1: The English version of the model's main production rules

For example, if the current goal is to solve A+2, then A will spread activation to all traces of previous problems that contain A either as the first letter or as the answer. The same is true for the operator and the number. Hence, the DME that represents the past solution to the current problem will receive activation from all three elements and will, most likely, be the most active DME.

The base level activation of a DME reflects the log prior odds that the DME will be matched by a production rule. Act-R assumes that these odds increase as a function of use and decrease as a function of delay. This is given by the optimized base-level learning equation.

$$B_i = \ln\left(\frac{nL^{-d}}{1-d}\right) + \beta \qquad \text{Equation 3}$$

where β represents the initial base-level, d is the decay rate, L is the time since the DME was created, and n is the number of times the DME has been used. This equation assumes that the uses of the DME are evenly spaced in time. This is a reasonable assumption for the present model, because each trial occurs only once in a given block. Act-R's exact base-level learning equation does not make this assumption, but is much more expensive to compute.

A use count of a DME is incremented whenever the DME is retrieved by a rule or when a duplicate DME is created. As noted above, when a goal is popped from the stack it remains in declarative memory. However, if Act-R detects that a newly created DME is identical to an existing DME, then it destroys the new DME and increments the use count of the old DME. This is important during initial skill acquisition, because a newly created DME might be too inactive to recall after a brief delay. When this happens, the model must recompute the answer. Since the subgoal creates a duplicate DME, the original DME is strengthened, increasing the chances of recall in future trials.

A DME that matches a rule's condition will be successfully retrieved whenever its activation exceeds the global retrieval threshold. Act-R assumes that DME activation contains permanent noise with mean 0 and variance σ_1^2. When a DME is first created, its base-level activation is set to a base level constant plus the permanent activation noise.

We can now see how the model might learn to retrieve declarative traces in the consistent training condition, but not in the varied training condition. In the consistent condition, the model is exposed to each problem 36 times. These frequent exposures boost the base-level activation of the memory traces, allowing the retrieval rules to directly recall the solutions. In contrast, in the varied condition the model is exposed to each problem only six times. In addition, the varied condition takes longer because the first 72 trials can only be solved by counting. In the consistent condition there is a chance of recalling one or more answers after the first 12 trials.

The speed-up of participants in the consistent condition is predicted by Equation 1, which governs retrieval latency. It predicts that retrieval latency is inversely proportional to activation and rule strength. Without considering rule strength we can see that an increase in DME activation will lead to lower predicted retrieval times and hence lower trial times in the consistent condition.

The model predicts that speed-up in the varied condition and part of the speed up in the consistent condition is due to speed-up of procedural knowledge. As discussed earlier in this section, Act-R assumes that the latency of a rule application is inversely proportional to its strength and the activation of the DMEs that it matches (see the discussion surrounding Equations 1 and 2). Rule strength is governed by the same equation that governs base-level learning (Equation 3) except that L is the time since the rule was created, d is a separate strength decay constant, and n is the number of times the rule has been fired.

Strength learning, combined with the latency equations (Equations 1 and 2), predict the speed-up in the varied condition and why varied training produces perfect transfer to new addition problems, whereas consistent training shows no transfer. In the varied condition, the model receives a lot of practice counting up the alphabet. Thus, the rules for counting, which are not specific to a single problem, are strengthened throughout training, and this strengthening continues during the transfer phase. In contrast, when the model is given consistent training, it learns to retrieve the answers to the 12 problems, so it rarely uses the counting rules. Once the model reaches the transfer phase it must begin to use the counting rules again, but their strengths will be either at or below their initial values, producing the dramatic slowdown observed in the data.

The model also accounts for the subtraction transfer results. In the consistent condition, the model acquires and strengthens DMEs representing each problem and its solution. When transferred to subtraction, these DMEs have a high enough activation to be retrieved and inverted by RETRIEVE-MINUS-RESULT. The model predicts that performance will be slower than at the end of training, because it has not yet strengthened RETRIEVE-MINUS-RESULT. In contrast, when the model is in the varied training condition, the DMEs rarely become active enough to retrieve, so they are not available during transfer. Although the model has strengthened its rules for counting up the alphabet, very few of these rules are used to count down, so the model must use counting down rules that have not yet been used, and hence are much slower to fire.

Four parameters were estimated to fit the model to the data. These were the base-level learning decay parameter (d in Equation 3), production strength decay parameter, retrieval threshold, and permanent activation noise. Transient noise was not used. These four parameters are critical to fitting the data. The rule strength decay parameter affects the learning rate of procedural knowledge. The interaction of the retrieval threshold with the three other parameters determines the amount of practice needed before the model can switch from computation to retrieval. To fit the data, these parameters must be set so that consistent training leads the model to retrieve the answers, whereas varied training leads the model to continue to compute the answers. In addition, the parameters must also produce the right learning

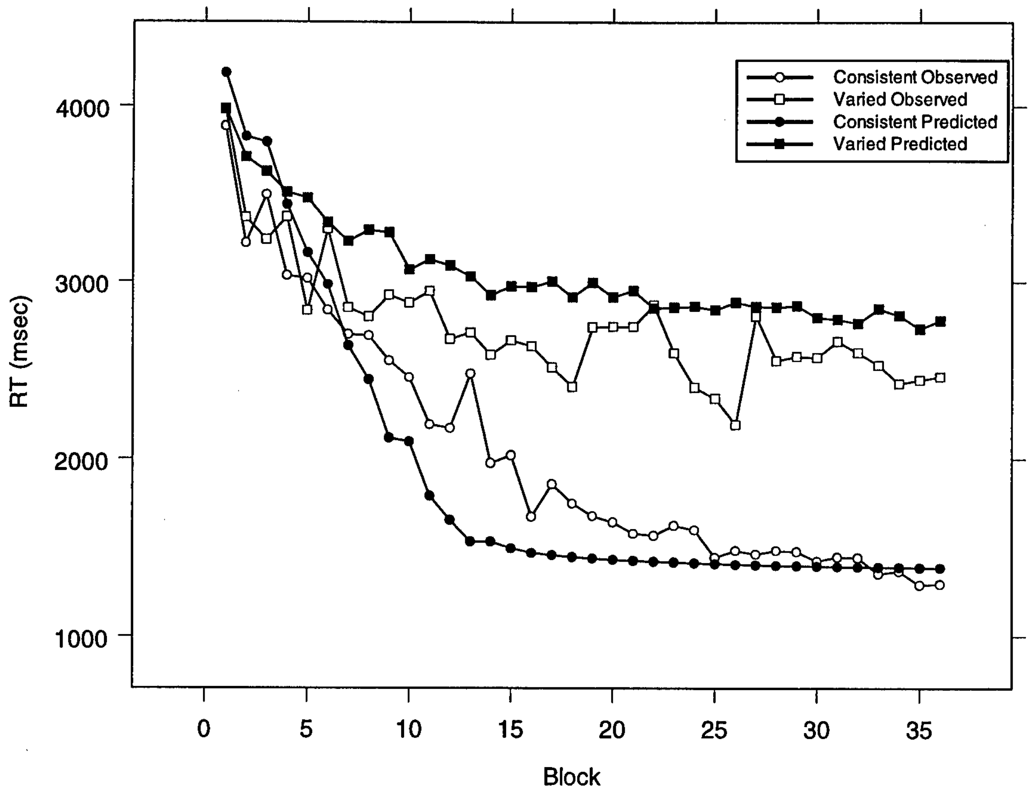


Figure 4: Observed and predicted mean response times during alphabet arithmetic training as a function of training group and practice block. Observed data replotted from Rabinowitz and Goldberg (1995).

curves for the two conditions.

The best fit was obtained with base-level learning decay set to .7, strength decay set to .5, retrieval threshold set to .55, and permanent activation noise variance set to .15. In addition, the total time to read the problem and type a letter was estimated at a constant 1.25 sec. This defines the lower bound of the model's response times. To reflect familiarity with the alphabet, all alphabet DMEs were given initial base-level activations of .974, reflecting 100 uses in the last 1000 seconds. Production rule strengths were initially set to .486, reflecting 25 uses in the past 1000 seconds. All other parameters used the default Act-R 4.0 values.

The model's predictions for the training phase in Experiment 1 are shown in Figure 4 along with the observed data. The model predictions were produced by simulating 15 subjects in each condition. The same model and parameter values were used for both conditions. The R^2 for the consistent condition was .89 and for the varied condition .78. This is pretty good considering that two different groups of subjects were modeled using the same parameters. In addition, the model captures the qualitative trends in the data—consistent simulations get much faster than varied simulations.

The transfer results are shown in Figures 5 and 6. The model closely fits the quantitative and qualitative results for alphabet addition transfer: consistent training leads to a large slow down in the transfer phase, whereas varied training results in perfect transfer. The subtraction transfer simulation matches the qualitative results, but not the quantitative ones: consistent training leads to better performance on subtraction than does varied training, but the model underestimates the latency of subtraction problems. Overall though, the fit is quite impressive, considering that four groups of subjects in four different conditions are fit using the same model and parameter values.

The modeling results raise several issues that will be addressed in the next section. The poor fit of the model to the quantitative subtraction data for the varied condition is easy to fix. It is possible to increase the time to compute a subtraction problem answer by either decreasing the strength of the subtraction counting rules or by switching to a different technique to solve the problems. A decrease in the rules' strengths is justifiable because most people rarely need to recite the alphabet backwards. However, it is also possible that people use a different strategy, such as guessing an answer and then counting forward to see if it is the right one.

The poor match to the subtraction latency in the consistent condition is much more puzzling. Specifically, why do the participants need over 4 seconds to solve each problem? If they are really recalling an alphabet addition problem and inverting it, then they should be closer to the predicted times, but instead their times are more than double the predictions. One possibility is that only a subset of varied participants actually switched to retrieval, whereas the remainder used computation.

The model's good fit to the data shows that active declarative knowledge is not needed to account for the results. Thus, the two experiments do not discriminate between declarative knowledge being inert or active.

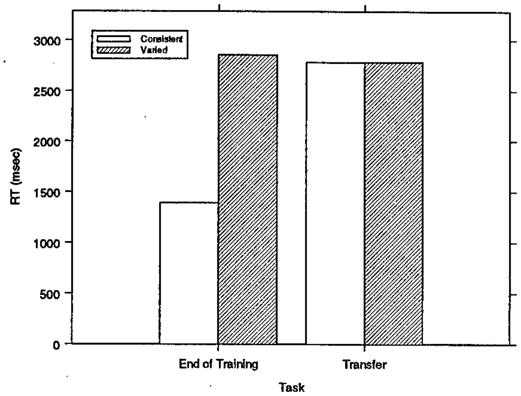


Figure 5: Mean predicted response times for Experiment 1 as a function of task and group.

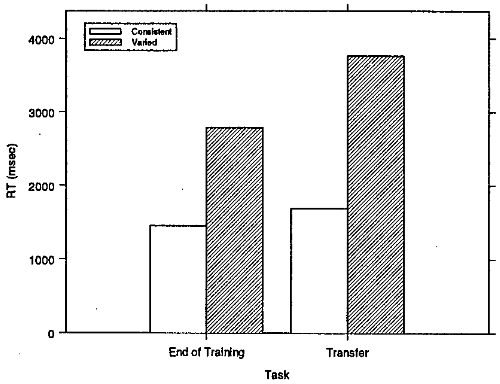


Figure 6: Mean predicted response times for Experiment 2 as a function of task and group.

However, it is possible that protocol data might provide evidence concerning this issue.

PROTOCOL ANALYSIS

To better understand the strategies that people use for alphabet arithmetic, particularly with respect to subtraction, a variant of Experiment 2 was run at The Ohio State University. Participants were 42 undergraduate students at The Ohio State University who received course credit for their effort. This experiment was similar to Rabinowitz and Goldberg's except that participants answered a questionnaire halfway through training and immediately after the transfer phase. Part 1 of the questionnaire contained the question: "Please describe all strategies that you used to solve the alphabet addition problems. If you used multiple strategies (or changed strategies), be as specific as possible about where and when you used them." Part 2 (completed at the end of the experiment) contained two questions: 1) "Please describe all strategies that you used to solve the alphabet **ADDITION** problems since the break. If you used multiple strategies (or changed strategies), be as specific as possible about where and when you used them." and 2) Please describe all strategies that you used to solve the alphabet **SUBTRACTION** problems. If you used multiple strategies (or changed strategies), be as specific as possible about where and when you used them."

Three main strategies were mentioned during the training phase: counting only, counting plus recall, and computing (in an unspecified way) plus recall. Many more strategies were mentioned in the transfer phase: counting backwards, recall plus inversion only, computing initially then switching to recall and inversion, and generate and test. Table 2 shows the results in terms of the percentage of participants in each category. For this analysis, responses to both training questions were coded together. The results clearly support the assumption that varied training leads to faster counting, whereas consistent training leads to direct retrieval. 95% of the participants in the consistent group reported using recall during training, versus only 32% of those in the varied condition. Most participants in the varied group (68%) reported that they used only counting throughout the entire training phase, in contrast to only 5% of participants in the consistent group.

The transfer protocol results are consistent with the hypothesis that varied training leads to strengthened asymmetrically accessible procedural knowledge for counting up, whereas consistent training leads to symmetrically accessible declarative knowledge. 70% of the consistent group reported recalling and inverting the addition problems, versus only 5% of the varied group. Likewise, only 15% of the consistent group reported counting back only, versus 36% of the varied group. Another 18% of the varied group used the generate and test strategy.

These results help clarify the model's problems of underestimating the difficulty of subtraction. First, they show that at least 15% of the consistent group used computation instead of recall, offering a possible explanation for the higher than predicted response times for this group on the transfer task. Second, the results indicate that the model's strategy of counting backward is consistent with the majority of participants in the varied group, but that the model is simply underestimating the time required to count back. In fact, two participants who used generate and test, mentioned that they switched to this method because counting back was too difficult. In contrast, counting back in the model within an alphabet chunk is just as fast as counting forward. The model's slower subtraction times are due only to the increased time needed to retrieve the previous chunk, thus subtraction problems that do not cross a chunk boundary are just as fast as addition problems. Resolving this problem should bring the model's predictions closer to the observed data.

The protocol data provides little evidence of whether declarative knowledge is inert or active. Only 10% of the consistent group mentioned computing the answers to a few subtraction problems before recognizing them as inverted addition problems.

CONCLUSION

This paper has three main results. The first is that the successful fit of the model to the alphabet arithmetic results shows that the two experiments fail to discriminate between active or inert declarative memory. Declarative memory in Act-R is inert—it can only be retrieved in the service of a production rule. Although the protocol data provided little insight into this issue, it does

Table 2: Reported strategy use based on training group and task.

	Condition	
	Consistent (n = 20)	**Varied** (n = 22)
Training		
Counting only	5 % (1)	68% (15)
Count + Recall	80% (16)	32% (7)
Compute + Recall	15% (3)	0%
Transfer		
Counting back only	15% (3)	36% (8)
Recall and Invert	60% (12)	5% (1)
Count back then recall and invert	5% (1)	0%
Compute then Recall and Invert	5% (1)	0%
Generate and Test	5% (1)	18% (4)
Count back + Generate and Test	0%	9% (2)
Other	5% (1)	5% (1)
Not codable	5% (1)	27% (6)

suggest that some kind of recognition process is needed before a participant can switch to recall and inversion. Recent work on feeling-of-knowing (i.e., the feeling that you know an answer to a problem) provides some support for this claim. Schunn, et al. (1997) have shown that feeling-of-knowing is based on similarity of the problem to previously seen problems, not on the availability of an answer to the problem. Since subtraction problems are so different from the inverted addition problems, it seems likely that solving one or two subtraction problems might lead to a feeling of knowing based on similarity between the solved subtraction problem and previously seen addition problems. This feeling-of-knowing might then prompt a person to consciously explore the similarities.

Second, the model's successful fit to the data and the protocol results provide additional support for separate declarative and procedural long-term memory stores. In addition, the model also shows that the separate strengthening of procedural and declarative knowledge can produce the observed results.

Finally, the paper shows that Act-R is sufficient to capture both the qualitative and quantitative details of the acquisition and transfer of procedural and declarative memory. Even more importantly, the model shows that several Act-R mechanisms working together can predict whether training will lead to procedural strengthening or the recall of declarative knowledge.

ACKNOWLEDGMENTS

Special thanks to Mitchell Rabinowitz for graciously supplying the experimental data presented in this paper, and to Hongbin Wang for his help during data collection. This research was supported in part by grants N00014-95-1-0241 and N00014-96-1-0472 from the Office of Naval Research, Cognitive and Neural Science and Technology Division.

REFERENCES

Anderson, J. R. (1993). *Rules of the Mind.* Hillsdale, NJ: Lawrence Erlbaum.

Anderson, J. R., & Lebiere, C. (in press). *The Atomic Components of Thought.* Hillsdale, NJ: Lawrence Erlbaum.

Larkin, J. H. (1981). Enriching formal knowledge: A model for learning to solve textbook physics problems. In J. R. Anderson (Ed.*), Cognitive Skills and Their Acquisition* (pp. 311-334). Hillsdale, New Jersey: Lawrence Erlbaum Assoc.

Miller, C. S. (1993). *Modeling Concept Acquisition in the Context of a Unified Theory of Cognition.* Unpublished PhD, Univ. of Michigan.

Newell, A. (1990). *Unified Theories of Cognition.* Cambridge, MA: Harvard University Press.

Newell, A., & Rosenbloom, P. S. (1981). Mechanisms of skill acquisition and the law of practice. In J. R. Anderson (Ed.), *Cognitive Skills and Their Acquisition* (pp. 1-55). Hillsdale, New Jersey: Lawrence Erlbaum.

Rabinowitz, M., & Goldberg, N. (1995). Evaluating the structure-process hypothesis. In F. E. Weinert & W. Schneider (Eds.), *Memory Performance and Competencies: Issues in Growth and Development* (pp. 225-242). Hillsdale, NJ: Lawrence Erlbaum.

Rosenbloom, P. S., & Aasman, J. (1990). Knowledge Level and Inductive Uses of Chunking (EBL). Paper presented at the *Proceedings of the Eighth National Conference on Artificial Intelligence.*

Schunn, C. D., Reder, L. M., Nhouyvanisvong, A., Richards, D. R., & Stroffolino, P. J. (1997). To calculate or not to calculate: A Source activation confusion model of problem familiarity's role in strategy selection. *Journal of Experimental Psychology: Learning, Memory, and Cognition,* 23(1), 3-29.

Steier, D. M., Laird, J. E., Newell, A., Rosenbloom, P. S., Flynn, R. A., Golding, A., Polk, T. A., Shivers, O. G., Unruh, A., & Yost, G. R. (1987). Varieties of learning in Soar: 1987. In P. S. Rosenbloom, J. E. Laird, & A. Newell (Eds.), *The Soar Papers: Research on Integrated Intelligence* (Vol. 1, pp. 537-548). Cambridge, Mass: MIT Press.

Stillings, N. A., Weisler, S. E., Chase, C. H., Feinstein, M. H., Garfield, J. L., & Rissland, E. L. (1995). *Cognitive Science: An Introduction.* (Second ed.). Cambridge, MA: MIT Press.

Trafton, J. G. (1996). Alphabet arithmetic and Act-R: A reply to Rabinowitz and Goldberg, *Proceedings of the Eighteenth Annual Conference of the Cognitive Science Society* . Mahwah, NJ: Lawrence Erlbaum Associates.

VanLehn, K. (1989). Problem solving and cognitive skill acquisition. In M. I. Posner (Ed.), *Foundations of Cognitive Science* (pp. 527-579). Cambridge, MA: MIT Press.

VanLehn, K. (1996). Cognitive skill acquisition. *Annual Review of Psychology,* 47, 513-539.

Skill Learning Using A Bottom-Up Hybrid Model

Ron Sun, Edward Merrill, Todd Peterson
The University of Alabama
Tuscaloosa, AL 35487, USA
rsun@cs.ua.edu

Abstract

This paper presents a skill learning model CLARION. Different from existing models of mostly high-level skill learning that use a top-down approach (that is, turning declarative knowledge into procedural knowledge), we adopt a bottom-up approach toward low-level skill learning, where procedural knowledge develops first and declarative knowledge develops from it. CLARION which follows this approach is formed by integrating connectionist, reinforcement, and symbolic learning methods to perform on-line learning. We compare the model with human data in a minefield navigation task. A match between the model and human data is observed in several comparisons.

1 Introduction

Skills vary in complexity and the degree of cognitive involvement. They range from simple motor movements and other routine tasks in everyday activities to high-level intellectual skills. We want to study "lower-level" cognitive skills, which have not received sufficient research attention. One type of task that exemplifies what we call low-level cognitive skill is reactive sequential decision making (Sun and Peterson 1995). It involves an agent selecting and performing a sequence of actions to accomplish an objective on the basis of moment-to-moment information (hence the term "reactive"). An example of this kind of task is the minefield navigation task developed at The Naval Research Lab (see Gordon et al. 1994). This kind of task setting appears to tap into real-world skills associated with decision making under conditions of time pressure and limited information. Thus, the results we obtain from human experiments will likely be transferable to real-world skill learning situations. Yet this kind of task is suitable for computational modeling given the recent development of machine learning techniques (Sun et al 1996, Watkins 1989).

The distinction between procedural knowledge and declarative knowledge has been made in many theories of learning and cognition (for example, Anderson 1982, 1993, Keil 1989, Damasio et al. 1994, and Sun 1995). It is believed that both procedural and declarative knowledge are essential to cognitive agents in complex environments. Anderson (1982) originally proposed the distinction based on data from a variety of skill learning studies, ranging from arithmetic to geometric theorem proving, to account for changes resulting from extensive practice. Similar distinctions have been made by other researchers based on different sets of data, in the areas of skill learning, concept formation, and verbal informal reasoning (e.g., Fitts and Posner, 1967; Keil, 1989; Sun, 1995).

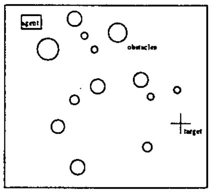


Figure 1: Navigating Through Mines

Most of the work in skill learning that makes the declarative/procedural distinction assumes a top-down approach; that is, learners first acquire a great deal of explicit declarative knowledge in a domain and then through practice, turn this knowledge into a procedural form ("proceduralization"), which leads to skilled performance. However, these models were not developed to account for skill learning in the absence of, or independent from, prexisting explicit domain knowledge. Several lines of research demonstrate that individuals can learn to perform complex skills without first obtaining a large amount of explicit declarative knowledge (e.g., Berry and Broadbent 1988, Stanley et al 1989, Lewicki et al 1992, Willingham et al 1992, Reber 1989, Karmiloff-Smith 1986, Schacter 1987, and Schraagen 1993). In research on *implicit learning*, Berry and Broadbent (1988), Willingham et al (1992), and Reber (1989) expressly demonstrate a *dissociation* between explicit knowledge and skilled performance in a variety of tasks including dynamic decision tasks (Berry and Broadbent 1988), artificial grammar learning tasks (Reber 1989), and serial reaction tasks (Willingham et al 1992). Berry and Broadbent (1988) argue that the psychological data in dynamic decision tasks are not consistent with exclusively top-down learning models, because subjects can learn to perform the task without being provided a priori declarative knowledge and without being able to verbalize the rules they used to perform the task. This indicates that procedural skills are not necessarily accompanied by explicit declarative knowledge, which would not be the case if top-down learning is the only way to acquire skill. Willingham et al (1989) similarly demonstrate that procedural knowledge is not *always* preceded by declarative knowledge in human learning, and show that declarative and procedural learning are not necessarily correlated. There are even indications that explicit knowledge may arise from procedural skills in some circumstances (see Stanley et al 1989). Using a dynamic decision task, Stanley et al. (1989) found that the development of declarative knowledge paralleled but lagged behind the development of procedural knowledge.

Similar claims concerning the development of procedural knowledge prior to the development of declarative knowledge have surfaced in a number of research areas outside the skill learning literature and provided additional support for the bottom-up approach. *Implicit memory* research (e.g., Schacter 1987) demonstrates a dissociation between explicit and implicit knowledge/memories in that an individual's performance can improve by virtue of implicit "retrieval" from memory and the individual can be unaware of the process. This is not amenable to the exclusively top-down approach. *Instrumental conditioning* also reflects a learning process that differs from the top-down approach, because the process is typically non-verbal and involves the formation of action sequences without requiring a priori explicit knowledge. It may be applied to simple organisms as well as humans (Gluck and Bower 1988). In *developmental psychology*, Karmiloff-Smith (1986) proposed the idea of "representational redescription". During development, low-level implicit representations are transformed into more abstract and explicit representations and thereby made more accessible. This process is not top-down either, but in the opposite direction.

2 The Model

The difference between declarative and procedural knowledge leads naturally to "two-level" architectures (Sun 1995). We thus developed the model CLARION, which stands for *Connectionist Learning with Adaptive Rule Induction ON-line* (Sun et al 1996). It embodies the distinction of declarative and procedural knowledge (or, conceptual and subconceptual knowledge), and it performs learning in a bottom-up direction. It consists of two main components: the top level encodes explicit declarative knowledge in the form of propositional rules, and the bottom level encodes implicit procedural knowledge in neural networks. In addition, there is an episodic memory, which stores recent experiences in the form of "input, output, result" (i.e., stimulus, response, and consequence).

A high-level pseudo-code algorithm that describes CLARION is as follows:

```
1. Observe the current state $x$.
2. Compute in the bottom level the Q-value of each of the possible actions ($a_i$'s) associated with the perceptual state $x$: $Q(x,a_1)$, $Q(x,a_2)$, ......, $Q(x,a_n)$.
3. Find out all the possible actions ($b_1$, $b_2$, ...., $b_m$) at the top level, based on the the perceptual information $x$ and other available information (which goes up from the bottom level) and the rules in place at the top level.
4. Compare the values of $a_i$'s with those of $b_j$'s (which are sent down from the top level), and choose an appropriate action $a$.
5. Perform the action $a$, and observe the next state $y$ and (possibly) the reinforcement $r$.
6. Update the bottom level in accordance with the Q-Learning-Backpropagation algorithm, based on the feedback information.
7. Update the top level using the Rule-Extraction-Refinement algorithm.
8. Go back to Step 1.
```

In the bottom level, a Q-value is an evaluation of the "quality" of an action in a given state: $Q(x,a)$ indicates how desirable action a is in state x. We can choose an action based on Q-values. To acquire the Q-values, supervised and/or reinforcement learning methods may be applied. A widely applicable option is the *Q-learning* algorithm (Watkins 1989), a reinforcement learning algorithm. In the algorithm, $Q(x,a)$ estimates the maximum discounted cumulative reinforcement that the agent will receive from the current state x on. The updating of $Q(x,a)$ is based on minimizing $r+\gamma e(y)-Q(x,a)$, where γ is a discount factor and $e(y) = \max_a Q(y,a)$. Thus, the updating is based on the *temporal difference* in evaluating the current state and the action chosen: In the above formula, $Q(x,a)$ estimates, before action a is performed, the (discounted) cumulative reinforcement to be received if action a is performed, and $r + \gamma e(y)$ estimates the (discounted) cumulative reinforcement that the agent will receive, after action a is performed; so their difference (the temporal difference in evaluating an action) enables the learning of Q-values that approximate the (discounted) cumulative reinforcement. Using Q-learning allows sequential behavior to emerge in an agent. Through successive updates of the Q function, the agent can learn to take into account future steps in longer and longer sequences.

To implement Q functions, we chose to use a four-layered network (see Figure 2), in which the first three layers form a (either recurrent or feedforward) backpropagation network for computing Q-values and the fourth layer (with only one node) performs stochastic decision making. The output of the third layer (i.e., the output layer of the backpropagation network) indicates the Q-value of each action (represented by an individual node), and the node in the fourth layer determines probabilistically the action to be performed based on a Boltzmann distribution (i.e., Luce's choice axiom; Watkins 1989). This learning process performs both structural credit assignment (with backpropagation), so that the agent knows which element in a state should be assigned credit/blame, as well as temporal credit assignment, so that the agent knows which action leads to success or failure. This learning process enables the development of procedural skills potentially solely based on the agent independently exploring a particular world on a continuous and on-going basis.

In the top level, declarative knowledge is captured in a simple propositional rule form. To facilitate correspondence with the bottom level and to encourage uniformity and integration (Clark and Karmiloff-Smith 1993), we chose to use a localist connectionist model for implementing these rules (e.g., Sun 1992, Towell and Shavlik 1993). Basically, we translate the structure of a set of rules into that of a network. For each rule, a set of links are established, each of which connects a node representing a concept in the condition of a rule to the node representing the conclusion of the rule. For more complex rule forms including predicate rules and variable binding, see Sun (1992).

To fully capture bottom-up learning processes, we devised an algorithm for learning declarative knowledge (rules) using information in the bottom level (the *Rule-Extraction-Refinement* algorithm). The basic idea is as follows: if an action decided by the bottom level is successful then the agent extracts a rule (with its action corresponding to that selected by the bottom level and with its conditions corresponding to the current sensory state), and adds the rule to the top-level rule network. Then, in subsequent interactions with the world, the agent refines the extracted rule by considering the outcome of applying the rule: if the outcome is successful, the agent may try to generalize the conditions of the rule to make it more universal; if the outcome is not successful, then the conditions of the rule should be made more specific and exclusive of the current case.

We perform rule extraction at each step, based on the following information: (x, y, r, a), where x is the state before action a is performed, y is the new state entered after an action a is performed, and r is the reinforcement received after action a. Rules are in the following form: *conditions* $\longrightarrow$ *action*, where the left-hand side is a conjunction of individual conditions each of which refers to the value of an element in the (sensory) input state. Three different criteria can be used for rule learning at each step: (1) direct reinforcement received at a step, (2) temporal difference (as used in updating Q-values), and (3) maximum Q-values in a state. We adopt a three-phase approach, with each phase lasting for a certain number of episodes. Phase transition can be automatically determined based on the current performance level of the model. At each step, we apply the current-phase criterion to determine whether we should construct a rule. If so, a rule is wired up in the rule network. After rules are extracted, at each step, the algorithm reexamines the rules matching the current step to decide if each of them should be kept, revised, or discarded. See Sun et al. 1996 for the full details of rule learning.

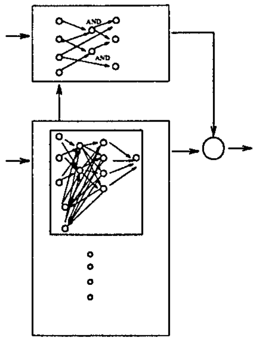


Figure 2: The implementation of CLARION.

Step 4 is for making the final decision on which action to take by incorporating outcomes from both levels. We combine the corresponding values for an action from the two levels by a weighted sum; that is, if the top level indicates that action a has an activation value v (which should be 0 or 1 as rules are binary) and the bottom level indicates that a has an activation value q (the Q-value), then the final outcome is $w_1 * v + w_2 * q$. Stochastic decision making with Boltzmann distribution (based on the weighted sums) is then performed. Figure 2 shows the two levels of the model.

3 Experiments

In all of the human experiments, subjects were seated in front of a computer monitor that displayed an instrument panel containing several gauges that provided current information (see Figure 3). The following instruction was given to explain the setting:

> I. Imagine yourself navigating an underwater submarine that has to go through a minefield to reach a target location. The readings from the following instruments are available:
>
> (1) Sonar gauges show you how close the mines are to the submarine. This information is presented in 8 equal areas that range from 45 degrees to your left, to directly in front of you and then to 45 degrees to your right. Mines are detected by the sonars and the sonar readings in each of these directions are shown as circles in these boxes. A circle becomes larger as you approach mines in that direction.
>
> (2) A fuel gauge shows you how much time you have left before you run out fuels. Obviously, you must reach the target before you run out of fuel to successfully complete the task.
>
> (3) A bearing gauge shows you the direction of the target from your present direction; that is, the angle from your current direction of motion to the direction of the target.
>
> (4) A range gauge shows you how far your current location is from the target.
>
> II. At the beginning of each episode you are located on one side of the minefield and the target is on the other side of the minefield. You task is to navigate through the minefield to get to the target before you run out of fuel. An episode ends when: (a) you get to the goal (success); (b)

you hit a mine (failure); (c) you run out of fuel (failure).

A random mine layout was generated for each episode. This setting is *stochastic* and *non-Markovian.* Five training conditions were used:

- The standard training condition. Subjects received five blocks of 20 episodes on each of five consecutive days (100 episodes per day). In each episode the minefield contained 60 mines. The subjects were allowed 200 steps.
- The verbalization training condition. This condition was identical to the standard training condition except that subjects were asked to step through slow replays of selected episodes and to verbalize what they were thinking during the episode. Subjects received replays on the first, third, and fifth days of training. The subjects were replayed five episodes after the first block of 20 episodes and five episodes after the fifth block of 20 episodes on these days.
- The over-verbalization training condition. In this condition subjects were presented replays of 15 of their first 25 episodes, and asked to verbalize during the slow playback. Replay of an episode occurred immediately after the subject finished the episode.
- The 30-to-60 transfer condition. This condition was also identical to the standard training condition except that subjects performed the task with 30 mines on the first two days of training and switched to 60 mines starting the third day.
- The mixed training condition. "Mixed" refers to the fact that mine density was manipulated during training. Subjects performed the task with 30, 50, 70, or 90 mines. Subjects received eight blocks of 10 episodes per day over five days, two at each mine density. Order of presentation was randomized.

In CLARION each gauge was represented by a set of nodes that corresponded to what human subjects would see on screen. This input setup yielded a total of 43 primary perceptual inputs. Thus, there were more than 10^{12} possible input states. Thus the model had to deal with the problem of high dimensionality. As a result, a lookup table implementation for Q-learning at the bottom level was not possible (Tesauro 1992, Lin 1992). To deal with the situation. a functional approximator such as backpropagation networks must be used. Also in correspondence to the human experimental setting, the action outputs consisted of two clusters of nodes representing turn and speed.

The model started out with no more a priori knowledge about the task than a typical human subject, so that bottom-up learning can be captured. The bottom level contained randomly initialized weights (with a pre-chosen, fixed topology). The top level started empty and contained no a priori knowledge about the task, either in the form of instructions or instances. The episodic memory was empty at the beginning. There was no supervised learning (i.e., no teacher input). The reinforcement signals embodied some a priori notions regarding getting close to target and avoiding explosion that were also provided to human subjects through instructions. The learning algorithm with all the requisite parameters was pre-set, presumably reflecting the learning mechanisms in humans.

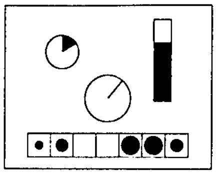


Figure 3: The Navigation Input

The display at the upper left corner is the fuel gauge; the vertical one at the upper right corner is the range gauge; the round one in the middle is the bearing gauge; the 7 sonar gauges are at the bottom.

The results of the experiments are analyzed as follows.

The standard training condition. We obtained performance data over 500 episodes per subject. We averaged the data over 10 human subjects. We did the same with the model: Each model run was initialized with different random number sequences and thus produced different results; we averaged 10 such runs in exact correspondence with human experiments (i.e., we did not tune the random number sequences to generate a match, but randomly set seeds for random number generators, analogous to random selection of human subjects in this experiment). We compared *average* success rates because in this way we can eliminate the uninteresting impact of individual differences and instead focus on essential features of learning in this task. These data are presented in Figure 4. Both sets of data were best fit by power functions (for failure rate). The degree of similarity is evident. A Pearson product moment correlation coefficient was calculated (treating blocks as individuals and human versus model as the X and Y variables). The analysis yielded a high positive correlation (r = .82), indicating a high degree of similarity between human subjects and model runs.

The verbalization training condition. Obviously, we could not require verbalization from the model. However, we posited that much of the effect of verbalization on learning was associated with rehearsing previous steps and episodes (although there may be additional factors involved). Thus for the model, we used episode memory playback (Lin 1992) in a first attempt to capture this effect. Episode memory playback involves training the model with previously performed episodes between blocks of actual trial episodes in exactly the same manner as in human experiments. In this case, the data from 5 human subjects was com-

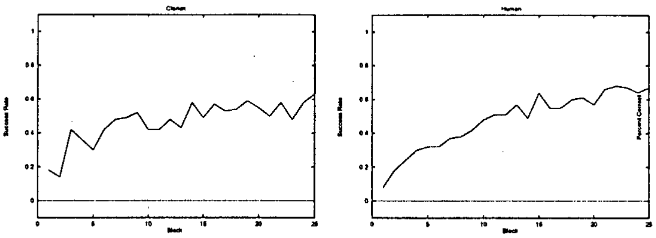


Figure 4: The learning curves in terms of success rates in the standard condition. The right side is the human data and the left side is the model data.

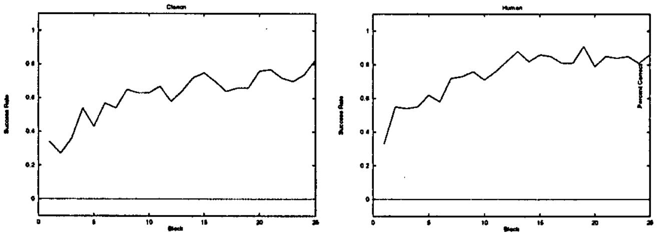


Figure 5: The learning curves in terms of success rates in the verbalization condition.

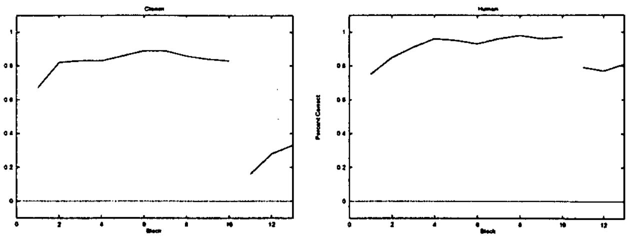


Figure 6: The 30-to-60 transfer data in terms of success rates.

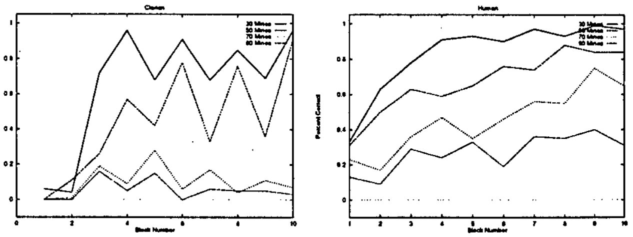


Figure 7: Average success rates for each mine densities in the mixed condition.

pared to that of 5 model runs. Data was averaged for each of 25 blocks (see Figure 5). Again, both sets of data were highly similar and both were best fit by power functions. We also calculated a Pearson product moment correlation coefficient, which yielded a high positive correlation (r = .84).

We subsequently compared the changes in performance due to verbalization for the human subjects and the model runs. This was done by averaging failure rates across blocks separately for each human subject and for each model run and subjecting that data to a 2 × 2 ANOVA. The analysis of these data indicated the both groups exhibited a significant increase in performance due to verbalization ($p < .01$), and that the changes due to verbalization for the two groups were not significantly different (52 to 25 percent failure rate for the human subjects versus 53 to 38 percent failure rate for the model runs). The effect of explication of implicit knowledge which likely results from verbalization was captured through the usual rule learning process, which was also at work during episode replay.

The 30-to-60 transfer condition. Subjects were first trained on 30-mine minefields, and then transferred to 60-mine minefields. The model was tested under the same condition. Both human and model data were averaged over 10 subjects. Comparing the human and model data (see Figure 6), we noticed that both learned well at 30 mines, although human data was slightly better. When transferred to 60 mines, both exhibited a significant drop in performance, although the model exhibited a deeper drop. Specifically, we compared performance of the last block before the change in mine density and the first block after the change. Success rates were 98% and 79% for the human subjects and 83% and 26% for the model runs respectively. The drops were both statistically significant. At first look, it might appear that the drop in performance for the model runs was much greater than that for the human subjects. However, this might not be a fair assessment in that we did not allow the model runs to reach the same performance as the human subjects before changing the mine density. Indeed, the 5 highest performing of the model runs before the change performed 8 times better after the change than did the 5 lowest performing ones.

The mixed training condition. We plotted learning curves in terms of success rates for each mine density separately. The data were averaged over 8 human subjects and 8 model runs, respectively. The average curves are shown in Figure 7. We calculated overall success rates for each of the mine densities. Both the human subjects and model runs performed best with the lowest mine density and performance decreased with each increase in the number of mines. Thus, we observed a similar pattern. The drop in performance was roughly the same for human subjects and model runs between the 30 and 50 mine densities (16% versus 13%, respectively). We do not know for sure what accounts for the failure of the model at the 70 and 90 mine densities. However, questionnaires completed by the human subjects indicated that they treated the higher density conditions as different from the lower density conditions. Because the model runs did not "start over" at each density, they were applying what was learned to conditions in which it did not work. In contrast, human subjects could sense the change in conditions and discard their old strategies.

The over-verbalization condition. Human subjects under the over-verbalization condition failed to learned. During the 25 episodes of training, their success rates were well below 10%, compared with the

33% performance for the subjects under the (sparse) verbalization condition. If we eliminate one subject who performed at 60%, the remaining subjects only achieved approximately 3% success rate. CLARION accounts for this phenomenon by positing that too much verbalization (e.g., verbalizing for more than half of the training episodes) caused the learner to switch to a completely explicit mode of learning; they tended to rely completely on the top-level learning mechanism and shut down the bottom level. This is consistent with the similar hypothesis by Stanley et al (1989), for explaining their findings regarding the difficulty their subjects had in learning a dynamic decision task after being given instructions that encouraged them to be explicit. Schooler et al (1993) also reported that requiring verbalization impaired subjects' ability to solve problems that require "insight", by forcing them to be overly explicit. CLARION explains the findings readily with the shut-down mechanism. The top-level learning mechanism when disconnected from the bottom level, clearly has trouble learning this kind of sequential task, because of its lack of a temporal credit assignment process (comparable in power to Q-learning) and its all-or-nothing learning process. On the other hand, in the bottom level, the distributed network representation and learning process that incorporates gradedness and temporal information handle complex sequences well.

Verbalization segments indicating bottom-up learning. The verbalization data we collected from the subjects (under the verbalization training condition) were consistent, in an informal sense, with our assumption of bottom-up learning being prominent in this task setting, as exemplified by the following segments.

> S: I thought about it after I started doing it. I said, look at me .... look what I'm doing. I didn't start thinking about it until I started doing it. I figured out that it started helping me and that's when I started doing it myself. (subj.38)
>
> S: When I started off ...... I didn't understand at all .... I couldn't grasp the whole sonar concept at all. (subj.38)
>
> S: So, basically what I do – not thinking about driving a submarine or mine. (subj.38)
>
> S: When you get in a situation like this, where there are gaps, it's purely instinctual. (subj.37)
>
> S: That's pretty much I've done the whole game [being instinctual], with the exception of a couple of patterns I've started to recognize. (subj.37)

In sum, the verbalization by the subjects suggested that some degree of bottom-level (implicit) learning/decision making and gradual bottom-up learning existed. This is the kind of learning CLARION was meant to capture.

We also compared the verbalizations of good performers (subjects) vs. poor performers. Our analysis indicated a lack of difference: we failed to notice any significant difference across a variety of measures (such as length of verbalization, detailedness, and types of statements uttered). We suggest that this is one more piece of evidence that indicates the importance/prominence of bottom-level (implicit) learning: The performance is mostly determined by implicit procedural learning, which cannot be easily verbalized, while verbalized explicit knowledge is nonspecific and has relatively minor impact during learning.

4 Conclusions

In sum, we discussed a hybrid connectionist model CLARION as a demonstration of the approach of bottom-up skill learning, which consists of two levels for capturing both procedural and declarative knowledge and performing bottom-up learning. Some degree of match with human data was found across a number of different experimental conditions.

Acknowledgements

This work is supported in part by Office of Naval Research grant N00014-95-1-0440. The simulator was provided by NRL.

References

J. R. Anderson, (1982). Acquisition of cognitive skill. *Psychological Review.* Vol.89, pp.369-406.

D. Berry and D. Broadbent, (1988). Interactive tasks and the implicit-explicit distinction. *British Journal of Psychology.* 79, 251-272.

A. Clark and A. Karmiloff-Smith, (1993). The cognizer's innards: a psychological and philosophical perspective on the development of thought. *Mind and Language.* 8 (4), 487-519.

A. Damasio, (1994). *Decartes' Error.* Grosset/Putnam, NY.

D. Gordon, et al. (1994). *User's Guide to the Navigation and Collision Avoidance Task.* Naval Research Lab. DC.

H. Dreyfus and S. Dreyfus, (1987). *Mind Over Machine: The Power of Human Intuition,* The Free Press, New York, NY.

P. Fitts and M. Posner, (1967). *Human Performance.* Brooks/Cole, Monterey, CA.

M. Gluck and G. Bower, (1988). From conditioning to category learning. *Journal of Experimental Psychology: General.* 117 (3), 227-247.

W. James, (1890). *The Principles of Psychology.* Dover, NY.

A. Karmiloff-Smith, (1986). From meta-processes to conscious access. *Cognition.* 23. 95-147.

F. Keil, (1989). *Concepts, Kinds, and Cognitive Development.* MIT Press. Cambridge, MA.

P. Lewicki, et al. (1992). Nonconscious acquisition of information. *American Psychologist.* 47, 796-801.

L. Lin, (1992). Self-improving reactive agents based on reinforcement learning, planning, and teaching. *Machine Learning.* Vol.8, pp.293-321.

A. Reber, (1989). Implicit learning and tacit knowledge. *Journal of Exp Psychology: General.* 118 (3), 219-235.

J. Schooler, S. Ohlsson, and K. Brooks, (1993). Thoughts beyond words: when language overshadows insight. *Journal of Experimental Psychology: General.* 122 (2). 166-183.

D. Schacter, (1987). Implicit memory: History and current status. *Journal of Experimental Psychology: Learning, Memory, and Cognition,* 13, 501-518.

R. Shiffrin and W. Schneider, (1977). Controlled and automatic human information processing II. *Psychological Review.* 84. 127-190.

P. Smolensky, (1988). On the proper treatment of connectionism. *Behavioral and Brain Sciences,* 11(1):1–74.

W. Stanley, et al (1989). Insight without awareness *Quarterly Journal of Experimental Psychology.* 41A (3), 553-577.

R. Sun, (1992). On Variable Binding in Connectionist Networks, *Connection Science,* Vol.4, No.2, pp.93-124.

R. Sun, (1995). Robust reasoning: integrating rule-based and similarity-based reasoning. *Artificial Intelligence.* 75, 2. 241-296.

R. Sun, T. Peterson, and E. Merrill, (1996). Bottom-up skill learning. Technical Report TR-CS-96-0021, University of Alabama. shortened version in *Proc.of 18th Cognitive Science Society Conference,*

T. Tesauro, (1992). Practical issues in temporal difference learning. *Machine Learning.* Vol.8, 257-277.

C. Watkins, (1989). *Learning with Delayed Rewards.* Ph.D Thesis, Cambridge University, Cambridge, UK.

D. Willingham, et al (1989). On the development of procedural knowledge. *Journal of Experimental Psychology: Learning, Memory, and Cognition.* 15, 1047-1060.

Modelling Memory-Updating Characteristics of 3- and 4-Year Olds

C. Philip Beaman
John Morton
MRC Cognitive Development Unit
4 Taviton Street
London WC1H 0BT, UK
+44 171 387 4692
P.Beaman@cdu.ucl.ac.uk
J.Morton@cdu.ucl.ac.uk

ABSTRACT

In this paper a memory perspective on young children's performance at a particular false belief task, the Smarties task, is described. The theoretical analysis focuses on the computational conditions that are required to resolve the Smarties task, on the possible limitation in the developing memory system that may lead to a computational breakdown resulting in a failure to resolve, and on ways of bypassing such limitations to ensure correct resolution. A symbolic model of this analysis implemented using the COGENT modelling environment is described, and its fit to the data considered.

Keywords

Developmental modelling, false belief, memory updating, COGENT

INTRODUCTION

One of the many constraints identified by Newell (1990) on any form of cognitive architecture which attempts to model human cognition is that it should be capable of arising from earlier forms by a process of developmental maturation. Developmental constraints, and discrete developmental stages, have received surprisingly little attention from symbolic modellers, although questions of how a mature system might develop from a relatively simple template are now being considered within the connectionist research program (e.g., Elman et al., 1996). The present study considers a developmental stage believed to be crucial to the maturation of memory processes, and aims to demonstrate how the failure of 3- and 4-year olds at a task which adults find trivially easy (the Smarties task; Perner, Leekam & Wimmer, 1987) can be modelled using a destructive-updating process. A subtle alteration of the memory encoding characteristics of this task enables 3- and 4-year olds to perform the task correctly. The patterns of children's performances are modelled as discrete developmental stages using the COGENT (Cognitive Objects in a Graphical EnvironmeNT) modelling environment of Cooper and Fox (in press).

The Smarties Task

The basic procedure for the Smarties task is as follows. The subjects are shown a tube of Smarties (a popular brand of sweet) and asked what the tube contains. Children of around the age of four are usually both able and willing to provide an answer to this question. The top is then taken off the tube, and its contents are shown to the child. The contents of the tube are pencils rather than the anticipated Smarties. The top is then replaced on the tube, and the child is asked two questions, the reality question (what is in the tube?) and the belief question (when you first saw the tube, what did you think was in the tube?). Typically, 70% of 3-year-old children who are able to answer the first question correctly (pencils) now also give the same answer to the second question.

A Memory-Updating Explanation

The original form of the Smarties task implies some peculiar memory characteristics. Children who fail this task are incorrectly reporting a belief which they had held, and told to the experimenter, only seconds previously. Although a conceptual deficit, an inability to comprehend false belief, can be put forward to explain these results, it seems strange to suppose that this deficit manifests itself in the child's inability to correctly recall the contents of this belief, even though they were able to report to the experimenter what the contents of this belief were immediately before it was shown to be false. Instead, it is argued (Barreau, 1997; Morton, 1997) that the child's inability is centred around a memory updating system, such that the false belief (that the tube contains Smarties) is never encoded as a stable, long-term representation, and so is immediately supplanted by the incoming information that the tube contains pencils. Thus, when such children are asked the belief question, the only source of information available to them is the representation of the current state of reality: *in(tube, pencils)*.

The Bag Experiment.

A variation on this experimental procedure designed to maximise the possibility that the contents of the tube are translated into a long-term format is described by Barreau (1997). Immediately after showing the tube to the child, and asking the child what they believed the tube to contain, the contents of the tube were emptied into a bag. Although the child witnessed this operation, at no time were they able to see the contents of the tube either at first or during the transfer. The tube was then shown to the child to demonstrate that it was empty, and then ostentatiously hidden from view. The child is then asked what they believe to be in the bag. All children replied "Smarties". The contents of the bag were then shown to the child. In this case, the bag contained marbles, rather than Smarties. The child was then asked five questions concerning the contents of the bag and the tube:

1. Before I opened the bag, what did you think was in the bag? (BAG:BELIEF: PAST)
2. What is really in the bag? (BAG:REALITY: PRESENT)
3. When I first showed you the tube, what did you think was in the tube? (TUBE: BELIEF: PAST)
4. What is inside the tube now? (TUBE: REALITY: PRESENT)
5. What was really inside the tube? (TUBE: REALITY: PAST)

In Barreau's (1997) experiment, twenty-four children were questioned in this manner, the results of this experiment are shown in the table below:

TABLE 1: Table of answers to the tube and bag questions.

Questions	Correct	Reversed	Double
BAG	8	8	8
TUBE	15	3	6

In order to be scored correct, both the bag questions, (belief and reality) had to be correctly answered. To be scored correct in the tube condition, the belief questions and at least one of the reality questions had to be correctly answered. A "double" score refers to a repeat answer, i.e. a reality response to a belief question. This category also includes one child who gave belief answers to reality questions. The reversed response indicates a reversal between the belief and reality answers in the bag questions, and the belief and one of the reality answers in the tube questions.

The assumptions underlying this experiment were that when the tube was removed from view, the tube→ bag transferral episode would be coded as ended, and details of the whole episode would be translated into long-term memory. Thus, when the current representation of the bag's contents is updated, the representation of the tube's contents will be invulnerable.

The data has also been analysed as suggesting that three qualitatively different developmental processes are occurring amongst the children tested (Barreau, 1997).The children were divided into three groups on the basis of the scores they were given for the bag questions. Of the 8 children who were scored as correct for the bag questions, 7 were also correct for the tube question, and 1 gave a "double" response. Of the 8 children who gave reversed responses for the bag question, 6 were scored as correct on the tube question, there was 1 reversed response, and 1 double response, and for the 8 children who scored "double" responses for the bag questions, 2 were correct on the tube questions, 2 gave reversed responses, and 4 gave double responses. This pattern of data was considered to be a little too complex to be easily handled by a traditional verbal theory.

A COGENT IMPLEMENTATION

To properly test the theory against the data, a family of models were produced using the COGENT modelling environment. The basic architecture used in this approach is reproduced below: In this figure, hexagons represent processes, rounded rectangles represent buffers, and diamonds represent data boxes. Square boxes represent compounds, which may contain buffers and processes. Arrows with standard heads indicate message sending. Arrows with black triangular tails indicate buffer reading. Compound arrows (which are denoted by triangular and standard heads) allow both functions.

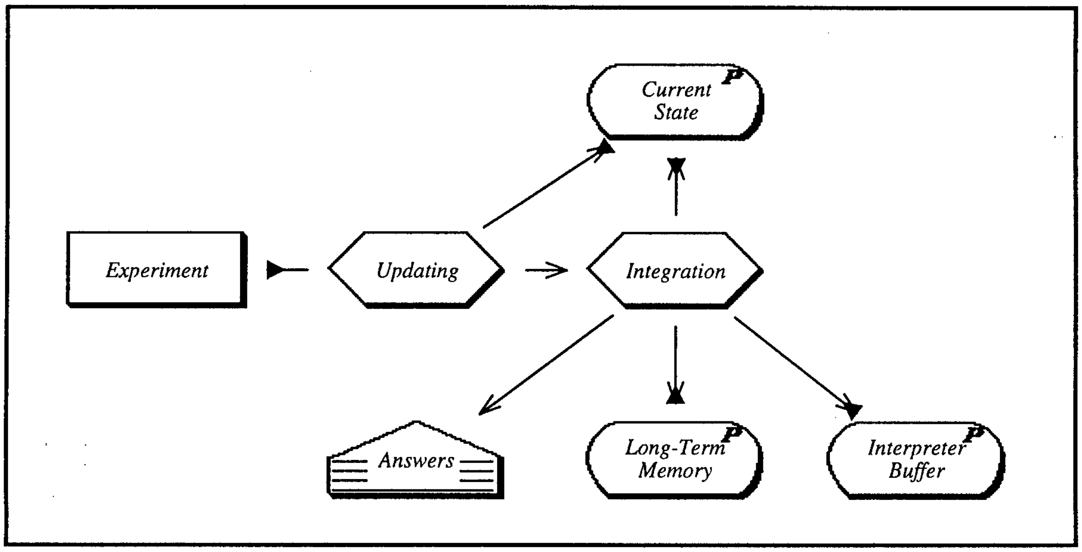


figure 1 - the COGENT object-level representation of the simulation

For the purposes of this paper, the "Experiment" compound is used only as a means of feeding information to the system simulating the child's mental processes, and will not be discussed in any great detail. Note that for the bag experiment, the simulation must include the correct answering of three "belief establishing" questions prior to the five questions of main interest within the experiment. The belief establishing questions were included within the experiment to ensure that the child had formed the correct representations of the state of the world prior to being tested on their memory for the sequence of events. These questions include the initial question of the Smarties task (What do you think is in the tube?), a repeat of the question to ascertain that the child believes the tube is empty (What is in the tube now?) once the transfer operation has taken place, and a question to ensure that the child has tracked the transferral of the supposed Smarties (What do you think is in the bag?). In the experiment, after asking one of the belief establishing questions, the experimenter waited until the child had answered before continuing with the procedure. Accordingly, in the simulation, no further input was fed to the system until the cycle after the system had output the answer to the previous question. This protocol was observed throughout all the simulations.

The Smarties Simulation.

We assume that the 30% of 3- and 4-year olds who pass the Smarties test do so by accessing a long-term memory (LTM) representation of the likely contents of a Smarties tube, so we do not attempt to deal with this question in any detail here. This is consistent with the developmental literature, which has focused only upon those children who fail. The initial simulation then, must be one that gives a "reality" answer to a "belief" question under the circumstances of the Smarties experiment. The experimental procedure is modelled by adding propositions about the current state of the environment a cycle at a time to an "environment" buffer, within the Experiment compound, which is read by the updating process. The Current State Buffer is a representation of current environmental contingencies. This is kept up-to-date by **destructive updating** which occurs by the operation of the following rules:

```
RULE 1.
IF:     A is in Experiment: Environment
        not A is in Current State
THEN:   add A to Current State

RULE 2.
IF:     in(X,Y) is in Experiment: Environment
        in(X,Z) is in Current State
THEN:   delete in(X,Z) from Current State
```

Thus, if *in(tube,smarties)* is in the Current State and *in(tube,pencils)* appears in the Environment, *in(tube,smarties)* is deleted from the Current State by the second of the above rules and is replaced by *in(tube,pencils)*.

The basic workings of the model of the Smarties task are as follows:

In LTM there is a generic representation of past experience of Smarties tubes,

g(in(tube,smarties)),

and a further rule in the integration process that states the contents can be matched to their containers on the basis of such past experience:

```
RULE 3.
IF:     g(in(X,Y)) is in Long-Term Memory
        object(X) is in Current State
        not in(X,Z) is in Current State
THEN:   add in(X,Y) to Current State
```

This rule is refracted, so that it only fires the first time its conditions are satisfied within a COGENT run. When a tube representation is added to the Current State Buffer, this rule fires and the inference is made that the tube contains Smarties. This information is overwritten, however, when the further information is added from the environment that the tube contains pencils. Thus, when the question regarding the contents of the tube is presented to the system

question(present(in(tube, What))),

the present representation of the current contents of the tube in the Current State Buffer instantiates the unknown variable in the question, and provides the only possible answer: *in(tube, pencils)*.

Questions are dealt with by being passed immediately over from the Current State Buffer to the Interpreter Buffer. Once a question is received in the Interpreter Buffer, it activates the relevant search processes according to the following rules:

```
RULE 4.
IF:     question(present(X)) is in Interpreter Buffer
        X is in Current State
THEN:   clear Interpreter Buffer
        add answer(X) to Interpreter Buffer

RULE 5.
IF:     question(past(X)) is in Interpreter Buffer
        record(Y) is in Long-Term Memory
        X is a member of Y
        not X is in Current State
THEN:   clear Interpreter Buffer
        add record (Y) to Interpreter Buffer
        add answer (X) to Interpreter Buffer
```

Thus, the unknown variables within the question are instantiated either in the Current State Buffer or in LTM, and translated into an answer format. All answers within the Interpreter Buffer are immediately sent to the output processes represented in the diagram by the triangular "Answers" block.

The Bag Simulation.

In the case of the bag experiment, the simulation is a little more complex. In particular, we have to tackle the creation of event records. To do this, a rule must fire when an event is perceived to end. This rule translates all information currently being processed (the contents of the Interpreter Buffer), together with the current representation of the environment (the contents of the Current State Buffer) into an LTM format. In the hypothesis underlying the experimental procedure, the event was signalled to be at an end by a contextual change, the removal of the tube. In the simulation, a record is closed if there are more objects represented in the Current State Buffer than are present in the environment. This is captured formally by the updating rule:

```
RULE 6.
IF:     Objects is the list of all object(X) such that
            object(X) is in Experiment: Environment
        Representations is the list of all object(X)
            such that object(X) is in Current State
        A is the length of Objects
        B is the length of Representations
        B > A
THEN:   send close_record to Integration
```

Upon receiving the close_record trigger, a further rule fires within the integration process which transforms the information within the Current State Buffer and the Interpreter Buffer into a list structure in LTM. The Interpreter Buffer is then cleared.

Simulation Results.

The basic simulation can easily handle the results of the first group of children, those who were scored correct on the bag question (group A). When asked the bag questions, the simulation of this group of children has a record available containing the previous belief concerning the bag's contents,

in(tube,smarties)

which it can use to answer the first question (BAG: BELIEF: PAST), in accordance with rule 5. When asked the second bag question (BAG: REALITY: PRESENT), a Current State representation of the bag's current contents is employed to answer this question in accordance with rule 4.

Seven out of eight of this group of children were also scored as correct for the tube question. In the model, the tube question is handled by the existence of a record available in LTM which can be retrieved to answer the question. The creation of this record was triggered by the removal of the tube. Note that the record does *not* contain a verbatim representation that the tube contained marbles. Instead, the record contains the representation that the contents of the tube were emptied into the bag:

action(empty(tube,bag)),

that the tube is now empty:

in(tube,[]) (where [] denotes the empty set),

and that the bag contained marbles. To correctly answer questions regarding the initial contents of the tube (questions 3 and 5, TUBE: BELIEF: PAST and TUBE: REALITY: PAST) a further rule is necessary to allow the inference that the tube's contents can be ascertained by backwards reasoning from the bag's contents, and the fact that the contents of the tube were entered into the bag. Formally, this rule is:

```
RULE 7.
IF      record(Y) is in Interpreter Buffer
        question(past(in(A,B))) is in Interpreter Buffer
        action(empty(A,C)) is a member of record(Y)
        in(C,D) is a member of record(Y)
THEN:   clear Interpreter Buffer
        add answer(in(A,D)) to Interpreter Buffer
```

This rule is triggered if the current representation of the tube's contents is identical to the retrieved LTM representation. Since the child is presumably not expecting to answer a "present" question at this point, the rule allows the search, via inference, for an alternative "past" answer. Note that the simulation demonstrates that Morton's (1997, p. 938) comment that "the conditions are the same" for the tube questions of the bag experiment and for the same questions in the Smarties experiment is not strictly necessary when analysed in terms of the underlying theory. In this simulation, when the inference rule regarding the transferral operation is manually prevented from firing the default answer from the system to the tube questions is that the tube was empty. Since the child was shown the empty tube during the bag episode this forms part of the same record. The full contents of this record are displayed below:

```
record([[in(bag,smarties), in(tube,[]), object(bag),
        action(empty(tube,bag)), object(tube)]
        action(remove(tube))]).
```

With this set of rules, the simulation therefore produces the same answers in the bag experiment as seven out of eight of the children in group A.

The initial results of those children who were scored as giving "reversed" answers (group B) to the bag question need to be explained differently. Recall that these children gave reality answers to belief questions and vice versa. The simulation of this situation uses the same basic structure as the simulation of group A (the "corrects"). However, it is assumed that the group B children attempt to answer all questions initially from their current state representation of the world. Arguably this is less effortful than retrieving information from LTM (see Morton, Hammersley & Bekerian, 1985 for a discussion of the complexities of retrieval from LTM). In effect, we assume that the tagging of questions as referring to past and present is not as well established in this group as in group A. The group B children, then, are not forced to search LTM in response to a PAST question. Rather, they only look in LTM when the Current State search has failed. Since the Current State Buffer representation is one of reality rather than belief, these children's default strategy results in a reversal of belief and reality answers.

Briefly, the simulation of this state of affairs works as follows. The "past" and "present" modifiers in the input are ignored in the integration process by rules 4 and 5, and, instead, all questions are followed by an initial search in CS. This leads to the initial mistake. The reversal of the situation with the next question is simply implemented by making that the look-up rule for information in Current State into a refracted rule so that it cannot be used as a default when the next question is asked. This is the "present" reality question, and the only way the child can answer the question is by searching for a long-term memory representation with information about the contents of the bag. This is found in the record which specifies

in(bag,smarties)

resulting in a reversed pattern of results.

This simulation works well when only the bag question is considered, but runs into problems when the tube questions are also added to the simulation's input, since it produces a further "reversed" pattern of results for these questions. In fact only one child in this group was scored as giving "reversed" responses to the tube question, and six were scored as correct. This failing will be considered in more detail later.

The final group of children to be considered (group C) gave the "reality" answers to "belief" questions. Working on the logic employed in the simulation of group B's results it is assumed that these children also ignore the past/present modifiers and attempt to answer the question in the simplest way possible, by retrieving an answer from the Current State Buffer representation. However, for these children the assumption is that the search rule for the Current State Buffer is not refracted. Accordingly, the simulation produces repeated answers from the Current State Buffer, which are identical to the "double" responses given by this group. Of the eight children who were scored as "doubles" on the bag questions, this simulation matches the repeated "double" scores of four of these children on the tube questions.

GENERAL DISCUSSION

Successes and Failings

The memory-updating explanation of the Smarties task is outlined by Morton (1997), and the 3-buffer architecture used here to simulate this theory was derived from Barreau (1997), (see Barreau, 1997 for an account of why a 3-buffer system is necessary). The resulting simulation, however, differs in significant ways from either of these accounts. It is intended to be a forerunner of a number of such simulations, building up a set of mutual constraints on later models of on-line processing by this age group (c.f. Barnard, 1985). As such, it has a number of distinct successes and flaws. Not least amongst its successes is that it is - to our knowledge - the only fully specified computational theory of 3- and 4- year olds failings at "false belief" tasks. Other accounts of these phenomena rely upon the assumption that children of this age suffer from a conceptual deficit in representing the beliefs of others, and their own earlier beliefs if inconsistent with current reality (e.g., Hogrefe, Wimmer & Perner, 1986; Perner, Leekam & Wimmer 1987), or else are in other ways not as completely specified as the account given here (Halford, Wilson & Phillips, in press).

Viewed as a modelling project in its own right, a number of flaws become evident with the current account. Firstly, if it is considered to be a straightforward account of the current data independent of theoretical statements put forward elsewhere (Barreau, 1997; Morton, 1997), then it suffers from a rather poor fit to the data in the case of group B, the "reversed" response children. The mechanism which allows for a reversed response to the bag questions should also produce reversed responses for the tube questions. However, the majority of children in this group (six out of eight) were scored as correct in this case.

Elsewhere, the fit to the data is better. The account given by the basic bag simulation is also able to account for the failure of children at the Smarties task with no change to the model, merely altering the input to simulate the change in task. This simulation correctly produces the same results as the "correct" group (A) on all the questions. The modified simulations for groups B and C also give the identical patterns of results to the children they were intended to model for the bag questions, and in the case of group C (the "double" responses) this success is repeated with the simulation giving the same results as the largest subset of these children.

The conclusion to be drawn from this pattern of success and failure is that although there is a large degree of agreement between the performance of the children and that of the underlying model, there is a flaw in the manner in which the model operates. In particular, it should not function in the same way in response to the tube questions as it did to the bag questions. There are two broad ways of accomplishing this. The first is to add other rules which would interpret the material in the record in response to questions concerning the tube. A backwards inference using rule 7 concerning belief could take

action(empty(tube,bag))
in(bag,smarties)

and come up with

in(tube,smarties)

to go along with the *in(tube,[])* already available in the record. The ordering of these two contradictory options in the buffers could give rise to the differences in responding to the tube questions among the children in group B.

The second general approach to the mismatch is to change the way in which the Group A children solve the questions. One approach is to create records of questions and answers. This would make the answer to the initial belief question available, even though the primary representation *in(tube,smarties)* has been deleted. Use of the record

```
record([[in(bag,smarties), in(tube,[]), object(bag),
        action(empty(tube,bag)), object(tube)]
        action(remove(tube))]).
```

would then be restricted to questions about the tube. This resembles the account given by Barreau (1997). To achieve all this, we will have to characterise the differences among the three groups of children somewhat differently. Both these options will be explored in the next phase of simulation.

ACKNOWLEDGEMENTS

The experimental data which provided the impetus for this simulation were provided by Sofka Barreau. The resulting simulation has greatly benefitted in various ways by discussions with Paul Abeles, Sofka Barreau, Rick Cooper, Cathy Dillon, John Fox and Kerry Sims.

REFERENCES

Anderson, J. R. (1996). *Rules of the mind.* Erlbaum, Hillsdale N. J.

Barnard, P.J. (1985) Interacting Cognitive Subsystems: A psycholinguistic approach to short term memory. In A. Ellis (Ed.) Progress in the Psychology of Language, (Vol. 2), Chapter 6, London: Lawrence Erlbaum Associates, 197-258

Barreau, S. (1997) Developmental constraints on theories of memory. PhD thesis, Department of Psychology, University College London.

Cooper, R., & Fox, J. (in press). COGENT: A visual design environment for cognitive modelling. *Behavior Research Methods, Instruments and Computers.*

Elman, J. L., Bates, E. A., Johnson, M. H., Karmiloff-Smith, A., Parisi, D., & Plunkett, K. (1996). *Rethinking innateness: A connectionist perspective on development.* MIT Press, Cambridge, Ma.

Halford, G., Wilson, W. H., & Phillips, S. (in press). Processing capacity defined by relational complexity: Implications for comparative, developmental, and cognitive psychology. *Behavioral and Brain Sciences.*

Hogrefe, G.-J., Wimmer, H., & Perner, J. (1986). Ignorance versus false belief: A developmental lag in attribution of epistemic states. *Child Development*, 57, 567-582.

Morton, J. (1997). Free associations with EPS and memory. *Quarterly Journal of Experimental Psychology,* 50A, 924-941.

Morton, J., Hammersley, R. H., & Bekerian, D. A. (1985). Headed records: A model for memory and its failures. *Cognition*, 20, 1-23.

Newell, A. (1990). *Unified theories of cognition.* Harvard University Press, Cambridge, Ma.

Perner, J., Leekam, S. R., & Wimmer, H. (1987). Three-year-olds' difficulty with false belief: The case for a conceptual deficit. *British Journal of Developmental Psychology*, 5, 125-137.

Modelling Common-Sense Psychology and the False Belief Test

Stuart Watt
Knowledge Media Institute and Department of Psychology
Open University
Walton Hall, Milton Keynes, MK7 6AA, UK
+44 1908 653169
S.N.K.Watt@open.ac.uk

ABSTRACT

In this paper, we describe a cognitive modelling framework for common-sense psychology. We'll show a number of comparable cognitive models for different theories of common-sense psychology, and show that these models can help to illuminate some of similarities and differences between the differing theories.

Keywords

Common-sense psychology, theory of mind, false belief test, cognitive model

INTRODUCTION

Common-sense psychology — or people's common sense ability to think about our own and other people's minds — is currently being researched actively in several different disciplines. While this interdisciplinary collaboration can be very productive, it can lead to its own problems. This is exacerbated by complexity, both methodological and theoretical, of common-sense psychology itself.

Much of the problem is that nobody is really sure what common-sense psychology is, theoretically. Astington and Gopnik (1991), for example, distinguish between six different possible interpretations, all of which are subtly different. There are many different theories of common-sense psychology. Unfortunately, there is no common ground which allows these different theories to be compared and contrasted. In this paper, we'll introduce a cognitive model that can begin to play that role.

To compare the different theories, we'll use a standard tool from common-sense psychology, Baron-Cohen *et al.*'s (1985) false belief test. We'll begin by introducing and describing this test, and one of the theories of common-sense psychology, Leslie's (1987) 'decoupler' model. Although common-sense psychology is hugely complex, and can only be modelled in the most sketchy form, we'll show how Leslie's theory can be implemented as a cognitive model. Finally, we'll show how alternative theories of common-sense psychology can be represented as small variations on this model, and that we can draw some conclusions about the similarities and differences between the theories with this modelling framework.

MODELS OF COMMON-SENSE PSYCHOLOGY

While common-sense psychology has been a focus for recent research, most work in this either has either been experimental or purely theoretical; there are few cognitive models in this area, even though it is precisely the kind of area that modelling has proved so helpful for in the past (Samet, 1993). The exception is the work of Shultz (1988, 1991). All the models which have been developed, though, focus on small parts of the problem; for example, studying how people assess whether or not planned actions were intentional (Shultz, 1988).

We propose a different strategy. Instead of a narrow but deep model, we propose using a broad but shallow one; one which can be used to compare theories on a grand scale. With this level of modelling, we believe that even in the limited false belief test, we can help to clarify the similarities and differences between some of the grand scale theories in the field.

THE FALSE BELIEF TEST

The false belief test has its origins in Premack and Woodruff's (1978) experiment to determine whether or not chimpanzees could reason about one another's mental states — whether or not they had a "theory of mind", another term for common-sense psychology. Unfortunately, there was a methodological problem with this experiment; their chimpanzee subject, Sarah, could use her own beliefs rather than reasoning about another's, because the two were identical. To prove that Sarah was really able to reason about another's beliefs, they had to show that Sarah could still predict another's behaviour when her beliefs were different from that other's — that is, when the other had beliefs which Sarah believed to be false.

Following these problems with Premack and Woodruff's experiment, Wimmer and Perner (1983) devised a false belief test, which evaluated a (human) subject's to ascribe definite but false beliefs to another. Baron-Cohen *et al.* (1985) later simplified Wimmer and Perner's test so they could compare autistic, Down's syndrome, and normal children at different ages. Baron-Cohen *et al.*'s simplified false belief test is shown in figure 1.

Baron-Cohen *et al.*'s false belief test is presented as a simple story. There are two puppets, Sally and Anne. Sally has a marble, which she keeps in a basket. Then Sally leaves the room, and while she is away Anne takes the marble out of the basket and hides it in the box. Sally comes back into the room.. The child subject is then asked the question: "where will Sally look for her marble?" Older children say that she will look in the basket, because although they know the marble is in the box, they know that Sally doesn't know it has been moved from the basket, and they can distinguish Sally's (false) belief from their own (true) belief. Younger children, on the other hand, and autistic children, do not distinguish between the two They simply say that Sally will look in the box. The false belief test, therefore, explores the change that happens as common-sense psychology develops.

Baron-Cohen *et al.*'s theory was that a failure in the development of common-sense psychology might be responsible for autism, and the results from their experiment (and others which followed) certainly seemed to bear that out. As a result, there has been a focus of interdisciplinary research which has led to a number of different hypotheses about the nature and development processes involved in common-sense psychology.

Figure 2 shows a model for one possible theory of common-sense psychology, Leslie's 'decoupler' model. At the heart of Leslie's model is a manipulator that is capable of pretence — of decoupling beliefs from one context and applying them in another. It is this that makes reasoning about false beliefs possible, because a child can use this decoupling mechanism to separate someone else's beliefs into a different context from their own.

Given this simple theory of common-sense psychology, we will now turn to the cognitive model, and show how Leslie's 'decoupler' model can be represented in a model. But first, a few words on the modelling environment that we'll be using.

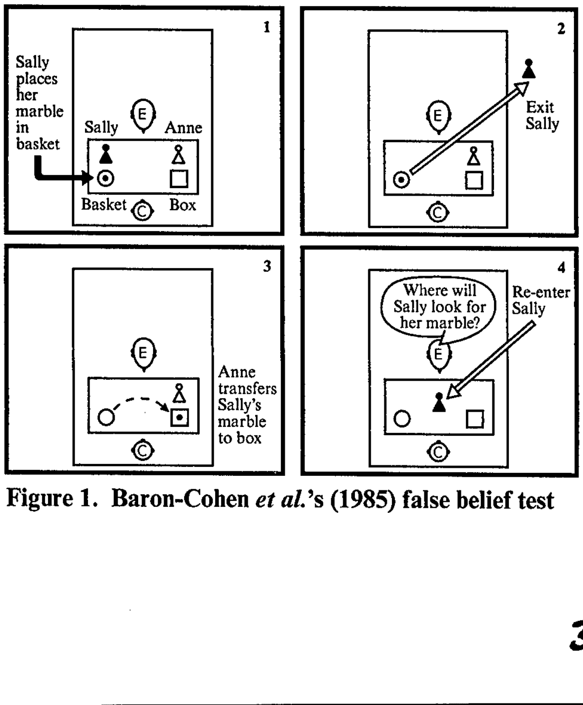

1
Sally places her marble in basket
Sally
E
Anne
Basket
C
Box
2
E
Exit Sally
C
3
E
Anne transfers Sally's marble to box
C
4
Where will Sally look for her marble?
Re-enter Sally
E
C

Figure 1. Baron-Cohen *et al.*'s (1985) false belief test

THE MODELLING ENVIRONMENT

Before we can build the models adequately, we need a representation language that is strong enough to do the physical and psychological reasoning required. In practice, the psychological parts of the model require the ability to reason about different contexts, distinguishing one agent's false beliefs from another agent's true beliefs. Something like a modal logic, therefore, is going to be required (Leslie, 1988, makes a direct comparison between the requirements for common-sense psychology and the properties of modal logics).

The model we present borrows this from McCarthy's (McCarthy & Hayes, 1969) 'situation calculus', where the effects of an event are described as a consequence relation between one state and another. At the core of McCarthy's calculus is a special function *result*, which represents the effects of an action on a situation by returning a new, modified, situation. The function *result(p, σ, s)*, where *p* is a person, σ is an action, and *s* is a situation, has a value which is a new situation representing the effects of *p* doing σ in *s*. For example:

$$\mathit{inside}(\mathit{marble}, X, s) \land \neg\, \mathit{inside}(\mathit{marble}, \mathit{box}, s) \Rightarrow \mathit{inside}(\mathit{marble}, \mathit{box}, t) \land \neg\, \mathit{inside}(\mathit{marble}, X, t)$$

where $t = \mathit{result}(\mathit{alison}, \mathit{putin}(\mathit{marble}, \mathit{box}), s)$

This says that if *marble* is inside something that isn't *box* in situation *s*, the effect of *alison* putting *marble* in *box* is a new situation *t* such that *marble* is no longer where it was (in *X*), but is now inside *box*.

The full situation calculus is more powerful and more complicated than this implies, but this subset of it is sufficient for the purposes of this model, and further, it doesn't need the heavy inference machinery that a complete modal logic would. The situation calculus, then, is strong enough for the model, fairly easy to use computationally, yet it retains the referential properties of modal logics (McCarthy & Hayes, 1969).

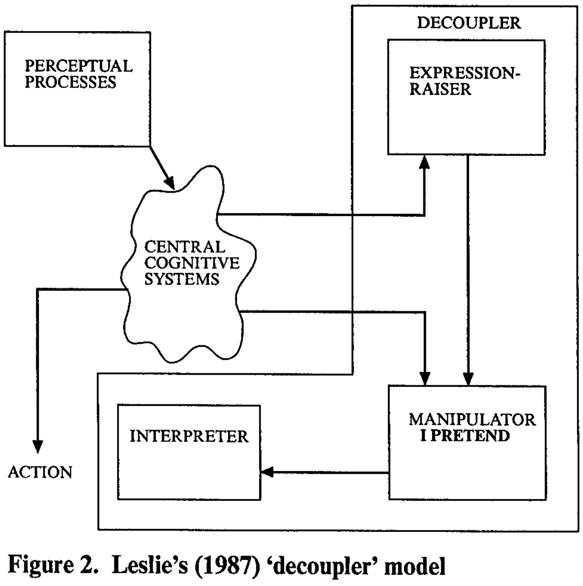


Figure 2. Leslie's (1987) 'decoupler' model

The model implements a modified subset of the situation calculus in a Prolog-like language embedded in Common Lisp. Apart from the Lisp-like syntax, there is only one significant difference from standard Prolog — variables are normally prefixed with a *?* question mark, but output variables in a clause head are prefixed with a ^caret. *?value* and *^value* refer to the same variable.

MODELLING LESLIE'S 'DECOUPLER'

The base model for the false belief test comprises a number of separate modules. There include;

- a physical environment model,
- a basic physical reasoning module,
- a basic psychological reasoning module, and
- a script for the false belief test.

The Physical Environment Model

The first part of the modelling environment is a physical environment model which implements an event-driven simulation environment. As objects are physically moved from one place to another events are generated and passed to all objects equipped with sufficient perceptual apparatus to be aware of them.

The Physical Reasoning Module

Even in the false belief test, physical reasoning is needed. The basic physical reasoning module is shown in figure 3. This implements the rules that Alison (as we'll call the subject in the false belief test) uses to make predictions about what happens as a result of physical actions and events.

```
;;; If we see ?object in a place ?container, then we find out
;;; where it was in the situation, and return a new situation
;;; so that it is now in ?container.

((result yes ?stance-to (place ?object ?container)
   ?situation ^new-situation) :-
 (member (inside ?object ?outer) ?situation)
 (difference ?situation
   ((inside ?object ?outer)) ?situation1)
 (append ?situation1
   ((inside ?object ?container)) ?new-situation))

;;; If we see an object being put into a new place, ?container,
;;; then again we find out where it was before in the situation,
;;; and return a new situation so that it is now in ?container.

((result yes ?stance-to (put-in ?object ?container)
   ?situation ^new-situation) :-
 (member (inside ?object ?outer) ?situation)
 (difference ?situation
   ((inside ?object ?outer)) ?situation1)
 (append ?situation1
   ((inside ?object ?container)) ?new-situation))

;;; If we see an object being taken out of a place ?container,
;;; we return a new situation so that it is no longer in
;;; ?container, but is now outside it, in ?outer-container.

((result yes ?stance-to
   (take-out ?object ?container)
   ?situation ^new-situation) :-
 (member (inside ?container ?outer) ?situation)
 (difference ?situation
   ((inside ?object ?container)) ?situation1)
 (append ?situation1
   ((inside ?object ?outer)) ?new-situation))
```

Figure 3. The basic physical reasoning module

As far as physical reasoning is concerned, only three *result* actions are of interest. First, people can see an object being put into a container. Second, people can see an object being taken out of a container. And third, if a person enters a room, they can see all the objects (but not contained, or hidden, objects) within that room. All three of these actions serve to keep a person's model of the physical

The Psychological Reasoning Module

At the core of the model is a representation of one person's ability to reason about other people's mental states. This basic psychological reasoning module, corresponding to Leslie's theory of mind mechanism, is shown in figure 4. There are three *result* rules. The first rule is associated with *perceived* events; this is where the essence of psychological reasoning happens. The other two rules are associated with *believes* events, and are used for modelling the answering of questions; for this reason they print out an answer.

The first *result* rule uses the *ascribe* rule to keep all the notional worlds up to date with the *perceived* event. The *ascribe* rule implements the decoupler model in figure 2. It works like this. First, the *those* procedure is used to get all of *?self's* beliefs out of the situation; this corresponds to *?self's* notional world. Next, the *requote* procedure is used to raise all the expressions in the notional world, to create a new situation, *?situation2*. Then, the rule passes this new situation to the interpreter, through the manipulator. The manipulator is played by the *in-stance* procedure, which 'pretends' to be in the right context to handle the given event. The interpreter is called by the nested call to the *result* procedure. Finally, the nested call to result returns a new situation, *?situation3*, which is passed to *requote* again to restore its expression status in *?new-self-notional-world*. This is then used to replace the old notional world in the situation, and the modified situation is returned.

Perhaps this will be clearer with a more concrete example. Imagine that we ask *(result ?response sally (perceived sally (put-in marble box) ?S, ?NewS)*, in a situation *?S*. Because this is a *perceived* event, the first *result* rule will be applied, calling *ascribe*. The *those* and *require* procedures are used to go through the situation *?S*, decoupling all the relations *(believes sally ?X)* and generating a new situation *?S'*. Then the model applies the physical reasoning rules in this new situation *?S'*, to generate an updated physical situation *?R'*. The second requote call goes through *?R'* to restore its quotation status to normal, and returns *?R*. Finally, *?R* is used to replace all Sally's beliefs in *?S*, and the final situation returned in *?NewS*.

The Script for the False Belief Test

The final component of the model is a script for the false belief test. This is shown in figure 5. There are two parts to this script. First, there are a serious actions which corresponds more or less to the movements of the characters in Baron-Cohen *et al.*'s story, shown in figure 1. Second, there are a number of questions; these are the kind of questions that an experimenter might ask a subject after acting out the scenario. It is the answers to these questions which reveal whether or not, or how, the child passes the false belief test.

So far, we have described a basic version of the theory of mind mechanism, a version which successfully models the passing of the false belief test. With this in place, we can now begin to compare this with some of the alternatives. In this paper, we will only look at three alternative theories of common-sense psychology, the simulation theory, the copy theory, and the situation theory.

```
;;; The rules for handling perceived events. When you
;;; perceive something and see that ?someone, sees the
;;; same thing, get ?someone's notional world into ?self-
;;; notional-world, and then, in that world, predict its
;;; physical effects. Then map these physical effects into
;;; changes to ?someone's notional world.

;;; Rule perceive
((result ^response ?someone
   (perceived ?object (?action ?other-object ?event))
   ?situation ^new-situation) :-
 (ascribe ?someone ^response ?someone
   (perceived ?object (?action ?other-object ?event))
   ?situation ^new-situation))

;;; Rule ascribe
((ascribe ?someone ^response ?other
   (perceived ?object (?action ?other-object ?event))
   ?situation ^new-situation) :-
 (those (believes ?someone ?something) ?situation
   ?notional-world)
 (requote (believes ?someone ?something)
   ?notional-world ?something ?situation2)
 (in-stance ?other-object ?action
   (result ?response ?other-object
     (?action ?other-object ?event)
     ?situation2 ?situation3))
 (requote ?something ?situation3
   (believes ?someone ?something) ?new-notional-world)
 (difference ?situation ?notional-world ?situation1)
 (append ?new-notional-world ?situation1
   ?new-situation))

;;; These are the rules for answering questions about
;;; people's beliefs. In effect, all that happens is that we
;;; look for the truth of the question in ?object's notional
;;; world.

;;; Rule answer-yes
((result yes ?someone (believes ?object ?something)
   ?situation ^situation) :-
 (member (believes ?object ?something) ?situation)
 (write-list (yes ?object believes ?something)))

;;; Rule answer-no
((result no ?someone (believes ?object ?something)
   ?situation ^situation) :-
 (not (member (believes ?object ?something)
              ?situation))
 (write-list (no ?object does not believe ?something)))
```

Figure 4. The basic psychological reasoning module

COMPARING MODELS 1: THE SIMULATION THEORY

The first alternative theory to be compared against Leslie's is the 'simulation theory', which is typified by a 'role taking' or 'perspective taking' approach. Gordon illustrates this by saying that "*Smith believes that* Dewey won the election" should be read as "let's do a Smith simulation. Ready? *Dewey won the election*" (Gordon, 1986, original emphasis).

According to the simulation theory, young children are simply unable to take other people's points of view. This can be modelled by dividing the main *perceive* rule into two — one for self, and one for others. In young children, the *perceive* rule for self functions as before, but the *perceive* rule for others does nothing. This is shown in figure 6.

When run, this seems to fail the false belief test correctly in that Alison doesn't give answers at all for either Sally or Anne; before Alison can pass the test she needs to acquire the ability to simulate, or take the role of, other people. This corresponds to the development of a simulation ability: "before internalising this system, the child would simply be unable to predict or explain human action [but] after internalising the system the child could deal indifferently with ac-

```
;;; Start by introducing the characters. The order doesn't
;;; matter much. Alison will become aware of all the other
;;; objects as soon as she enters the room.

(tell-model (put-in basket room))
(tell-model (put-in box room))
(tell-model (put-in marble room))

(tell-model (put-in sally room))
(tell-model (put-in anne room))

(tell-model (put-in alison room))

;;; Put the marble in the basket
(tell-model (put-in marble basket))

;;; Sally leaves the room
(tell-model (take-out sally room))

;;; Move the marble from the basket into the box
(tell-model (take-out marble basket))
(tell-model (put-in marble box))

;;; Sally comes back into the room
(tell-model (put-in sally room))

;;; Where does Alison think that the marble is?
(ask-object-if alison
  (believes alison (inside marble ?where)))

;;; Where does Alison think that Sally thinks the marble is?
(ask-object-if alison
  (believes sally (inside marble ?where)))

;;; Where does Alison think that Anne thinks the marble is?
(ask-object-if alison
  (believes anne (inside marble ?where)))
```

Figure 5. Actions and questions for the false belief test

tions caused by true beliefs and actions caused by false beliefs" (Gordon, 1986). This is why the kind of failure in the simulation theory is interesting; Alison simply fails to give answers for either Sally or Anne, because she failed to take their roles properly.

The second stage in the model, then, is the complete simulation rule, which implements a role taking strategy through the *in-self* primitive. This primitive has the effect of temporarily pretending to be a different self, and then handling the whole event in that context instead. It is this replacement second rule that allows Alison to pass the false belief test. The replacement rule which models this strategy is shown in figure 7.

There are a number of important conclusions to be drawn from this idea. First, in the simulation theory the behaviour involved in ascribing mentality to oneself is different from that involved in ascribing mentality to others. This contrasts with the theory of mind mechanism described earlier, where there is no difference between first person and third person ascription. This is shown by the rules' sensitivity to the *self* relation, which shows that there is an egocentricity involved in the simulation theory. The second point to note is that, in practice, the behaviour of this system is the same as that of the basic psychological reasoning module shown in figure 4, because the replacement second rule combines with the first to behave just as if there was a single rule using the *ascribe* action, a rule identical to the first *result* rule in figure 4. This is in accord with Perner's (1994) suggestion that, in practice, the difference between a theory and a simulation may be at worst one of emphasis.

```
;;; Here are the rules for the simulation theory. Initially, if
;;; we are seeing something ourselves, then we do the right
;;; ascription, otherwise we leave the situation alone. These
;;; two rules, together, replace the perceive rule in figure 4.

;;; Rule perceive-self, compare to perceive in figure 4
((result ^response ?someone
   (perceived ?object (?action ?other-object ?event))
   ?situation ^new-situation) :-
 (self ?someone)
 (ascribe ?someone ?response ?someone
   (perceived ?object (?action ?other-object ?event))
   ?situation ?new-situation))

;;; Rule perceive-other, compare to perceive in figure 4
((result ^response ?someone
   (perceived ?object (?action ?other-object ?event))
   ?situation ^situation) :-
 (not (self ?someone)))
```

Figure 6. Rules for the simulation theory (first version)

```
;;; The replacement second rule for the simulation theory. If
;;; we are not seeing something for ourselves, then we
;;; "pretend" to be someone else through the in-self primitive,
;;; and process the event as if we were that person. This rule
;;; replaces the perceive-other rule in figure 6.

;;; Rule perceive-other, compare to perceive-other in
;;; figure 6.
((result ^response ?someone
   (perceived ?object (?action ?other-object ?event))
   ?situation ^new-situation) :-
 (not (self ?someone))
 (in-self ?someone
   (result ?response ?someone
     (perceived ?object (?action ?other-object ?event))
     ?situation ?new-situation)))
```

Figure 7. Replacement rule for the simulation theory

COMPARING MODELS 2: THE COPY THEORY

The second model I'll compare against Leslie's theory of mind mechanism is Chandler's 'copy theory'. Chandler and Boyes describe younger children as behaving "as though they believe objects to transmit, in a direct-line-of-sight fashion, faint copies of themselves which actively assault and impress themselves upon anyone who happens in the path of such 'objective' knowledge" (Chandler and Boyes, 1982). They argue that this is the precursor to a complete theory of mind such as Leslie's, and therefore I'll only show the version which fails the false belief test — a version which passed the test would be identical to the complete model in figure 4.

From the complete model of the theory of mind mechanism corresponding to an adult theory of mind,

```
;;; Here are the ascription rules for the copy theory. Initially,
;;; if we are seeing something ourselves, then we do the right
;;; ascription, otherwise we leave the situation alone. These
;;; two rules, together, replace the perceive rule in figure 4.
;;; Note that these replacement rules are identical to those
;;; in figure 6.

;;; Rule perceive-self, compare to perceive in figure 4
((result ^response ?someone
   (perceived ?object (?action ?other-object ?event))
   ?situation ^new-situation) :-
 (self ?someone)
 (ascribe ?someone ?response ?someone
   (perceived ?object (?action ?other-object ?event))
   ?situation ?new-situation))

;;; Rule perceive-other, compare to perceive in figure 4
((result ^response ?someone
   (perceived ?object (?action ?other-object ?event))
   ?situation ^situation) :-
 (not (self ?someone)))

;;; Here are the answering rules for the copy theory. They
;;; have the effect of considering the target's notional world
;;; to be a 'copy' of the ascriber's. These rules replace the
;;; rules answer-yes and answer-no in figure 4.

;;; Rule answer-yes-self, compare to answer-yes in
;;; figure 4
((result yes ?someone (believes ?object ?something)
   ?situation ^situation) :-
 (self ?self)
 (member (believes ?self ?something) ?situation)
 (write-list (yes ?object believes ?something)))

;;; Rule answer-no-self, compare to answer-no in figure 4
((result no ?someone (believes ?object ?something)
   ?situation ^situation) :-
 (self ?self)
 (not (member (believes ?self ?something) ?situation))
 (write-list (no ?object does not believe ?something)))
```

Figure 8. Rules for the copy theory

we can modify the psychological reasoning module slightly to represent a child with a copy theory of belief. The main point of the copy theory is, in effect, that instead of ascribing beliefs to others, a 'copy' of one's own beliefs is used instead. Instead of building different notional worlds for Sally and Anne, both use the same, a copy of Alison's.

According to the copy theory, children simply do not ascribe real beliefs to others. This is shown by the modified *result* rules in figure 8, which replace the *result* rule in figure 4 so that beliefs are only ascribed to oneself. Note that these *result* rules are identical to the first (before full theory of mind) version of the simulation theory in figure 6. This is to be expected — Chandler's theory is an account of how children escape the kind of egocentricity that marks a simulation theory. But this is not the whole story in the copy theory; when children are asked about other people's beliefs, they answer by drawing on their own. For this, we also need to change the result rules for the believes relation; these are the rules which model how the child answers the kind of questions used in the false belief test. These changes are also shown in figure 8. Both the question rules are changed from figure 3 by using the *self* relation to find and use one's own beliefs, rather than anybody else's, to answer the given question. Because of this dependence on the self relation, this model shows that the copy theory, like the simulation theory, has an implicit (if rather better hidden) egocentricity.

There are more complex variations on the copy theory; for instance, Wellman (1990) argues that younger children have a copy theory of belief, but not of desires. This is outside the scope of this model because desire psychology isn't yet part of the modelling environment — this is an area for future work. But while the copy theory works to the extent that, when run, it correctly fails the false belief test, the model is quite radically different from an adult theory of mind, and it does seem to require a developmental jump of significant magnitude. All the egocentricity of the rules in figure 8 must be lost, and the child needs to learn to extend notional worlds to other people. This matches all the empirical evidence that is against a copy theory; Perner (1991) has argued convincingly that experiments involving inference from parts to wholes show that the evidence is against children having a copy theory at any age. Even so, this is something which could, in principle, be investigated further quite easily with this modelling approach.

COMPARING MODELS 3: THE SITUATION THEORY

The third reference comparison I'll make against the theory of mind mechanism is Perner's (1991) 'situation theory'. Perner's theory is substantially different from those presented so far because he draws a hard distinction between real and non-real situations, or contexts. The notional world an agent has of itself has a unique status. This is not mirrored in the basic psychological reasoning module in figure 3.

Perner argues that the reason younger children don't pass the false belief test is because the child subject applies the verbal form of questions incorrectly to the situation corresponding to reality, not to the non-real situation which has been played out by the puppets. According to the situation theory, unlike the copy theory, young children do have notional worlds, but they are not so good at understanding that a real question can apply to a non-real situation. Perner uses this distinction to explain why children who fail the false belief test are still capable of sophisticated notional world reasoning, such as that required by Zaitchik's (1990) 'false photograph' test.

Figure 9 shows the rules for the first version of the situation theory model — the version which models a child who cannot yet pass the false belief test. Note

```
;;; The key to Perner's model is a clear distinction between
;;; the status of one's own notional world, and those of others.
;;; This is represented in these models by adding a status flag
;;; to the rules which ascribe those notional worlds. This
;;; status value is knows for one's own notional world, and
;;; believes for other people's. These two rules, together,
;;; replace the perceive rule in figure 4.

;;; Rule perceive-self; compare to perceive in figure 4
((result ^response ?someone
   (perceived ?someone (?action ?other-object ?event))
   ?situation ^new-situation) :-
 (self ?someone)
 (ascribe ?someone knows ?response ?someone
   (perceived ?someone (?action ?other-object ?event))
   ?situation ?new-situation))

;;; Rule perceive-other; compare to perceive in figure 4
((result ^response ?someone
   (perceived ?object (?action ?other-object ?event))
   ?situation ^new-situation) :-
 (not (self ?someone))
 (ascribe ?someone believes ?response ?someone
   (perceived ?someone (?action ?other-object ?event))
   ?situation ?new-situation))

;;; The ascription rule is extended to take the additional
;;; status value. This value is used, instead of the fixed status
;;; value believes, to distinguish between one's own notional
;;; worlds and other people's. This rule replaces the ascribe
;;; rule in figure 3.

;;; Rule ascribe; compare to ascribe in figure 4
((ascribe ?someone ?status ^response ?other
   (perceived ?object (?action ?other-object ?event))
   ?situation ^new-situation) :-
 (those (?status ?someone ?something)
   ?situation ?notional-world)
 (requote (?status ?someone ?something)
   ?notional-world ?something ?situation2)
 (in-stance ?other-object ?action
   (result ?response ?other-object
     (?action ?other-object ?event)
     ?situation2 ?situation3))
 (requote ?something ?situation3
   (?status ?someone ?something) ?new-notional-world)
 (difference ?situation ?notional-world ?situation1)
 (append ?new-notional-world ?situation1
   ?new-situation))
```

Figure 9. Ascription rules for the situation theory

that the main *result* rule has been split into two: one for self and one for others. Superficially, this might look like egocentricity again, but this time the only difference between them is in the status they assign to different notional worlds, *knows* for self, and *believes* for others. Initially, as shown by the modified answer rules in figure 10, children can only link verbal questions to the world for self beliefs — the notional world with the status *knows*. Other notional worlds can and do exist, though; it is just that they cannot be accessed through verbal questions.

Perner claims that the principal change in children between the ages of two and a half and four is the acquisition of a representation theory, which allows them to recognise that questions can refer not to reality, but to worlds or situations that are represented — that is, worlds or theories with the believes predicate. This corresponds to the child's development from a situation theorist into a representation theorist, shown in the modified rules in figure 11.

Perner argues that this change isn't a radical overturning of the existing theory — the kind of radical change that makes the copy theory implausible. Instead, he suggests that the change that happens is a "theory extension" (Perner, 1991), a relatively minor change to the existing theory. This character if theory extension is important to any developmental account of common-sense psychology, because the empirical evidence is that common-sense psychology develops gradually, not in big jumps (Carey, 1985).

```
;;; These are the rules for answering questions about one's
;;; own beliefs. In this group, the "believes" question is
;;; coupled to the knows predicate of a notional world. These
;;; implement the 'self' half of the answer rules in figure 4.

;;; Rule answer-yes-self, compare to answer-yes in figure 4.
((result yes ?someone (believes ?self ?something)
   ?situation ^situation) :-
 (self ?self)
 (member (knows ?self ?something) ?situation)
 (write-list (yes ?self believes ?something)))

;;; Rule answer-no-self, compare to answer-no in figure 4.
((result no ?someone (believes ?self ?something)
   ?situation ^situation) :-
 (self ?self)
 (not (member (knows ?self ?something) ?situation))
 (write-list (no ?self does not believe ?something)))

;;; These are the rules for answering questions about other
;;; people's beliefs. This is a model of what happens before
;;; the representation theory is acquired, where the effect is
;;; to link into the knows predicate instead of the believes
;;; predicate. These implement the 'other' half of the answer
;;; rules in figure 4.

;;; Rule answer-yes-other, compare to answer-yes in
;;; figure 4.
((result yes ?someone (believes ?object ?something)
   ?situation ^situation) :-
 (not (self ?object))
 (member (knows ?self ?something) ?situation)
 (write-list (yes ?object believes ?something)))

;;; Rule answer-no-other, compare to answer-no in
;;; figure 4.
((result no ?someone (believes ?object ?something)
   ?situation ^situation) :-
 (not (self ?object))
 (not (member (knows ?self ?something) ?situation))
 (write-list (no ?object does not believe ?something)))
```

Figure 10. Answer rules for the situation theory

DISCUSSION

These models highlight several of the most important features of the common-sense psychology that underlies the false belief test, and show that these features can be emphasised by models that represent the different and competing theories in this field. Of the models presented, the one that seems to work best in this modelling framework is Perner's 'situation theory' model. The principal reason for this is that the apparent distance between passing and failing the false belief test is much smaller. For both the simulation theory and for Chandler's 'copy theory' there must be a radical development to the ascription of notional worlds. Perner's model clearly shows the character of theory extension which he suggests should be expected of a theory which matches the empirical psychological data on the development of these theories (Carey, 1985).

The simulation theory is quite similar to the version of Leslie's theory of mind mechanism that we have used as a base model — but both it and Chandler's copy theory show an apparent egocentricity. In practice, as I've argued, there are good reasons for supposing that in any real common-sense psychology, both theory and simulation aspects will be required and, therefore, a simulation theory will actually be complementary to, rather than alternative to, the models presented here (Perner, 1994). However, most of the people who have argued for a simulation theory have argued for it as an alternative to something

```
;;; These are the rules for answering questions about other
;;; people's beliefs. In this group, the "believes" question is
;;; correctly coupled to the believes predicate of a notional
;;; world. These rules override the default which gives the
;;; wrong answer in the first version of the situation theory.

;;; Rule answer-yes-other, compare to answer-yes-other
;;; in figure 10.
((result yes ?someone (believes ?object ?something)
   ?situation ^situation) :-
 (not (self ?object))
 (member (believes ?object ?something) ?situation)
 (write-list (yes ?object believes ?something)))

;;; Rule answer-no-other, compare to answer-no-other
;;; in figure 10.
((result no ?someone (believes ?object ?something)
   ?situation ^situation) :-
 (not (self ?object))
 (not (member (believes ?object ?something)
            ?situation))
 (write-list (no ?object does not believe ?something)))
```

Figure 11. Changes from the situation theory to the representation theory

like Leslie's 'decoupler' theory of mind mechanism, and therefore don't give much thought to how a simulation theory and a theory of mind mechanism might be combined in practice. But there is a twist to the simulation model; although it shows an apparent egocentricity, it can actually be functionally identical to Leslie's 'decoupler' model. This further backs up the arguments that the distinction between a theory and a simulation is one of interpretation rather than a real difference in behaviour (Perner, 1994).

It is, of course, possible to pursue this strategy still further developing models of some of the other models of common-sense psychology. Unfortunately, for an accurate model many of these require more complex models of perceptual apparatus (e.g. Baron-Cohen's, 1995, shared attention mechanism), or more complete models of common-sense psychology (e.g. Wellman's, 1990, simple-desire psychology) than have yet been developed within this framework. Even so, as a first attempt at the problem, the technique does seem to back up the existing points and arguments remarkably well, and to clarify the distinctions between the models which have been developed so far. And apart from anything else, at least within this limited scenario, it seems to work!

The usefulness of the modelling approach as a tool for studying common-sense psychology is a topic which deserves fuller discussion than is possible here. Even so, we believe that these models show cognitive modelling can help in this area.

REFERENCES

Astington, J. W., & Gopnik, A. (1991). Theoretical Explanations of Children's Understanding of the Mind. *British Journal of Developmental Psychology, 9*, 7-31.

Baron-Cohen, S., Leslie, A. M., & Frith, U. (1985). Does the Autistic Child have a 'Theory of Mind'? *Cognition, 21*(1), 37-46.

Baron-Cohen, S. (1995). *Mindblindness*. MIT Press.

Carey, S. (1985). *Conceptual Change in Childhood*. Cambridge, Massachusetts: MIT Press.

Chandler, M. J., & Boyes, M. (1982). Social-Cognitive Development. In B. B. Wolman (Ed.), *Handbook of Developmental Psychology* (pp. 387-402). Prentice-Hall.

Gordon, R. M. (1986). Folk Psychology as Simulation. *Mind and Language, 1*(2), 158-171.

Leslie, A. M. (1987). Pretence and Representation: the Origins of 'Theory of Mind'. *Psychological Review, 94*(412-426).

Leslie, A. M. (1988). Some Implications of Pretense for Mechanisms Underlying the Child's Theory of Mind. In J. W. Astington, P. L. Harris, & D. R. Olson (Eds.), *Developing Theories of Mind*. Cambridge University Press.

McCarthy, J., & Hayes, P. J. (1969). Some Philosophical Problems From the Standpoint of Artificial Intelligence. In B. Meltzer & D. Michie (Eds.), *Machine Intelligence 4* (pp. 463-502). Edinburgh: Edinburgh University Press.

Perner, J. (1991). *Understanding the Representational Mind*. Cambridge, Massachusetts: MIT Press.

Perner, J. (1994). The Necessity and Impossibility of Simulation. In C. Peacocke (Ed.), *Objectivity, Simulation and the Unity of Consciousness* (pp. 145-154). Oxford University Press for the British Academy.

Premack, D., & Woodruff, G. (1978). Does the Chimpanzee Have a 'Theory of Mind'? *Behavioural and Brain Sciences, 4*, 515-526.

Samet, J. (1993). Autism and Theory of Mind: Some Philosophical Perspectives. In S. Baron-Cohen, H. Tager-Flusberg, & D. J. Cohen (Eds.), *Understanding Other Minds: Perspectives from Autism* (pp. 427-449). Oxford: Oxford University Press.

Shultz, T. R. (1988). Assessing Intention: A Computational Model. In J. W. Astington, P. L. Harris, & D. R. Olson (Eds.), *Developing Theories of Mind*. Cambridge: Cambridge University Press.

Shultz, T. R. (1991). From Agency to Intention: A Rule-Based Computational Approach. In A. Whiten (Ed.), *Natural Theories of Mind: Evolution, Development and Simulation of Everyday Mindreading* (pp. 79-95). Oxford: Basil Blackwell.

Wellman, H. M. (1990). *The Child's Theory of Mind*. Cambridge, Massachusetts: MIT Press.

Wimmer, H., & Perner, J. (1983). Beliefs about Beliefs: Representation and Constraining Function of Wrong Beliefs in Young Children's Understanding of Deception. *Cognition, 13*, 103-128.

Zaitchik, D. (1990). When Representations Conflict With Reality: The Preschooler's Problem With False Beliefs and 'False' Photographs. *Cognition, 35*(41-68).

```
;;; Trace output for Leslie's 'decoupler' model, simulation
;;; theory (final version), and situation theory (final
;;; version). Compare to the results of Baron-Cohen et al.'s
;;; (1985) false belief test.

yes alison believes (inside marble box)
yes sally believes (inside marble basket)
yes anne believes (inside marble box)

;;; Trace output for simulation theory (first version).

yes alison believes (inside marble box)
no sally does not believe (inside marble ?where)
no anne does not believe (inside marble ?where)

;;; Trace output for copy theory and situation theory (first
;;; version).

yes alison believes (inside marble box)
yes sally believes (inside marble box)
yes anne believes (inside marble box)
```

Figure 12. Trace output from the different models

Initial explorations of modifying architectures to simulate cognitive and perceptual development

Gary Jones
Department of Psychology
University of Nottingham
Nottingham NG7 2RD, UK
+44 (0) 115 951 5361
gaj@psychology.nottingham.ac.uk

Frank E. Ritter
Department of Psychology
University of Nottingham
Nottingham NG7 2RD, UK
+44 (0) 115 951 5292
frank.ritter@nottingham.ac.uk

ABSTRACT

We modified a cognitive architecture (ACT-R) and an attached interaction architecture (the Nottingham interaction architecture) to simulate developmental changes in problem solving. We started with an existing model that fits adult data on a blocks world task used to study the development of problem solving in children. We modified the model and architectures in three, independent ways to simulate a younger problem solver: (a) reduced the working memory, (b) deleted a piece of knowledge, and (c) reduced the accuracy of vision. We found that our modifications allowed the model to fit 7 year old's data better but not perfectly. These results suggest that cognitive models and their architectures can help answer the question of "What develops?"

Keywords

Cognitive architectures, development, problem solving, working memory, vision, ACT-R, interaction.

INTRODUCTION

As children grow older, they tend to be more able to learn new strategies and tasks, and be more efficient at those strategies and tasks that they knew previously (e.g. Siegler, 1986). What changes are occurring in order for this to happen? It would be useful to be able to specify in information processing terms how the behaviour seen at each age is achieved, and therefore what the differences are between ages (Simon, 1962).

The solving of physical puzzles is a good area in which to examine differences in behaviour. A detailed analysis of the task behaviour is possible via videotape. Many strategies will be readily visible, reducing the need for the experimenter to infer what mental structures and strategies are being used. For this reason, a physical problem solving puzzle, the "Tower of Nottingham", is used to study differences in children's behaviour and the factors influencing them.

The Tower of Nottingham

The Tower of Nottingham task involves building a pyramid from 21 wooden blocks (see Figure 1). There are six layers to the pyramid, the lower five consisting of four blocks each, with a single block as the top layer. The blocks in the lower five layers all share the same characteristics, differing only in size. Each layer is normally formed via two sets of paired blocks. For example, placing the peg of block A into the hole of block B brings the two half holes together to form a pair having a hole (a hole-pair). Similarly, placing block C and block D together forms a pair with a peg (a peg-pair).

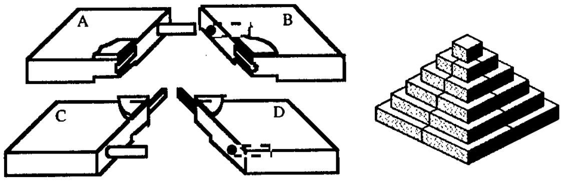


Figure 1. The blocks, on the left, that make up each layer, which are then stacked to create a tower, shown on the right.

Other strategies for creating a layer also exist, however, such as forming a pair having two pegs (blocks A and C) and a pair having two holes (blocks B and D).

There are two other features that may give rise to additional construction strategies. Each block has a quarter circle indent on top and a quarter circle depression underneath. When a layer is created, the quarter circles form circles in the centre such that layers can be stacked on top of each other by placing the circular depression of the upper layer onto the circular indentation of the lower layer. Constructions can be created by aligning the quarter circles so that they form a semi-circle.

Behaviour on the Task Varies with Age

Children of three are able to complete the Tower of Nottingham, yet performance improves with age all the way up to adulthood. For example, older children on the task accomplish more correct operations, produce less errors and take less time than their younger counterparts (Murphy & Wood, 1981; Wood & Middleton, 1975). Studying performance across ages on this task allows us to examine problem solving behaviour at each age and the differences in problem solving between ages.

The Use of Cognitive Models and Cognitive Architectures

Computational modelling across ages requires defining the behaviours that occur at each age (or performance level), because the model will require the knowledge and procedures that children may be using at each age. Where the behaviour cannot be defined in these terms, the model makes predictions about the missing elements. Therefore modelling task behaviour can help provide a means of defining how the different behaviours are generated.

This enables a method for examining to what extent changes in task performance can be attributed to differences in knowledge and to what extent changes in task performance can be attributed to developmental processes. Existing models of development have only really considered differences in knowledge as the reason for changes in task performance, and have largely ignored the developmental processes that various developmental theories put forward (e.g. changes in working memory).

Early production system models of development, such as that of Young (1973), model differences in task performance by altering the rule set (i.e. the knowledge) within the production system. Klahr and Wallace (1976) implement possible developmental factors in their production system model of development (such as visual memory), but do not explore their effects.

Modelling techniques which have not used the production system style view development as being experience with the task, which can be seen as implicit knowledge. In the connectionist model of McClelland and Jenkins (1991), improved performance is attained by further training of the network on the task. In Siegler and Shipley's (1995) Adaptive Strategy Choice Model, improved performance is achieved by the model learning through experience of the task which strategies to employ for which sums.

All of these models have had success when they have been compared to subject data. However, developmental theory suggests that there are further changes occurring that also influence development. To what extent are these changes able to influence performance?

Two approaches stand out for creating a model of our task. One method is to model a lower performance level and see if that model can then progress to the higher performance levels that we see on the task. The other method is to begin at the highest performance level (that of adults), and then see if reduced versions of this model show behaviour that looks like lower performance levels. We have chosen to start with the simpler (adult) behaviour and work towards the more chaotic (child-like) behaviour.

We wish to examine how changes in both knowledge *and* development can influence task performance. To do this, we will begin with an adult model of our task and then impair it in theoretically motivated ways. By examining performance of the model after these changes, we hope to see to what extent the impairment can account for lower performance levels (those of children).

Cognitive architectures are important here as well, for they should also guide us (together with developmental theory) as to what are the sensible changes to make to the architecture. However, the role of change in architectures, with particular reference to development, has been rarely studied. The first definitions and implementations of cognitive architectures stressed that architectures do not change across tasks (Newell, 1990, p. 81). Newell (1990) argues that within Soar, development is just learning, and the architecture remains the same. Development is not mentioned with respect to ACT-R (Anderson, 1993). For these reasons we will look towards developmental theory as to what changes to make to the architecture.

Overview of the Paper

In the remainder of this paper, we will first describe the adult model upon which we base the other models. We describe its structure and the set of blocks that it interacts with. The model has been improved since it was last reported (in Jones & Ritter, 1997), and although the fit to the data is not improved substantially, it does enable the model to be broken in more theoretically motivated ways. We therefore describe the model in detail here. The stage is thus set for describing the three changes we make to the architecture. Each of the changes is described in terms of why they are suggested by developmental data, how they have been implemented, either in ACT-R or the Nottingham interaction architecture, and the effect they have on the model's behaviour. We conclude with a summary of these changes and the implications they have for the disentangling of what changes in cognitive development.

THE ADULT MODEL

The adult model is based on the ACT-R cognitive architecture (Anderson, 1993). In the development of the adult model the architecture has in part been used as a vehicle for the development of our own theories of performance on the task, although the model is consistent with most of the principles of ACT-R such as being goal driven, giving activation to memory elements, subjecting activation to both decay and noise, being rule based, and so on.

A simulation of the task also exists (see Figure 2), which is written in Garnet (Myers, et al., 1990). The simulation contains a full graphical representation of the task (all blocks and features), which is 2 1/2 dimensional—blocks cannot be turned on their side or held in mid-air, but can be face-up or face-down.

The simulation also represents an eye and two hands. The eye and hands are designed to meet a set of requirements identified for creating a psychologically plausible architecture for interacting with an external task (Baxter & Ritter, 1996).

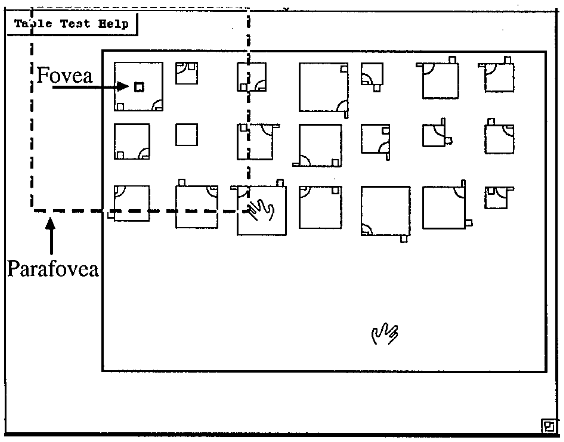


Figure 2. The Tower of Nottingham interaction interface.

The eye is able to saccade and fixate, and passes to the model what it sees with regard to blocks and constructions (e.g. a peg-pair will be represented as a construction having two blocks that are flush on their outer edges and have their quarter circles and halfpegs aligned).

The visual information passed to the model is based upon where blocks are positioned in relation to the fovea. Three areas are defined: fovea, parafovea and periphery. Full information is passed for blocks or features in the fovea and parafovea, though the parafovea subjects features and block sizes to noise. For items in the periphery, the eye only returns to cognition a block ID. The hands are able to pick up, drop, rotate, turn over, fit, and disassemble blocks.

The model contains 226 rules which allow it to complete the task. The rules also interact with the simulation of the task, directing the eye and the hands. Within the model, all blocks and block features have an associated activation level. When several rules are instantiated, the one with the highest activation is selected. Therefore, in general, rules fire whose conditions have the most active blocks and block features in them. The activation levels are subject to *decay* each cycle, such that when they fall below a specified level (the *retrieval threshold*) they can no longer be matched in conditions of rules. Activation is raised based on what the goals of the model currently are, and by what blocks the fovea is looking at.

The learning mechanism that we included in the architecture is a simple method of increasing the chances of fitting blocks by specific features if a previous fit using the same features was deemed a success. Success is determined by the blocks in the construction being flush on their outer edges and having their quarter circles aligned (this is consistent with adult data on the task). Therefore, on some occasions the model may believe a successful construction has been made when in fact it has not (e.g. aligning the quarter circles of blocks A and B such that the blocks are not connected via a peg/hole). This learning mechanism approximates adult learning on the task (Jones & Ritter, 1997).

The model contains working memory and visual memory. Working memory contains all blocks and block features that are active enough to be matched in the conditions of rules (i.e. their activation is above retrieval threshold). Therefore, working memory is variable based on how active blocks and block features are in the model. Visual memory means we can remember some of the blocks that have been looked at previously even though they are now in the periphery. Visual memory is static (it is set at seven items), and compliments working memory since blocks in visual memory that are not in working memory can also be matched in conditions of rules.

Comparing the models with the data

It would be useful to compare subject performance on the Tower of Nottingham with the performance of models of the task using a metric that cannot be set as a parameter of the architecture. One such metric is the proportion of productions fired in the construction of each layer compared to the proportion of time subjects take in the construction of each layer. However, the task involves interaction with an external world, so timings for subjects include their perceptual and motor actions whereas the model production firings do not. This means timing estimates for interaction must be used in part of our model/subject comparisons.

We use the ACT-R default timing of 50 ms per production firing, which increases to 250 ms (Baxter & Ritter, 1996) for productions involving perceptual actions (eye movements and fixations), and 550 ms (Jones & Ritter, 1997) for productions involving motor actions (fitting and disassembling blocks). This enables a more complete comparison between model and subject timings. Production firing latencies in ACT-R also take into account activation of memory elements. In order for the influence of memory elements on production firing latencies to be negligible, the base level activation of memory elements was set to 10.0. Where other ACT-R parameters were used (decay, retrieval threshold), we adhered to the suggested default settings. The models begin with the initial knowledge of the task that subjects had, such as blocks of the same size go together, pegs go in holes, etc.

For every run of the model, the activation noise parameter within ACT-R was set to 0.005. This causes the activation of constructions and features in the model to differ, making the model's behaviour variable.

For comparisons between the model and subjects, measurements are given on an overall and layer-by-layer basis. The reason for reporting times and errors per layer is that subjects learn throughout the task. Since the model includes a learning mechanism, we want to see not only the effect that impairment to the model has upon overall behaviour, but also the impact it has upon the learning of the task.

We provide r-squared estimates for correlations between the model and subjects on a layer-by-layer basis, and t-test comparisons for summary data. These should only be taken as initial guides to the quality of the fit between the model and the subject data.

Comparison of the model with adult subjects

The adult subjects (N=5; taken from Jones & Ritter, 1997) had completed the task once. We compare 5 runs of the model to the 5 adult subjects.

The comparison of the adult model to the adult subject data is favourable. On the measures we will be using when we break the adult model, it fits the adult subjects reasonably well (see Table 1), although the model makes more incorrect constructions than subjects.

If we compare the times to complete each layer for the adult model and the adult subjects (see Figure 3), the trend of the model is the same as subjects—the time to complete each layer decreases until the final layer where the time increases slightly ($r^2 = 0.92$). The model takes more time to complete the task because it makes slightly more errors (see Figure 4; $r^2 = 0.67$).

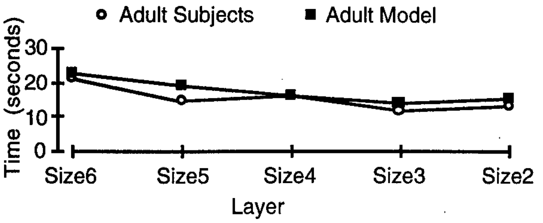


Figure 3: Time taken (seconds) to complete each layer for adult subjects and the adult model.

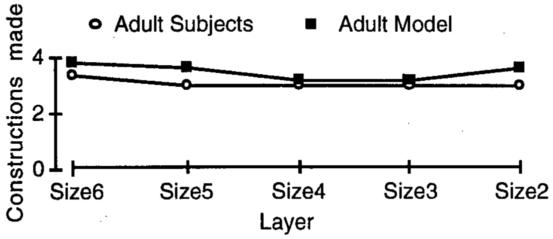


Figure 4: Construction attempts to complete each layer for adult subjects and the adult model.

The model provides a reasonable fit to the adult subject data in most of the behaviours that we are presently interested in. An exact fit on every measure is not essential because we will be examining the relative increases and decreases of these measures that result from the alterations that we carry out. The model fits the data well enough that it is fruitful to start exploring how problem solving changes when the architecture is changed to reflect that of younger problem solvers.

CHANGES TO THE ADULT MODEL

In order to examine how problem solving could change with development, we created three changed versions of the adult model. These changes are the most plausible based on the developmental literature and our knowledge of children's performance on the task. (a) We reduced the working memory capacity. (b) We removed a piece of knowledge. (c) We altered the accuracy of the parafovea. There are further changes that should be explored as well, such as basic processing speed, fovea size, and further changes to knowledge.

In this initial exploration we made each of these changes independently in order to keep the first order effects clear. For each change we explain its implementation, its rational, and its effect on problem solving.

The seven year olds we use to compare the altered models against were assisted on their first attempt at completing the Tower (contingently tutored, Wood & Middleton, 1975), and so we compared the model with their second attempt where they received no help in completing the Tower.

Reduced Working memory capacity model

Why

Several developmental theories suggest working memory capacity may influence task performance (e.g. Case, 1985; Halford, 1993). On the Tower of Nottingham, children have been noted to search with replacement (D.Wood, personal correspondence), a characteristic which may well be linked to working memory in that the children forget which blocks they have tried fitting together. On the Tower of Nottingham, seven year old children fit the same blocks together an average of 3.68 times, whereas this behaviour never occurs for adults completing the task.

How

Our model provides an easy way to manipulate working memory capacity to see what effect it has upon performance. In order to get a large, initial effect, we implemented this change to the model in three ways (the first two are parameters in ACT-R and the third is a parameter in the Nottingham interaction architecture). First, raising the retrieval threshold (from 0.0 to 2.5) means that constructions need to be higher in activation than in the adult model in order to be matched in rules. Second, raising decay (from 0.05 to 0.15) means constructions are forgotten more quickly than in the adult model. Third, reducing the number of items in visual memory (from 7 to 3) means that visual memory provides less support to working memory. The ACT-R parameters and mechanisms that we manipulate have also been used by Lovett, Reder and Lebiere (1997) in their ACT-R model of working memory differences, although they kept the parameter values constant and manipulated a third parameter. In this way they were able to model individual differences in working memory.

Measure	Adult Subjects	Adult Model	t-score
Total time taken to complete the Tower	80.6 s (13.3)	92.2 s (9.47)	t(8)=1.59 p>0.05
Total number of errors (incorrect constructions) made	0.2 (0.45)	2.4 (1.14)	t(8)=4.017 p<0.05
Errors where the blocks involved are of the same size	0.2 (0.45)	2.4 (1.14)	t(8)=4.017 p<0.05
Errors where the blocks involved are of different sizes	0	0	N/A
Number of times a construction attempt is made using the same blocks	0	0	N/A

Table 1: Mean (standard deviation) and t-scores for adult model and adult subject comparisons.

Measure	7yo Subjects	Reduced WM Model	t-score
Total time taken to complete the Tower	214.4 s (95.81)	134.0 s (24.1)	t(8)=1.82 p>0.05
Total number of errors made	7.6 (2.41)	5.4 (2.88)	t(8)=1.31 p>0.05
Number of times the same blocks are fitted together	1.75 (0.96)	2.0 (1.41)	t(4)=0.27 p>0.05

Table 2: Comparison between seven year old subjects and the reduced working memory model. Standard deviations, where appropriate, are given in parentheses.

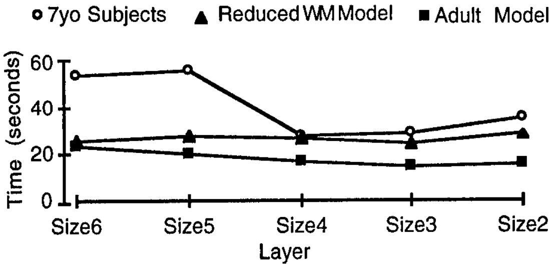


Figure 5: Time taken (seconds) to complete each layer.

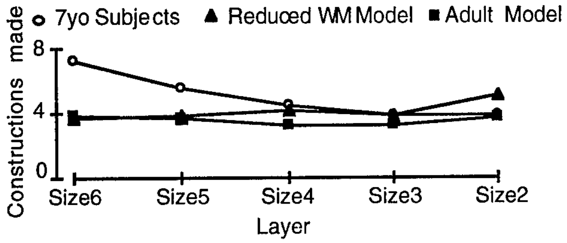


Figure 6: Construction attempts to complete each layer.

Predicted effect

Less working memory should lead to more search with replacement—the same pairs of blocks should be fitted together more often. A side-effect of searching with replacement is that the task should take longer and involve more errors.

Effect

Table 2 shows the summary statistics for the seven year old subjects and the reduced WM model. Figures 5 and 6 show comparisons on a layer by layer basis.

As predicted, reducing the working memory capacity in the adult model leads to fitting the same blocks together more often (from 0 in the adult model to 2.0 in the reduced WM Model). Increases are seen in both the time to complete the task (from 92.2 s in the adult model to 134.0 s in the reduced WM Model) and the number of errors (from 2.4 in the adult model to 5.4 in the reduced WM Model). This increase is not enough for the reduced WM Model to appear like a seven year old on the task. Although there are no reliable differences between the reduced WM Model and seven year olds in the total time taken and total number of errors, there are clear differences in the magnitude of these totals.

On a layer by layer basis, the reduced WM Model can be seen to not differ greatly from the adult model in terms of time and construction attempts made. However, the learning mechanism seems to be affected by the reduction in working memory capacity, because the original adult model provides a better fit to the seven year old subject data (times $r^2 = 0.85$; constructions $r^2 = 0.74$) than the reduced WM Model does (times $r^2 = 0.24$; constructions $r^2 = 0.63$). The original adult model and the reduced WM Model do not correlate at all (times $r^2 = 0.07$; constructions $r^2 = 0.05$).

Reducing the working memory capacity has allowed the model to fit the seven year old data a lot better than the adult model for overall times and errors, but at the cost of impeding the learning mechanism. This is probably because of the type of learning mechanism we use: there are less block features to be raised in activation upon success because working memory capacity is smaller. This suggests that further learning mechanisms must be used in order to fit the seven year old subject data better.

Less Knowledgeable model

Why

Children have a much smaller knowledge base to draw upon than do adults (e.g. Siegler, 1986). It is quite possible that children's knowledge of the Tower of Nottingham is less than that of adults. Examination of how seven year olds produce correct constructions compared to how adults produce correct constructions reveals that the children fit pegs into holes to produce a pair on 37 occasions yet only fit a halfpeg into a halfhole on 6 occasions. Adults fit via a peg and hole on 26 occasions as compared to fitting by halfpeg and halfhole 14 times. It is a possibility that children only learn about halfpegs and halfholes fitting together whilst they are completing the task.

How

Previously the model knew that halfpegs could fit into halfholes. This knowledge was deleted from the model.

Predicted Effect

The effect this will have upon performance is unclear. The number of constructions made via a peg and hole will rise sharply; however, the current learning mechanism offers no opportunity for learning that halfpegs and halfholes can fit together, and therefore it is expected that fitting by halfpegs and halfholes will be dramatically reduced. It will not be eradicated because there are other ways in which constructions can indirectly be made via a halfpeg/halfhole (e.g. quarter

Measure	7yo Subjects	Less Knowledge-able Model	t-score
Total time taken to complete the Tower	214.4 s (95.81)	164.8 s (40.4)	t(8)=1.07 p>0.05
Total number of errors made	7.6 (2.41)	5.6 (3.36)	t(8)=1.08 p>0.05
Ratio of correct constructions fitted via peg/hole:halfpeg/halfhole	37:6	31:6	N/A

Table 3: Comparison between seven year old subjects and the less knowledgeable model. Standard deviations, where appropriate, are given in parentheses.

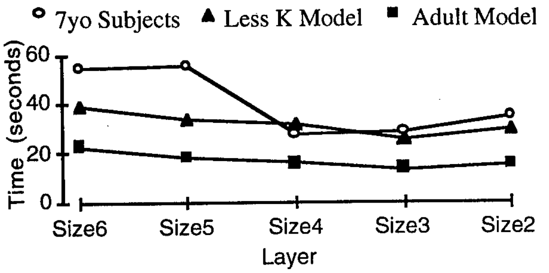


Figure 7: Time taken (seconds) to complete each layer.

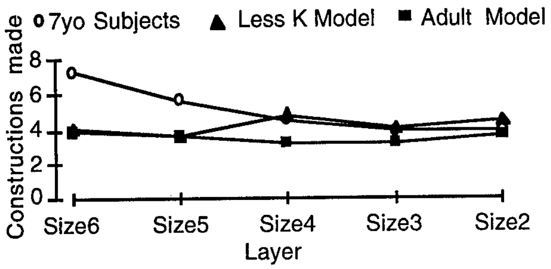


Figure 8: Construction attempts to complete each layer.

circles can be aligned in such a way that the halfpeg and halfhole fit together). We predict that the number of errors will remain the same. This is because fitting random blocks of the same size by a peg/hole arrangement and by a halfpeg/halfhole arrangement offer the same chances of success. The time to complete the task should not change, because no more errors are expected.

Effect

Table 3 shows the summary statistics for the seven year old subjects and the less knowledgeable (Less K) model. Figures 7 and 8 show comparisons on a layer by layer basis.

As predicted, deleting the knowledge that halfpegs fit into halfholes meant that fitting by pegs and holes rose sharply (from 14 in the original adult model to 31 in the less K Model), and fitting by halfpegs and halfholes dropped but was not eradicated (from 15 in the original adult model to 6 in the less K Model). The ratio of 31:6 compares favourably with the 37:6 ratio of seven year olds.

There were increases in both the total time taken to complete the task (from 92.2s in the original adult model to 164.8s in the less K Model), and the number of errors produced in completing the task (from 2.4 in the original adult model to 5.6 in the less K Model). This helps the less K model to fit the seven year old data (there are no reliable differences between the summary measures for the less K model and seven year old subjects, although there are clear differences on the layer-by-layer plots). Part of the increase in time can be attributed to more search being required (as we now have a reduced feature set because we no longer know that halfpegs fit into halfholes). However, most of the increase in time is because more errors are made. We do not yet have a valid reason for why this occurs.

As with the reduced WM model, we again see that the original adult model correlates better with the seven year old data on a layer by layer basis (original model and seven year olds: $r^2 = 0.85$ for times and $r^2 = 0.73$ for constructions; less K model: $r^2 = 0.73$ and $r^2 = 0.44$ respectively). This again suggests that the learning mechanism is impeded by the removal of knowledge. The type of knowledge removed means that learning must now occur over a reduced feature set. However, the reduced feature set still has the same chance of success as the old set, and it is therefore difficult to explain why the less K model does not learn as well as the original adult model.

Reduced Parafovea accuracy model

Why

Children find it more difficult to select blocks by size in the Tower of Nottingham task (Murphy & Wood, 1981). Although this is more pronounced for children of five years of age and below, seven year olds still average 1.8 constructions involving different sized blocks; the adults do not make any constructions involving blocks of different sizes.

How

We set the parafovea noise parameter for size to be 30 percent, representing a 30 percent chance that a block in the parafovea will be perceived as being a different size than it actually is (there are other possible mechanisms to implement this).

Predicted Effect

The increased size noise should mean that more incorrect constructions are produced involving blocks of different sizes. This increase in error should also lead to an increase in the time taken to construct each layer.

Effect

Table 4 shows the summary statistics for the seven year old subjects and the parafovea accuracy model. Figures 9 and 10 show comparisons on a layer by layer basis.

Measure	7yo Subjects	Reduced Parafovea Accuracy Model	t-score
Total time taken to complete the Tower	214.4 s (95.81)	126.2 s (24.6)	t(8)=1.99 p>0.05
Number of errors involving blocks of the same size	5.8 (2.59)	3.4 (1.34)	t(8)=1.84 p>0.05
Number of errors involving blocks of a different size	1.8 (2.68)	0 (0)	N/A

Table 4: Comparison between seven year old subjects and the reduced parafovea accuracy model. Standard deviations, where appropriate, are given in parentheses.

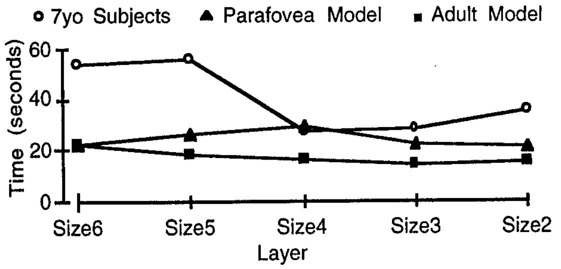


Figure 9: Time taken (seconds) to complete each layer.

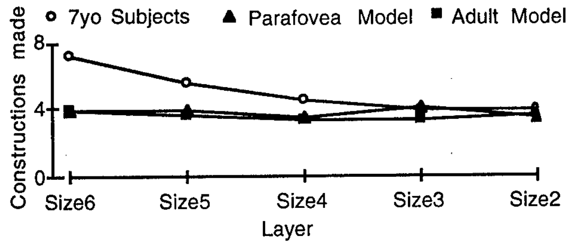


Figure 10: Construction attempts to complete each layer.

The results found go against our main prediction that there will be a greater number of constructions made which involve blocks of different sizes (neither the original adult model or the reduced parafovea accuracy model produce any). In hindsight, the reason for this is that when picking up a block, the model fixates upon it. Since at this point the block is in the fovea, the correct size is returned, and therefore if the block is the wrong size it is replaced. This provides an interesting result because it indicates that seven year olds either do not examine the block again once they have decided to pick it up, or their fovea vision is not as accurate as adults.

As predicted, there is an increase in the overall time taken (from 92.2 s for the original adult model to 126.2 s for the reduced parafovea accuracy model) and the number of errors produced (from 2.4 for the original adult model to 3.4 for the reduced parafovea accuracy model). This increase is not sufficient enough to make the reduced parafovea accuracy model appear to be like seven year old subjects on the task, although there are no reliable differences for either measure.

The reduced parafovea accuracy model does not correlate well with either the original adult model ($r^2 = 0.05$ for times; $r^2 = 0.03$ for constructions) or the seven year old subjects ($r^2 = 0.13$ and $r^2 = 0.29$ respectively). The increase in overall timings is probably due to the increase in visual search that is required due to the parafovea being less accurate. There should be no reason other than chance that there is an increase in construction attempts over the original adult model.

SUMMARY

We took an initial adult model and broke it in three ways to simulate a younger problem solver: cognitively (reducing working memory capacity), via knowledge (removing knowledge), and perceptually (reducing parafovea accuracy). All of these impaired the performance of the model to differing degrees and in different ways. None of the alterations was sufficient to produce behaviour similar to seven year old subjects, and all of the alterations indicated that more than one learning mechanism is required to fit the seven year old data properly. However, in breaking the adult model, we were able to show that changes that have been hypothesised to exist in younger problem solvers (i.e. developmental factors) do lead to different problem solving behaviour.

Further work must modify the model and its architecture in additional ways, motivated by developmental theory. There are several other ways to degrade the model's performance that we have not yet explored, such as changes in processing speed. These explorations will allow us to see how much each factor influences performance. The extent to which each factor contributes toward the observed behaviour indicates where our attention must lie in creating a complete model of seven year olds that is comparable and related to adult behaviour on the Tower.

However, we cannot simply consider each influencing factor independently because we have shown that this is not sufficient to produce the behaviour of seven year old subjects. The adult model will need several interacting changes to its architecture before its behaviour appears realistically to be like a younger problem solver. Therefore, not only will we be breaking the model in additional, independent ways, we will also be looking at combinations of modifications that interact. We expect the interactive effects to reveal more about performance at different ages, but simple changes are still required for our understanding and initial explorations.

This work indicates that the role of change in architectures, which has been little studied since the first definition, can be a fruitful way to use architectures. ACT-R includes many parameters. Before these parameters can be easily used for modelling development and abnormal problem solving, they need to be explored (or explained) to the extent that ranges for normal individual differences are known (e.g. Lovett, Reder, & Lebiere, 1997), and then that the interactions of these parameters are understood. A way to predict the performance of ACT-R models without running them in this area would be useful.

This work will eventually lead to models of five year old's and seven year old's behaviour solving the Tower that are based on modifying the adult model. We hope that these models will be able to explain individual differences within age groups as well as to explain the progression between ages (in terms of differences between the models rather than transition mechanisms). In both cases, we should be able to highlight the knowledge differences or architectural changes that lead to the differences in behaviour. Further learning mechanisms are also required in order that each model can learn from the task in order to perform to the standard of the older models. Explaining how and why problem solving changes with development is difficult, so further work will have to look at more than just this task.

We are now in a position to look at how problem solving changes across development. We have a cognitive model that performs the task. We can add and remove knowledge from the cognitive model and we can modify the architecture to represent developmental changes in cognition (the cognitive model based in ACT-R) and perception (the Nottingham interaction architecture). In the future we may be able to more directly answer "What develops?"

ACKNOWLEDGEMENTS

Josef Nerb wrote the first implementation of the Tower of Nottingham simulation. Heather Wood kindly supplied videos of seven year olds completing the Tower of Nottingham for us to analyse, and provided the tower picture. David Wood provided numerous useful discussions.

REFERENCES

Anderson, J. R. (1993). *Rules of the mind.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Baxter, G. D., & Ritter, F. E. (1996). Designing abstract visual perceptual and motor action capabilities for use by cognitive models (Tech. Report No. 36). ESRC CREDIT, Psychology, U. of Nottingham.

Case, R. (1985). Intellectual development: A systematic reinterpretation. New York: Academic Press.

Halford, G. S. (1993). *Children's understanding: The development of mental models.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Jones, G., & Ritter, F. E. (1997). Modelling transitions in children's development by starting with adults. In *Proceedings of the European Conference on Cognitive Science (ECCS '97).* 62-67. Manchester, UK: AISB.

Klahr, D., & Wallace, J. G. (1976). Cognitive Development: An information processing view. Hillsdale, NJ: Lawrence Erlbaum Associates.

Lovett, M. C., Reder, L. M., & Lebiere, C. (1997). Modeling individual differences in a digit working memory task. In M. G. Shafto & P. Langley (Eds.), *Proceedings of the Nineteenth Annual Conference of the Cognitive Science Society.* 460-465. Mahwah, NJ: Lawrence Erlbaum Associates.

McClelland, J. L., & Jenkins, E. (1991). Nature, nurture and connections: Implications of connectionist models for cognitive development. In K. VanLehn (Ed.) *Architectures for intelligence,* Chapter 3.

Murphy, C. M., & Wood, D. J. (1981). Learning from pictures: The use of pictorial information by young children. *Journal of Experimental Child Psychology, 32*, 279-297.

Myers, B. A., Guise, D. A., Dannenberg, R. B., Vander Zanden, V., Kosbie, D. S., Pervin, E., Mickish, A., & Marchal, P. (1990). Garnet: Comprehensive support for graphical, highly-interactive user interfaces. *IEEE Computer, 23*(11), 71-85.

Newell, A. (1990). *Unified theories of cognition.* Cambridge, MA: Harvard University Press.

Siegler, R. S. (1981). Developmental sequences within and between concepts. *Monographs of the Society for Research in Child Development, 46 (No. 189, 1-74).*

Siegler, R.S. (1986). Children's thinking. Englewood Cliffs, NJ: Prentice-Hall.

Siegler, R. S. & Shipley, C. (1995). Variation, selection and cognitive change. In T. J. Simon & G. S. Halford (Eds.), *Developing cognitive competence.* Hillsdale, NJ: Lawrence Erlbaum.

Simon, H. A. (1962). An information processing theory of intellectual development. *Monographs of the Society for Research in Child Development, 27 (2, Serial No. 82).*

Wood, D., & Middleton, D. (1975). A study of assisted problem solving. *British Journal of Psychology, 66*(2), 181-191.

Young, R. M. (1973). Children's seriation behaviour: A production-system analysis. Unpublished doctoral dissertation, Carnegie-Mellon University, 1973.

How to Fatigue ACT-R?

G.M.G. Jongman
Experimental & Work Psychology
University of Groningen
Grote Kruisstraat 2/1, 9712 TS Groningen, The Netherlands.
+31 50 3636289
Linda@tcw3.ppsw.rug.nl

ABSTRACT

In this paper, an ACT-R model of mental fatigue is presented. This model is loosely based on Hockey's state regulation model of compensatory effort (Hockey, 1997). It appears that when spreading of activation is reduced, the ACT-R model can predict the performance changes Hockey describes, and furthermore, show how these may depend on the motivation of the participant. In a model of the Sternberg memory-search task, a reduction of the spreading of activation results in a change in strategy.

Keywords
mental fatigue, strategy use, cognitive control, ACT-R

INTRODUCTION

This paper describes a computational approach towards the investigation of mental fatigue. Mental fatigue is defined as the deterioration of mental performance due to preceding exercise of mental or physical activity (Meijman, 1997). As Meijman explains, it can be conceived of as a problem of keeping attention focused on task goals, or as a deficit in the cognitive-energetic control mechanisms. From his research it appeared that in some task conditions fatigued participants could protect their performance by means of compensatory effort, but in the most unfavourable conditions of the experiment (after 8 hours of work combined with sleep loss) people were no longer able to prevent deterioration of their performance. According to Shiffrin & Schneider (1977) there are two types of information processing: automatic and controlled. It appears that tasks that require more controlled processing are more sensitive to mental fatigue (Meijman, 1997). However, which cognitive processes are responsible for the changes in behaviour which are observed when people have to perform tasks for an extending period of time is a question that has not been answered yet. Bartlett (1943) hypothesised that the processes involved in planning, which is often ascribed to prefrontal functioning, are the ones responsible for these changes in behaviour. West (1996) subdivides the functioning of the prefrontal cortex into three processes. The first one is the inhibition of interfering processes and stimuli. The second process is a working memory process which enables the retrieval of information. The third process involves the preparation of responses. Summarising, there is some evidence that indicates mental fatigue is related to problems with cognitive control.

From many previous studies we already know that people seldom show a total breakdown of performance when they become mentally fatigued. A possible explanation for maintaining adequate task performance is that people change their strategy. More than 20 years ago, Shingledecker and Holding already hypothesised that when people become mentally fatigued they will shift their strategy of task performance towards a strategy that requires less mental effort (Shingledecker & Holding, 1974). In 1997, this hypothesis was brought out again by Hockey (1997). So, some people have hypothesised that mental fatigue involves a change in choice. However, a controlled study that investigates the details of this possible relation between mental fatigue and strategy use, still has to be done.

In order to predict and explain the role of cognitive control and strategy choice on the performance changes associated with mental fatigue, it is necessary to construct a detailed model of how these processes take place, and how they are influenced when people become fatigued. As the models mostly used in this field are mainly descriptive, the main purpose of this paper is to show how the valuable aspects of one of these models can be used to construct a computational model of mental fatigue, from which it will be possible to derive useful predictions of participants' behaviour. To this end, the next paragraphs will describe Hockey's compensatory control model (Hockey, 1997), which is a commonly known descriptive model of mental fatigue, and a cognitive architecture, ACT-R (Anderson, 1983; 1993). Together these components will be the basis for a computational model of mental fatigue.

A DESCRIPTIVE MODEL OF MENTAL FATIGUE

A model currently used for the investigation of mental fatigue is the state regulation model of compensatory control (Hockey, 1997). It is based on the concept of resources, which is described as "the availability of one or more pools of general-purpose processing units, capable of performing elementary operations across a range of tasks, and drawing upon common energy" (Gopher, 1986; Kahneman, 1973; Wickens, 1984). The model makes three assumptions. Firstly, it assumes that behaviour is goal-directed. Further it is assumed that the control process is normally self-regulating. And, thirdly, the model assumes this regulation has costs (expressed in use of mental resources, levels of subjective strain, and physiological changes). An overview of the model is presented in figure 1.

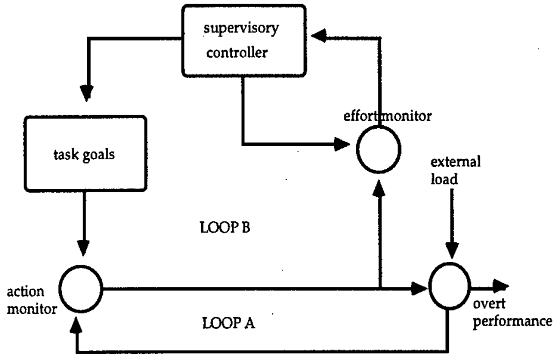


Figure 1. The state regulation model of compensatory control (Hockey, 1997)

The model distinguishes between two levels of control: a lower level, representing routine regulation (loop A), and an upper level, representing effort-based regulation (loop B). The effort-monitor monitors the level of demands in the lower loop. When the demands of the situation change, control will shift to the higher level (here called the supervisory controller) where several options for regulation are available. The model requires two levels for the effort monitor: a lower setpoint and an upper setpoint. This is the part of the model in which resources play an important role, for the upper setpoint represents the maximum level of effort that can be mobilised, which is dependent on motivation. Referring to Holding (1983), Hockey argues that this upper setpoint can be influenced by fatigue. When the perceived demands are too high, the maximum level of effort that can be mobilised should be increased, or the performance will decrease. Hockey describes four kinds of changes that can happen when people protect their performance. The first change he mentions is subsidiary task failure, for example the neglect of subsidiary activities or narrowing of attention. Second, people can make strategic adjustments as less use of working memory and greater use of closed-loop control. Third, maintaining performance could require compensatory costs. People would have to increase mental effort to attain the same performance. Finally, if no changes during task performance are observed, it is possible that people will show after-effects, for example express feelings of fatigue, or show a post-task preference for low-effort strategies.

To summarise, according to this model, task performance normally relies on routine regulation. In situations with high demands (e.g., stressful situations, situations in which the operator is mentally fatigued), task performance requires effort-based regulation (loop B). Thus, the model would predict that when people become mentally fatigued they would need a more effortful manner of control for the same task as before. However, it is not clear how that would lead to the four kinds of change Hockey predicts. It could be the case that when people become fatigued, they invest more effort in the task, change their strategy of performance, neglect subsidiary activities, or show after-effects. The model does not provide predictions about what people will actually do in these situations that require higher level control. A computational model is needed to refine these processes and deliver useful predictions for different situations. To this end, a rather brief explanation will be given of ACT-R (Anderson, 1993), an architecture of cognition, from which it is possible to construct a computational model of fatigue.

ACT-R

The reason for choosing the ACT-R architecture for the construction of a model of mental fatigue is twofold. For the investigation of mental fatigue the measurements of performance that are used most often are the reaction times for completing tasks, the (strategic) choices made during task performance, and the number of errors made by participants. A very attractive aspect of ACT-R is that it can make very detailed predictions about these three kinds of measurements. Furthermore, ACT-R is equipped with global parameters which, when changed, can cause qualitative, task-specific, changes in behaviour. These global parameters make ACT-R suitable for the construction of a model of mental fatigue.

The ACT-R Architecture

The ACT-R architecture distinguishes between two kinds of memory: production memory (memory for procedural knowledge, represented with production rules) and declarative memory (memory for fact knowledge, represented with chunks). Strategies are represented with (a number of) production rules, and additional declarative facts. The conflict resolution process selects production rules according their expected gain, as calculated by equation 1.

$$\text{Expected gain}_i = P_i G - C_i \qquad (1)$$

In this equation P represents the probability of success when using this production rule, G the value of the goal, and C the cost to reach the goal, using this production rule. The preliminary assumption of ACT-R is that cost is the time needed to reach the goal. From the production rules that match the current goal, the production rule that has the highest expected gain is tried first, which means that ACT-R tries to retrieve the declarative memory chunks necessary for the production to fire. Whether ACT-R succeeds in retrieving the chunks depends on the activation level of these chunks. When the activation of a chunk drops below a certain threshold, the *retrieval threshold*, it cannot be retrieved anymore. The activation level of a declarative memory chunk is determined by equation 2.

$$\text{Activation}_i = \text{base level activation}_i + \sum_j \text{source-activation}_j * \text{associative strength}_{ji} \quad (2)$$

In this equation base-level activation represents how recently and frequently the chunk has been used before. The second half of the equation represents spreading activation. Source activation represents the attention given to the elements of the goal and association strength represents the likelihood that fact i is needed if fact j is part of the current goal. If all retrievals succeed, the production will fire, if not, the second-best production is tried. Furthermore it must be mentioned that ACT-R can learn the parameters of the model itself (e.g., the base-level activation, the associative strengths, the probability of success of a production and its cost).

A COMPUTATIONAL MODEL OF MENTAL FATIGUE

In the introduction two aspects of mental fatigue were mentioned: mental fatigue as a cognitive control problem, and mental fatigue as a process involving a shift in choice, a more motivational aspect. How can these aspects be represented in a computational model of mental fatigue? Therefore we have to determine how global parameters can interact with knowledge-specific parameters. In ACT-R two global parameters can be related to these aspects of mental fatigue. In the next two subsections these two parameters will be explained and the third section illustrates the influence of the values of these two parameters on the performance on a Sternberg memory-search task.

Mental Fatigue as a Problem Concerning Cognitive Control

As already mentioned in the introduction, West (1996) distinguishes three cognitive control functions: inhibition of interfering processes and stimuli, and two memory functions. A global parameter in ACT-R related to these functions is the source activation, which was described as a part of equation (2). Source activation spreads from the goal to related chunks, thereby creating more contrast between chunks which are relevant and irrelevant to the current goal. When source activation is low, the contrast between relevant and irrelevant chunks is low. As such, source activation has the same function as inhibition of interfering stimuli, which was described as one of the cognitive control functions possibly harmed by mental fatigue. When source activation is high, the probability of interference is low. When source activation is low, however, interfering stimuli can become problematic. It is also possible that due to low source-activation, the activation level of relevant chunks drops below the retrieval threshold, which means that relevant facts cannot be retrieved at all. Furthermore, there are already some indications that source activation is related to working memory. Lovett, Reder & Lebière (1997), for example, found that individual differences in working memory capacity can be simulated by changing the source activation. Therefore, it can be hypothesised that when people are fatigued, their source activation is lower.

Mental Fatigue as a Motivational Problem

Shingledecker & Holding (1974) and Hockey (1997) hypothesise that mental fatigue may also involve a shift in choice, more specifically, a shift toward strategies requiring less mental effort. This can be related to the motivation of the participants. The parameter closest to the concept motivation is the G parameter described before in equation (1), which represents the value of the goal. Literally, the G parameter represents how much time you are willing to invest in reaching the current goal. When the task does not involve time pressure, the value of the G parameter is partly determined by the motivation of the participant (Taatgen, 1997). So, it can be predicted that a highly motivated participant will favour strategies with a high probability of success, while participants with low motivation will favour strategies with less costs.

An Example: a Model of the Sternberg Memory-Search Task

The model described in this subsection is adapted from Anderson & Lebière (in preparation). The task the model performs is a modified version of the Sternberg memory-search task (Sternberg, 1969). In this task three letters are shown on a computer screen, which the participant has to keep in memory. These three letters are referred to as the memory set. The time the memory set is shown is long enough to read the letters, but not long enough to rehearse them. After that, an attention dot is shown, followed by a set of four letters, called the display set. The participant has to decide whether one of the letters from the display set was part of the memory set. The probability that this is the case is 50 percent. A new memory set is presented on each trial, which immediately starts after the participant has given a response, making the task self-paced.

The two strategies which can be used to perform the task are described in Anderson & Lebière (in preparation). The strategy that generally has the best speed-accuracy properties will here be referred to as *retrieve-and-check*. When the display set is shown, the participant focuses on the first letter in the set. He then retrieves the letter from the memory set with the highest activation. If this retrieved letter equals the attended letter in the display set the participant responds with a yes, else he moves on to the next letter in the display set. If there is a letter in the memory set corresponding with the attended letter, this letter will have the highest activation.

The main production rules for retrieve-and-check are given below. This strategy will produce fast responses, since the retrieve-trace production will always succeed.

Retrieve-trace
IF the goal is to check if item x is in the memory set
   and there is some item y in the memory set
THEN the target is item y

Retrieve-yes
IF the goal is to check if item x is in the memory set
   and the target is item x
THEN say-yes

Retrieve-no
IF the goal is to check if item x is in the memory set
   and target is not equal to item x
THEN move on to the next item of the display set

The second strategy focuses on accuracy, but is less efficient. It is called *specific-retrieval,* since the participant specifically has to retrieve the memory set item that matches the current display set item. This will result in a higher accuracy, since it is impossible to retrieve a wrong item from the memory set. Another consequence, however, is that the retrieve-trace production will fail most of the time. This results in a longer reaction time, since failing production rules use the time it takes to retrieve items whose activation equals the retrieval threshold. The main production rules for this strategy are given below.

Retrieve-trace
IF the goal is to check if item x is in the memory set
   and item x is in the memory set
THEN the target is item x

Retrieve-yes
IF the goal is to check if item x is in the memory set
   and the target is item x
THEN say-yes

Retrieve-no
IF the goal is to check if item x is in the memory set
THEN move on to the next item of the display set

The retrieve-no rule has a lower expected gain than retrieve-trace, so it will only fire when retrieve-trace fails.

Source activation, which was proposed as a global parameter concerning mental fatigue, effects the retrieve-trace rule, since that rule tries to retrieve an item from the memory set. In the retrieve-and-check strategy the source activation ensures the right item is retrieved. Lowering the source activation will increase the probability of retrieving the wrong item, thereby producing more errors. In the specific-retrieval strategy lowering the source activation hardly influences the number of errors that will be made. This can be seen in figure 3 which presents some simulated data from the model. The figure also shows that for the retrieve-and-check strategy reaction times become slower when source activation is lowered. The reason for this is that the activation of the items in the memory set is lower, because they receive less source activation (see equation 2). In ACT-R it takes more time to retrieve an item when its activation is low.

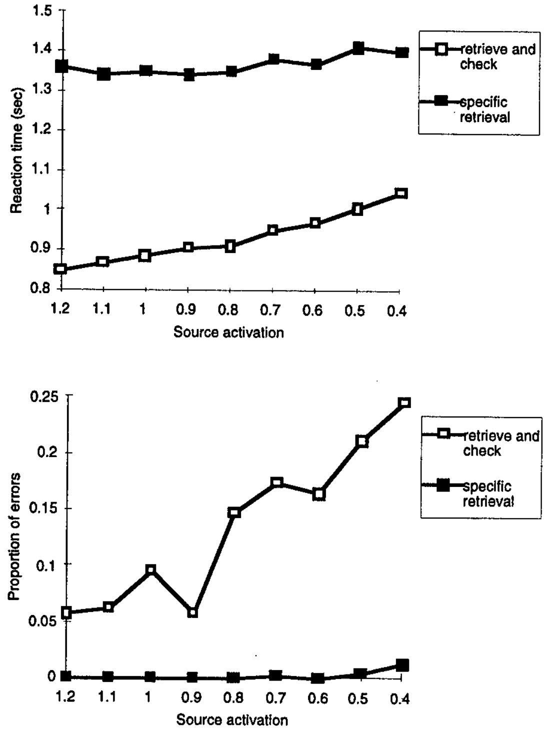


Figure 3. The changes in reaction times and proportion of errors for both strategies, as a result from lowering the source activation.

As already explained before, expected gain determines which strategy will be chosen in a particular situation. When people are fit, and thus have a high source activation, the expected gain of the retrieve-and-check strategy will be highest. However, according to figure 3, when source activation becomes lower, it can be predicted that at some point in time the expected gain of the specific-retrieval strategy will become the highest, and therefore a shift in strategy will be made. The exact timing of this strategy change is dependent on the motivation of the participant. Figure 4 illustrates the effect of motivation and source activation on the expected gain of the two strategies. The expected gain is calculated according to equation 1 using reaction time (from figure 3) as cost, and one minus the proportion of errors as probability of success. ACT-R's conflict resolution mechanism will choose the strategy with the highest expected gain. As can be seen from the figure, when the motivation of the participant is low (represented by a low value of the G parameter) and source activation is lowered, people still maintain the retrieve-and-check strategy, although this results in a great number of errors. However, when the motivation is higher and source activation is lowered, the participant will shift

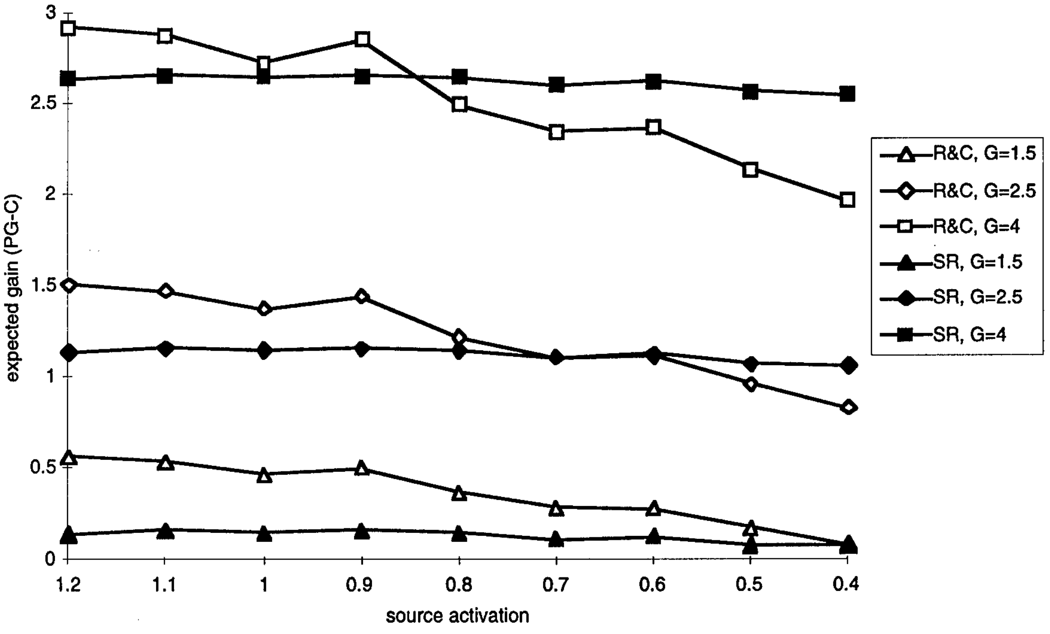


Figure 4. The expected gain of both strategies as a function of the source activation and the motivation (represented by the value of G) of the participant. R&C = retrieve-and-check, SR = specific retrieval.

towards the specific-retrieval strategy. Furthermore, the higher the motivation of the subject, the sooner this strategy shift will take place.

A shift in strategy, or strategic adjustment in Hockey's terms, is one change Hockey describes that can happen when people become mentally fatigued. The ACT-R model, however, can also predict such a change and show how this depends on the participant's motivation. Hockey's model describes that performance normally relies on routine regulation. When people become fatigued two situations can arise: either performance will decrease, or control will be shifted to a higher level (loop B in Hockey's model). What this shift in control involves is not completely clear from the model. The ACT-R model does show what a shift in control involves. When people become fatigued and routine-regulation is not adequate for task performance, the conflict resolution process in ACT-R will select a strategy that is less sensitive to fatigue. So, in this model, the change in cognitive control can be directly derived from the basic processes of the ACT-R theory.

Although an experiment to validate this model has not been done yet, some studies support the outcomes of the model. In two studies (Kerstholt, van Orden & Gaillard, 1994; van Orden, Gaillard & Langefeld, 1996) in which task instructions for the memory-search task focused on accuracy, mental fatigue manifested itself by increasing reaction times, which could indicate the use of the specific-retrieval strategy. In another study (Schellekens, Sijtsma & Vegter, in preparation) in which both accuracy and speed were emphasised, participants only had a fixed time to respond. In this experiment mental fatigue was accompanied by an increase in the number of errors. This decrease of accuracy can be explained by the fact that the time subjects had to respond was too short for the application of the specific-retrieval strategy, so participants had to stick to the retrieve-and-check strategy.

CONCLUSIONS AND RECOMMENDATIONS

As was shown in the previous section, the model provides detailed predictions of performance changes when people become mentally fatigued. Furthermore, the changes it predicts can be directly derived from the ACT-R theory, which allows for generalisation. Given an ACT-R model of a certain task, it is easy to predict the role of mental fatigue in task performance. It will be especially interesting to study the effects of manipulation of source activation on models of more complex tasks that allow participants more strategic freedom, since several authors have argued that these tasks are most influenced when people become fatigued (e.g., Bartlett, 1943; Meijman, 1997). The model also predicts that some tasks will hardly be sensitive to mental fatigue, for example, if the strategy used does not rely on source activation. However, the model has not been validated yet, so future experiments have to be carried out to support it.

REFERENCES

Anderson, J.R. (1983). *The architecture of cognition.* Cambridge, M.A.: Harvard university press.

Anderson, J.R. (1993). *Rules of the mind.* Hillsdale, NJ: Erlbaum.

Anderson, J.R. & Lebière, C. (in preparation). *The atomic components of thought.* Hillsdale, NJ: Erlbaum.

Bartlett, F.R.S. (1943). Fatigue following highly skilled work. *Proceed. Royal Society, 131*, 247-257.

Gopher, D. (1986). In defence of resources: on structures, energies, pools and the allocation of attention. In G.R. Hockey, A.W.K. Gaillard & M.G.H. Coles (eds.). *Energetics and human information processing.* Dordrecht: Martinus Nijhoff.

Hockey, G.R.J. (1997). Compensatory control in the regulation of human performance under stress and high workload: A cognitive-energetical framework. *Biological psychology, 45*, 73-93.

Holding, D.H. (1983). Fatigue. In G.R.J. Hockey (ed.) *Stress and fatigue in human performance.* Chichester: John Wiley.

Kahneman, D. (1973). *Attention and effort.* Englewood Cliffs, NJ: Prentice-Hall.

Kerstholt, J.H., van Orden, C.Y.D. & Gaillard, A.W.K. (1994). Effecten van vermoeidheid als functie van soort taak en sociale omgeving. TNO-rapport TM 1994 A-9, TNO Technische Menskunde, Postbus 23, 3769 ZG Soesterberg, The Netherlands.

Lovett, M.C., Reder, L.M. & Lebière, C. (1997). *Proceedings of the 19th Annual Conference of the Cognitive Science Society.* Hillsdale, NJ: Erlbaum.

Meijman, T.F. (1997). Mental fatigue and the efficiency of information processing in relation to work times. *International Journal of Industrial Ergonomics, 20*, 31-38.

Orden, C.Y.D., van, Gaillard, A.W.K. & Langefeld, J.J. (1996). Effecten van vermoeidheid en sociale omgeving op prestatie: de rol van feedback. TNO-rapport TM-96-A035, TNO Technische Menskunde, Postbus 23, 3769 ZG Soesterberg, The Netherlands.

Schellekens, J.M.H., Sijtsma, G.J. & Vegter, E. (in preparation). Immediate and delayed aftereffects of mentally demanding work hours. University of Groningen, Groningen, The Netherlands.

Shiffrin, R.M. & Schneider, W. (1977). Controlled and automatic human information processing: II. Perceptual learning, automatic attending, and a general theory. *Psychological Review, 84*, 127-190.

Shingledecker, C.A. & Holding, D.H. (1974). Risk and effort measures of fatique. *Journal of motor behavior, 1*, 17-25.

Sternberg, S. (1969). Memory scanning: Mental processes revealed by reaction time experiments. *American Scientist, 57*, 421-457.

Taatgen, N.A. (1997). A rational analysis of alternating search and reflection strategies in problem solving. *Proceedings of the 19th Annual Conference of the Cognitive Science Society.* Hillsdale, NJ: Erlbaum.

West, R.L. (1996). An application of prefrontal cortex function theory to cognitive aging. *Psychological bulletin, 120*, 272-292.

Wickens, C.D. (1984). Processing resources in attention. In R. Parasuraman & D.R. Davies (eds.) *Varieties of attention.* New York, NY: John Wiley.

Architectures and Tools for Human-Like Agents

Aaron Sloman and **Brian Logan**
School of Computer Science
The University of Birmingham
Birmingham, B15 2TT, UK
+44 121 414 {4775 (Sloman) 3712 (Logan)}
{A.Sloman, B.S.Logan}@cs.bham.ac.uk

ABSTRACT

This paper discusses agent architectures which are describable in terms of the "higher level" mental concepts applicable to human beings, e.g. "believes", "desires", "intends" and "feels". We conjecture that such concepts are grounded in a type of information processing architecture, and not simply in observable behaviour nor in Newell's knowledge-level concepts, nor Dennett's "intentional stance." A strategy for conceptual exploration of architectures in design-space and niche-space is outlined, including an analysis of design trade-offs. The SIM_AGENT toolkit, developed to support such exploration, including hybrid architectures, is described briefly.

Keywords

Architecture, hybrid, mind, emotion, evolution, toolkit.

MENTALISTIC DESCRIPTIONS

The usual motivation for studying architectures is to explain or replicate performance. Another, less common reason, is to account for concepts. This paper discusses "high level" architectures which can provide a systematic non-behavioural conceptual framework for mentality (including emotional states). This provides a new kind of semantics for mentalistic descriptions. We illustrate this using multi-layered architectures based in part on evolutionary considerations. We show briefly how different layers support different sorts of emotion concepts. This complements work by McCarthy(1979, 1995) on descriptive and notational requirements for intelligent robots with self-consciousness.

We provide pointers to an uncommitted software toolkit that supports exploration of hybrid architectures of various sorts, and we illustrate some of the architectural complexity it needs to support.

WHY USE MENTALISTIC LANGUAGE?

We shall need mentalistic descriptions for artificial agents for the same reasons as we need them for biological agents, e.g. (a) because such descriptions will (in some cases) be found irresistible and (b) because no other vocabulary will be as useful for describing, explaining, predicting capabilities and behaviour. ((b) provides part of the explanation for (a).) So, instead of the self-defeating strategy of trying to avoid mentalistic language, we need a disciplined approach to its use, basic mentalistic concepts on information-level architectural concepts.

The "Information level" design stance

Dennett (1978) recommends the "intentional stance" in describing sophisticated robots, as well as human beings. That restricts mentalistic language to descriptions of whole agents, and presupposes that the agents are largely rational. Similarly, Newell (1982) recommends the use of the "knowledge level", which also presupposes rationality. By contrast, we claim that mentality is primarily concerned with an "information level" architecture, close to the requirements specified by software engineers. This extends Dennett's "design stance" by using a level of description between physical levels (including physical design levels) and "holistic" intentional descriptions.

"Information level" design descriptions allow us to refer to various *internal* semantically rich short term and long term information structures and processes. This includes short term sensory buffers, longer term stored associations, generalisations about the environment and the agent, stored information about the local environment, currently active motives, motive generators that can produce motives under various conditions, mechanisms and rules for detecting and resolving conflicts, learnt automatic responses, mechanisms for constructing new plans, previously constructed plans or plan schemata, high level control states which can modulate the behaviour of other mechanisms, and many more.

Some mentalistic concepts refer to the information processing and control functions of the architecture. These functions include having and using information *about* things. E.g. an operating system has and uses information *about* the processes it is running. Here semantic content is present without full-blown intentionality or rationality. Restricting semantic notions to global states of a rational agent, or banning them altogether from explanatory theories, would be as crippling in the study of intelligent agents as it would be in the engineering design of complex control systems. (However, not all semantic states can be fully characterised in terms of *internal* functions, for instance those that refer to *particular* external objects, such as Buckingham Palace, a point beyond the scope of this paper.)

Many of the mechanisms in such an architecture are neither rational nor irrational: even though they acquire information, evaluate it, use it, store it, etc. (Sloman 1994*b*). They are neither rational nor irrational because they are *automatic*. Even a deliberative architecture at some level needs reactive mechanisms to drive the processing. If everything had to be based on prior goals and justifications nothing would ever happen.

ARCHITECTURAL ANALYSIS

Different architectures can correspond to different views of a system, e.g. a physical architecture, composed of the major physical parts, a physiological architecture, corresponding to the major functional roles of physical parts, and an information processing architecture composed of mechanisms involved in acquiring, transforming, storing, transmitting, and using information.

There need not be a one to one correspondence between components in different views. A physical component may be shared between several physiological functions: e.g. the circulatory system is involved in distribution of energy, waste disposal, temperature control, and information transfer.

There is a huge space of possible designs. We make no presumption that information processing mechanisms must all be computational (whatever that means). Nor is there a commitment regarding *forms* used to encode or express information. They may include logical databases, procedures encoding practical know-how, image structures, neural nets or even direct physical representations, as in thermostats and speed governors.

Biological plausibility requires evolvability as well as consistency with experimental data and brain physiology. The capabilities and neural structures of different sorts of animals (e.g. insects, rodents, apes, humans) suggest that different types of architectures evolved at different times, with newer architectures building new sorts of functionality on older ones. We suggest that human mental states and processes depend on interactions between old and new layers in a biologically plausible control architecture producing various kinds of internal and external behaviour, including "internal" processes such as motive generation, attention switching, global redirection in emergencies, problem solving, information storage, skill acquisition, self-evaluation and even modification of the architecture.

Besides the multi-layered central information processing architecture there are sensors and effectors of various kinds. These involve more than just transduction of energy or information into or out of the system. We suggest that both have evolved multiple layers interacting with the different layers in the central system as in Figure 1. Such an architecture can generate a huge variety of concepts relevant to describing its states and processes. It also supports a wide variety of types of learning, yet to be analysed.

Indeterminacy of architecture

Often boundaries between sub-mechanisms and levels of description are unclear, including the boundary between the control architecture and mere physiological infrastructure. In brains, chemical processes provide energy and other resources, along with damage repair and resistance to infections. However, effects of drugs, diseases and genetic defects involving brain chemicals suggest that chemistry forms more than a physiological infrastructure: chemically controlled mood changes may be an important part of an organism's intelligent reaction to changing circumstances, and alcohol can change "no" into "yes"! But we don't know how far chemical reactions play a direct role in information processing or high level control,

In both perception and action the "hardware/software" boundary is blurred. E.g. visual attention can be switched with or without redirection of gaze, and fine-grained manipulation can be shared between software and hardware, e.g. in compliant wrists, which reduce the control problem in pushing a close fitting cylinder into a hole. Simon (1969) pointed out long ago that there can be information sharing between internal and external structures.

It is too early for clear definitions of the boundaries of architectures or their components. However, important ideas are beginning to emerge including contrasts between:
(a) reactive *vs* deliberative functions,
(b) symbolic *vs* neural mechanisms,
(c) logical *vs* other sorts of information manipulation,
(d) continuous *vs* discrete control,
(e) using continuously available environmental information *vs* using information stored in memory,
(f) hierarchical *vs* distributed control,
(g) serial *vs* concurrent processing,
(h) synchronised *vs* asynchronous processing,
(i) genetically determined capabilities, those produced by adaptive mechanisms within individuals, and those absorbed from a culture (e.g. learnt poems and equations).

Instead of viewing these contrasts as specifying *rival* options, we should allow combinations of these alternatives to have roles in multifunctional architectures. Work on hybrid mechanisms (e.g. combinations of neural and symbolic systems) is now commonplace, but in order to explore agents rivalling human or even chimpanzee sophistication we need to understand far more complex combinations of subsystems, including complex sub-architectures *within* perceptual and motor control mechanisms, and a deep integration of cognitive and affective functions and mechanisms (Wright, Sloman & Beaudoin 1996, Sloman 1998(forthcoming)). However, there is no unique "correct" architecture: different designs have different trade-offs, as biological evolution shows. We need to understand the trade-offs and possible trajectories. This includes finding good concepts for describing systems with different designs.

ARCHITECTURES AND EMERGENT CONCEPTS

A deep conceptual framework takes account of the range of possible states and processes supported in an architecture, generating a system of high-level descriptive concepts for describing an organism, software agent, or robot, just as a knowledge of molecular architecture provides a basis for labelling chemical compounds and describing chemical processes.

A control architecture can support a collection of states and processes, often indefinitely large. Concepts derived in this way from the architecture are "deep concepts". "Shallow" concepts, based entirely on observed behavioural patterns bearing no relationship to the architecture, are likely to have reduced predictive and explanatory power, like concepts of physical matter based on visible properties rather than atomic and molecular structure.

Not all states require specific mechanisms in the architecture. A computing system that is "overloaded" does not have an "overloading" mechanism, since overloading results from interaction of many different mechanisms whose functions is not to produce overload. Similarly many mental states, e.g. some debilitating emotions, may *emerge* from interactions within an architecture, rather than from an emotion module.

If there are several coexisting, interacting sub-architectures (e.g. reactive and deliberative sub-architectures) then higher order concepts are needed to describe the variety of possible relationships between them. For instance, states in one subsystem can modulate processes in others. Such relationships can change over time: sometimes one part is dominant and sometimes the other. Moreover, when training increases fluency in a cognitive skill this may shift responsibility for a task from a general purpose module to a dedicated module.

Familiar prescientific concepts, e.g. "emotion", can be ambiguous if they sometimes refer to processes in a component of the architecture (e.g. being startled, or terrified by a fast approaching menace, may result from a specific module, perhaps part of the limbic system) and sometimes to emergent interactions between subsystems (e.g. guilt and self-reproach).

Unlike emotions which we share with rats, e.g. being startled, which use this old global alarm system, many human emotions involve a partial loss of control of thought processes, (e.g. extreme grief, ecstasy or hysteria). This presupposes the possibility of being in control. That, in turn, depends on the existence of an architecture that supports certain kinds of self monitoring, self evaluation, and self modulation. Being careful or careless requires an architecture able to control which checks are made during planning, deciding and acting.

Which animal architectures can support control of thought processes is not clear. Systems lacking such underpinnings may not be usefully describable as "restrained", "resisting temptation", etc. Can a rat sometimes control and sometimes lose control of its thought processes? Can a rat be careless in its deliberations? Over-simple architectures in software agents will also make such concepts inappropriate to them.

EVOLUTION AND MODULARITY

Our discussion has presupposed that architectures are to some extent intelligible. Will naturally evolved systems be modular and intelligible? In principle, any required finite behaviour could be produced by a genetically determined, unstructured, non-modular architecture, including myriad shallow condition-action rules with very specific conditions and actions providing flexibility. However, as the diversity of contexts grows and the need to cope with unexpected situations, including interactions with other other agents, increases, memory requirements for such a system can grow explosively, and it becomes more difficult find a design which anticipates all the conditions and actions in advance. Thus the time required to evolve all the shallow capabilities is far greater and the required diversity of evolutionary contexts far greater than for a system with planning abilities.

A shallow non-modular system would not only be hard to design, describe and explain: it would be hard to control or modify, whether controlled from outside or controlling itself, whether modified by a designer, or modified by evolution. (Contrast the use of bit-strings in genetic algorithms with the use of trees in genetic programming.)

All this suggests that for complex organisms there would be pressure towards more modular architectures with generic mechanisms that can be combined by a planner to handle new situations, and adaptive architectures that can change themselves to improve performance. Both the normal evolutionary pressures for modularity and reuse, and the need for economy in high level self-control mechanisms could have increased the pressure towards evolution of modular control architectures, in some organisms. So the existence of self-monitoring, self-evaluation and self-control processes could influence the further evolution of control architectures. Apparently insects found a different solution.

It may eventually be possible to investigate this issue in simulated evolution.

THE EMERGENCE OF "QUALIA"

If a system has the ability to monitor its own states and processes, a new variety of descriptions becomes applicable, labelling new forms of self control, including its own discovery of concepts for self-description. The objects of such self-monitoring processes may be virtual machine states as well as internal physical or physiological states.

Many of the spatial, temporal and causal categories used in perceiving the environment have evolved to support biological functions of organisms in those environments, even though precise details can vary widely between species and between individuals in a species. Likewise, it is possible that the basic and most general mentalistic categories that humans use in describing and thinking about themselves and other agents are not reinvented by different individuals (or cultures) but generated by evolutionary processes driving development of self-monitoring capabilities.

Phenomena described by philosophers as "qualia" may be explained in terms of high level control mechanisms with the ability to switch attention from things in the environment to *internal* states and processes, including intermediate sensory datastructures in layered perceptual systems. These introspective mechanisms may explain a child's ability to describe the location and quality of its pain to its mother, or an artist's ability to depict how things look (as opposed to how they are). Software agents able to inform us (or other artificial agents) about their own internal states and processes may need similar architectural underpinnings for qualia.

From this standpoint, the evolution of qualia would not be a single event, but would involve a number of steps as more kinds of internal states and processes became accessible to more and more kinds of self-monitoring processes with different functions, e.g. requesting help from others or discovering useful generalisations about oneself. Such step-wise development may also occur within an individual.

HOW TO MAKE PROGRESS

There are several ways in which we might try to explore the relationship between architecture and mentality. One approach is to push the approach based on "shallow" behaviour-based concepts as far as possible, and analyse where it breaks down, or where patching it is very difficult (e.g. dealing with new unexpected combinations of conditions where applicable rules conflict, or where no rule applies).

Another approach is to attempt a theoretical analysis of the types of situations that will make development increasingly difficult and to produce increasingly general architectures to cope with the difficulties, using any ideas that work, and then conducting experiments to find out where they break down. This approach need not be constrained by theories of how human minds work: there may be alternative architectures capable of producing extremely useful or even "believable" performances. Initially the constraints on this type of theorising will be very ill-defined because of paucity of relevant knowledge and the shallowness of current theories. However, it is likely that as the work progresses more and more constraints can come from advances in other fields, and more and more tests can be generated to help us choose between alternative hypotheses. (Compare the ancient Greek atomic theory with modern atomic theory.)

Yet another approach is to use whatever direct or indirect evidence is available from brain science, experimental psychology, forms of mental disorder, patterns of development in infancy and decay in old age, evolution, folklore, introspection, common observation, or conceptual analysis of everyday mental concepts. Plausible architectures based on such evidence can then be tested by running experimental implementations, or by analysing their consequences and performing empirical research.

Our work is based on the second and third approaches. The architectural ideas in this paper come from a wide range of sources.

ARCHITECTURAL LAYERS

Part of the task is to find increasingly accurate and explicit theories of the types of architecture to be found in various sorts of human minds (and others) to be used as frameworks for generating families of descriptive concepts applicable to different sorts of humans (including infants and people with various kinds of brain damage) and different sorts of animals and artificial agents.

We conjecture that human-like agents with powers of self-control need a type of architecture with at least three distinct classes of mechanisms which evolved at different times (Sloman 1998(forthcoming)):

(1) Very old reactive mechanisms, found in various forms in all animals, including insects — this includes "routine" reactive mechanisms and "global alarm" mechanisms (the limbic system).

(2) More recently evolved deliberative mechanisms, found in varying degrees of sophistication in some other animals (e.g. cats, monkeys);

(3) An even more recent meta-management (reflective) layer providing self-monitoring self-evaluation, and self-control, using in part deliberative mechanisms of type (2), and perhaps found only in humans and other primates (in simpler forms).

Such an architecture is shown schematically (without alarms) in Figure 1 and each of the layers is described in more detail below. Note that the layers occur in perceptual and motor subsystems as well as centrally.

This is one among many possible designs. Some animals or artefacts may have only one or two layers, and different kinds of reactive, deliberative and meta-management mechanisms are possible.

We are not claiming that these mechanisms are alike in all humans. Deliberative capabilities seem very primitive in new born infants, and the third layer may be non-existent at birth. Moreover a culture can influence development of these layers, as can effects of brain damage, disease or aging. Some architectures may be possible for synthetic agents that are never found in organisms (e.g. solely deliberative architectures, or hybrid systems without global alarms).

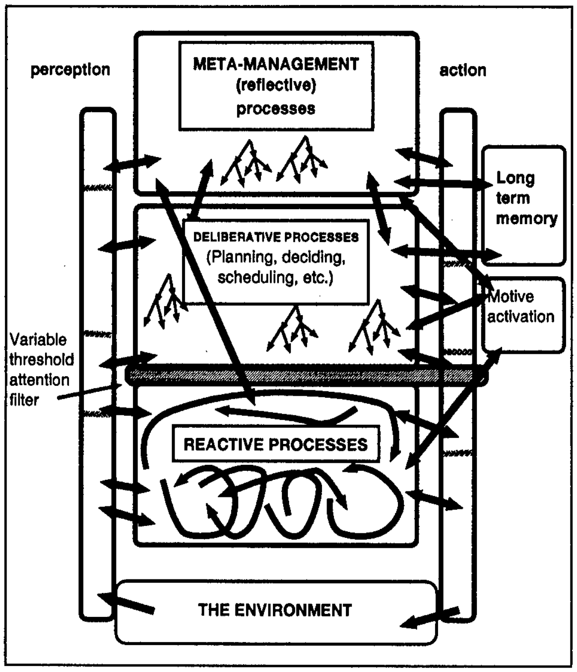


Figure 1: **A three layered agent Architecture**
(Note: global 'alarm' mechanisms not shown.)

Categories and strategies in all layers may be influenced by physical and social environments. A meta-management layer may use both categories and values absorbed from a culture as well as some genetically determined categories and strategies. For instance, certain motives for acting promote negative self-assessment and guilt in some cultures and not in others.

Within an individual, it is also possible for different modes of meta-management to take control in different contexts, e.g. in a family context, in a football game, and in the office. Individual variations might lead, at one extreme to multiple-personality disorder, and at another extreme to excessively rigid personalities.

Concurrent mechanisms

The layers are not assumed to form a rigidly hierarchical control architecture. Rather the three layers operate concurrently, with mutual influences. The reactive mechanisms will perform routine tasks using genetically determined or previously learnt strategies. When they cannot cope, deliberative mechanisms may be invoked, by the explicit generation of goals to be achieved. This can trigger various kinds of deliberative processes including considering whether to adopt the goal, evaluating its importance or urgency, working out how to achieve it, comparing it with other goals, deciding when to achieve it, deciding whether this requires reconsideration of other goals and plans, etc. (See chapter 6 of Sloman (1978).)

At other times the deliberative mechanisms may either attend to long term unfinished business or run in a "free-wheeling" mode, nudged by reactive processes which normally have low priority, including attention-diverting mechanisms in the perceptual subsystems. To allow direct communication with "higher" cognitive functions, perceptual systems may also have layered architectures in which different levels of processing occur in parallel, with a mixture of top-down and bottom-up processing. (Compare seeing a face as a face and as happy.)

If the internal layers operate concurrently, fed in part by sensory mechanisms which are also layered, they may also benefit from a layered architecture in motor systems. For example, reactive mechanisms may directly control some external behaviour, such as running, while the other mechanisms are capable of modulating that behaviour (e.g. changing the speed or style of running, or in extreme cases turning running into dancing). Likewise proprioceptive feedback of different sorts may go to different layers.

Where there is a global alarm system, there may be variations as regards which components provide its inputs and which can be modified by it. In humans connections to and from the limbic system seem to exist everywhere (Goleman 1996).

We now describe in a little more detail the differences between the layers (Figure 1) before discussing their implications for emotions. (The figure is much simplified, to reduce clutter).

Reactive agents

It is possible for an agent to have a purely reactive architecture, where:

- Mechanisms and space are permanently dedicated to specific functions, and can run concurrently, more or less independently, with consequent speed benefits. Some may be digital, some continuous.
- Conflicts may be handled by vector addition, voting, or winner-takes-all nets.
- Some learning is possible: e.g. tunable control loops, change of weights by reinforcement learning. Such learning merely alters links between pre-existing structures and behaviours.
- There is no explicit construction of new plans or structural descriptions or other complex internal objects, and therefore no explicit evaluation of alternative structures.
- Concurrent processing at different abstraction levels can encourage the evolution of different levels of processing in sensory and motor subsystems.
- Some of the reactions to external or internal conditions may be internal, e.g. various kinds of internal feedback control loops.
- If "routine" reactions are too slow a fast "global alarm" system taking control in emergencies may be useful.

As explained above, if all the main possible behaviours need to be built in by evolutionary adaptation or direct programming the space requirements may explode as combinations increase. Likewise the time required to evolve all relevant combinations. A partial solution is to provide "chaining" mechanisms so that simpler behaviours can be re-used in different longer sequences. Simple sub-goaling may achieve this, changing internal conditions that launch behaviours. This may be a precursor to deliberative mechanisms.

It appears that insects have purely reactive architectures, and cannot reflect on possible future actions. Yet the reactive behaviours can produce and maintain amazing construction, e.g. termites' "cathedrals".

There is no form of externally observable behaviour that cannot, in principle, be implemented in a purely reactive system, without any deliberative capabilities, though it seems that in some organisms the evolutionary pressures mentioned above have led towards a different solution — which may coexist with the old one.

Combining reactive and deliberative layers

The ability to construct new complex behaviours as required reduces the amount of genetic information that needs to be transmitted as well as the storage requirements for each individual. It also reduces the number of generations of evolution required to reach a certain range of competence. In a deliberative mechanism:

- Evaluating and comparing options for novel combinations before selecting them requires a new ability to build internal descriptions of internal structures. It also needs a long term associative memory.
- Using re-usable storage space for new plans and other temporary structures, and use of a single associative memory (even if based on neural nets), makes processes inherently serial.
- New behaviours developed by the deliberative system can be transferred to the reactive layer (e.g. learning new fluent skills).
- Sensory and action mechanisms may develop new, more abstract, processing layers, which communicate directly with deliberative mechanisms. This could explain high level sensory experiences (e.g. seeing a face as happy).
- Even if neural nets are used, operation may be resource-limited because learning from consequences becomes explosive if too many things are done in parallel. Limiting concurrent processes may also simplify integrated control.
- Deliberative resource limits may mean that a fast-changing environment can cause too many interrupts and re-directions. Filtering new interrupts via dynamically varying thresholds (see Figure 1) helps but does not solve all problems.
- A global alarm system may include inputs from and outputs to deliberative layers.

The need for self-monitoring (meta-management)

Deliberative mechanisms may be implemented in specialised reactive mechanisms which react to internal structures, and can interpret explicit rules and plans.

However, evolutionarily determined deliberative strategies for planning, problem solving, decision making, evaluating options, can be too rigid. Internal monitoring mechanisms may help to overcome this e.g. by recording deliberative processes and noticing which planning strategies or attention switching strategies work well in which conditions. This could include detecting when one goal is about to interfere with other goals, or noticing that a problem solving process is "stuck", e.g. in a loop, or noticing that a solution to one problem helps with another.

Internal monitoring combined with learning mechanisms may allow discovery of new ways of categorising internal states and processes and better ways of organising deliberation. Meta-management and deliberative mechanisms permit cultural influences via the absorption of new concepts and rules for self-categorisation, evaluation and control.

Attending to intermediate perceptual structures can also allow more effective communication about external objects, e.g. by using viewpoint-centred appearances to help direct attention, or using drawings and paintings to communicate about how things look.

The meta-management layer may share mechanisms with the other two, including the global alarm mechanism (limbic system?) but also needs new mechanisms that can access states and processes in various parts of the whole system, categorise what is going on internally, evaluate it, and in some cases modify it. This can help with proper management of limited deliberative resources.

ARCHITECTURAL LAYERS & EMOTION CONCEPTS

We conjecture that different layers account for different sorts of mental states and processes, including emotional states. Disagreements about the nature of emotions can arise from failure to see how different concepts of emotionality depend on different architectural features, not all shared by all the animals studied.

(1) The old reactive layer, with the global alarm system, produces rapid automatically stimulated emotional states found in many animals (being startled, terrified, sexually excited).

(2) A deliberative layer, in which plans can be created and executed, supports cognitively rich emotional states linked to current desires plans and beliefs (like being anxious, apprehensive, relieved, pleasantly surprised).

(3) Characteristically human emotional states (e.g. humiliation, guilt, infatuation, excited anticipation) can involve reduced ability to focus attention on important tasks because of reactive processes (including alarm processes) interrupting and diverting deliberative mechanisms, sometimes conflicting with meta-management decisions (Wright et al. 1996).

The second class of states depends on abilities possessed by fewer animals than those that have reactive capabilities. The architectural underpinnings for the third class are relatively rare: perhaps only a few primates have them.

Many theories of emotion postulate a system that operates in parallel with normal function and can react to abnormal occurrences by generating some kind of interrupt, like the global alarm mechanism. Consider an insect-like organism with a purely reactive architecture, which processes sensory input and engages in a variety of routine tasks (hunting, feeding, nest building, mating, etc.). It may be useful to detect certain patterns which imply an *urgent* need to react to danger or opportunity by freezing, or fleeing, or attacking, or protecting young, or increasing general alertness. Aspects of the limbic system in vertebrate brains seem to have this sort of function (Goleman 1996).

In architectures combining reactive and deliberative layers, the alarm mechanism can be extended to cause sudden changes also in *internal* behaviour, such as aborting planning or plan execution, switching attention to a new task, generating high priority goals (e.g. to escape, or to check source of a noise). Likewise processing patterns in the deliberative layer may be detected and fed into the alarm system, so that noticing a risk in a planned action can trigger an alarm.

Where a meta-management layer exists, data from it could also feed into the alarm system, and it too could be affected by global alarm signals. One meta-management function could involve learning which alarm signals to ignore or suppress. Another would extend the alarm system to react to new patterns, both internal and external. Another would be development of more effective and more focused (less global) high speed reactions, e.g. replacing a general startle reaction with the reactions of a highly trained tennis player.

This, admittedly still sketchy, architecture, explains how much argumentation about emotions is at cross-purposes, because people unwittingly refer to different sorts of mechanisms which are not mutually exclusive. An architecture-based set of concepts can be made far less ambiguous.

Familiar categories for describing mental states and processes (e.g. believes, desires, perceives, attends, decides, feels, etc.) may not survive unchanged as our knowledge of the underlying architecture deepens, just as our categories of kinds of physical stuff were refined after the development of a new theory of the architecture of matter. Researchers need to be sensitive to the relationships between pre-theoretical and architecture-based concepts as illustrated in (Wright et al. 1996).

THE SIM_AGENT TOOLKIT

We still have much to learn about different agent architectures. The properties of complex systems cannot all be determined by logical and mathematical analysis: there is a need for a great deal more exploration of various types of architectures, both in physical robots and in simulated systems.

Many robot laboratories are doing the former. We work on simulated systems so that we can focus on the issues that are of most interest to us, involving the kind of architecture sketched above including alarm systems, leaving details of sensory devices and motors till later. When simulations are well designed they can sometimes provide cheaper and faster forms of experimentation, though care is always necessary in extrapolating from simulations.

Many toolkits exist to support such exploration, usually based on a particular architecture or class of architectures (e.g. neural net architectures, or SOAR, or PRS). We wished to investigate diverse and increasingly complex architectures, including coexisting reactive and deliberative sub-architectures, along with self-monitoring and self-modifying capabilities, and including layered perceptual and action subsystems. We also wished to explore varying resource-limits imposed on different components of the architecture, so that, for example, we could compare the effects of speeding up or slowing down planning mechanisms relative to the remaining components of an architecture (e.g. in order to investigate various deliberation management strategies, such as "anytime" planning).

To support this exploration we designed and implemented (in the language Pop-11 (Sloman 1996)) the SIM_AGENT toolkit. It is being used at Birmingham for teaching and research, including research on evolutionary experiments, and also at DERA Malvern for designing simulated agents that could be used in training software. An early version of the toolkit developed jointly with Riccardo Poli, was described at ATAL95 (Sloman & Poli 1996). Since then development has continued in response to comments and suggestions from users (Baxter, Hepplewhite, Logan & Sloman. 1998).

The toolkit supports a collection of interacting agents and inanimate objects, where each agent has an internal architecture involving different sorts of coexisting interacting components, including deliberative and reactive components. Not all agents need have the same architecture.

The key idea is that each component within an agent is connected to other components in that agent via a forward-chaining condition-action rulesystem. Each agent's rulesystem is divided into a collection of different rulesets, where each ruleset is concerned with a specific function, e.g. analysing a type of sensory data, interpreting linguistic messages, creating, checking or executing plans, generating motives, etc. Rulesets can be concurrently active, and may be dynamically switched on and off. They may be assigned different resource limits.

Conditions and actions of rules within an agent can refer to databases in that agent. Thus one form of communication between sub-mechanisms is through the databases in the agent. It is possible for an agent to have some global databases accessed by all components of an agent and others which are used only by specific sub-groups. One agent cannot normally inspect another's databases.

An architecture for an agent class is defined by specifying a collection of rulesets and other mechanisms, along with the types of databases, sensor methods, action methods, communication methods and possibly tracing and debugging methods. It is hoped that users will develop re-usable libraries defining different mechanisms and architectures.

The rulesets are implemented in Poprulebase, a flexible and extendable forward-chaining rule-interpreter. Rulesets can be turned on and off dynamically, modelling one aspect of attention shift, and new ones added, modelling some forms of cognitive development. Although the main conditions and actions use patterns matching database components, some conditions and some actions can invoke sub-mechanisms directly implemented in Pop-11, e.g. low level vision or motor-control mechanisms. Other Poplog languages (e.g. Prolog) or external languages (e.g. C, Fortran) can also be invoked in conditions and actions. For example, a rule condition could in principle interrogate physical sensors and a rule action could send signals to motors. Sockets can run sub-systems on other machines, and unix pipes can communicate with processes on the same machine.

To illustrate the power, a Pop-11 rule action can run the rule interpreter recursively on a specialised rule system.

The rule-based formalism is easily extendable, allowing different sorts of condition-action rules to be defined. For example, one of the extensions designed by Riccardo Poli allows a set of conditions matched against a database to provide a set of input values for a neural net, whose output is a boolean vector which can be used to select a subset of actions to be run. A recent extension was a new class of ADD and DELETE actions for automatically maintaining sets of dependency information between database items, so that if an item is deleted then everything recorded as directly or indirectly depending on it, is also deleted. A Pop-11 condition can be used to perform backward chaining if desired.

The interpreter can be run with various control strategies, including the following options for each active ruleset on each cycle: (a) all runnable rules (those with all conditions satisfied) are run, (b) only the first runnable rule found is run, (c) the set of runnable rule instances is sorted and pruned (using a user-defined procedure) before the actions are run.

When the rule interpreter is applied to a ruleset, it can be allowed to run to completion (e.g. until no more rules have all conditions satisfied, or a "STOP" action is executed.) Alternatively it can be run with a cycle limit N, specifying that it should be suspended after N cycles even if there are still rules with satisfied conditions. Another possibility is to set a timer and halt it after a fixed time interval. Either of these mechanisms can be used to impose resource limits on one ruleset relative to others, within an agent.

The design of the toolkit supports multi-agent scenarios, using a time-sliced scheduler which in each time slice allows each agent to run its sensory methods, its internal rulesets, and, in a second pass at the end of the time slice, its *external* action methods.

The object oriented design uses Pop-11's Objectclass system, which supports multiple inheritance and generic functions. This makes it easy for users to extend the ontology by defining new sub-classes, with their own sensing, acting and internal processing methods, without any editing of the core toolkit code. A default class provides a default set of methods, including the sim_run_agent method used to run each the agent's rulesets, along with various tracing methods.

The object oriented approach allows a Pop-11 graphical library to be connected to the toolkit by re-defining tracing and other methods (e.g. move methods) to invoke graphical procedures. The graphical facilities support not only displays of agent actions but also asynchronous user intervention: e.g. using the mouse to move objects in an agent's environment, or turning tracing and profiling mechanisms on or off while the toolkit is running.

Scenarios implemented so far using the toolkit include a simulated robot using a hybrid modular architecture to propel a boat to follow the walls of an irregular room, evolution of a primitive language for cooperation between a blind and an immobile agent, a user controlled sheepdog and sheep to be penned, two purely reactive "teams" of agents able to move past each other and static obstacles to get to their target locations, a simulated nursemaid looking after troublesome infants while performing a construction task, a distributed minder (Davis 1996), one agent tracking another subject to path constraints in 3-D undulating terrain, and, at DERA Malvern, simulated tank commanders and tank drivers engaging in battle scenarios (Baxter 1996). We expect to continue developing the toolkit and building increasingly sophisticated simulations, moving towards the architecture depicted in Figure 1 and subsequently extended in various ways.

In particular we have plans for improving the self modifying and self monitoring capabilities by replacing the rulesystem, currently a list of rulesets and rulefamilies, with database entries. Thus rule actions can then change the processing architecture.

The toolkit is applicable to a wide range of agent development tasks, including simplified software agents which require only a small subset of beliefs, goals, plans, decisions, reactions to unexpected situations, etc. These might be web search agents, or "believable" entertainment agents whose observed behaviour invites mentalistic description whether or not the descriptions are justified by internal mechanisms, states and processes, e.g. the OZ project at CMU (Bates, Loyall & Reilly 1991). The toolkit could also be used to implement teaching and demonstration libraries, e.g. for students in psychology or the helping professions, where students can manipulate the architectures of simplified human-like agents, to gain a deeper understanding of the multiple ways in which things can go wrong.

CONCLUSION

Like software engineers, and unlike Dennett and Newell, we assume semantically competent sub-systems, but not rationality. Using this information-level design stance, we have sketched a framework accommodating multi-disciplinary investigation of many types of architecture of varying degrees of sophistication, with varying mixtures of information-processing capability, based on AI, Alife, Biology, Neuroscience, Psychology, Psychiatry, Anthropology, Linguistics and Philosophy. This framework can extend our understanding of both natural and artificial agents. Above all it generates systems of concepts for characterising various types of mentality. Information-based control architectures provide a new framework for analysing, justifying and extending familiar mentalistic concepts.

There is no uniquely "right" architecture. Types of architectures that are relevant, and dimensions of possible variation, are not yet well understood. More exploration and analysis is required, replacing premature (sometimes confrontational) commitment to particular mechanisms and strategies. We need to understand the structure of design space and niche space, and trajectories that are possible within those spaces (Sloman 1994*a*, Sloman 1994*b*, Sloman 1998(forthcoming)). This requires collaborative philosophical analysis, psychological and neurophysiological research, experiments with diverse working models of agents, and evolutionary investigations. Some of this exploration can be based in part on powerful new software tools.

Such work is likely to throw up types of architectures that we would not otherwise think of, which will force us to invent new concepts for describing synthetic minds which are not like our own, and help us understand our own by contrast.

ACKNOWLEDGEMENTS & NOTES

We acknowledge support from the UK Joint Council Initiative, The Renaissance Trust, and DERA Malvern, and much help from students and staff at Birmingham, especially Luc Beaudoin and Ian Wright.

Pointers to code and documentation for our toolkit are at http://www.cs.bham.ac.uk/~axs/cog_affect/sim_agent.html

Several papers developing these ideas are in the Cognition and Affect Project ftp directory: ftp://ftp.cs.bham.ac.uk/pub/groups/cog_affect

References

Bates, J., Loyall, A. B. & Reilly, W. S. (1991), Broad agents, *in* 'Paper presented at AAAI spring symposium on integrated intelligent architectures'. (Available in SIGART BULLETIN, 2(4), Aug. 1991, pp. 38–40).

Baxter, J., Hepplewhite, R., Logan, B. & Sloman., A. (1998), Sim_agent two years on, Technical Report CSRP-98-2, University of Birmingham, School of Computer Science.

Baxter, J. W. (1996), Executing plans in a land battlefield simulation, *in* 'Proceedings of the AAAI Fall symposium on Plan execution: Problems and issues November 1996', AAAI, pp. 15–18.

Davis, D. N. (1996), Reactive and motivational agents: Towards a collective minder, *in* J. Muller, M. Wooldridge & N. Jennings, eds, 'Intelligent Agents III — Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages', Springer-Verlag.

Dennett, D. C. (1978), *Brainstorms: Philosophical Essays on Mind and Psychology*, MIT Press, Cambridge, MA.

Goleman, D. (1996), *Emotional Intelligence: Why It Can Matter More than IQ*, Bloomsbury Publishing, London.

McCarthy, J. (1979), Ascribing mental qualities to machines, *in* M. Ringle, ed., 'Philosophical Perspectives in Artificial Intelligence', Humanities Press, Atlantic Highlands, NJ, pp. 161–195. (Also accessible at http://www-formal.stanford.edu/jmc/ascribing/ascribing.html).

McCarthy, J. (1995), Making robots conscious of their mental states, *in* 'AAAI Spring Symposium on Representing Mental States and Mechanisms'. Accessible via http://www-formal.stanford.edu/jmc/consciousness.html.

Newell, A. (1982), 'The knowledge level', *Artificial Intelligence* **18**(1), 87–127.

Simon, H. A. (1969), *The Sciences of the Artificial*, MIT Press, Cambridge, Mass. (Second edition 1981).

Sloman, A. (1978), *The Computer Revolution in Philosophy: Philosophy, Science and Models of Mind*, Harvester Press (and Humanities Press), Hassocks, Sussex.

Sloman, A. (1994*a*), Explorations in design space, *in* 'Proceedings 11th European Conference on AI', Amsterdam.

Sloman, A. (1994*b*), 'Semantics in an intelligent control system', *Philosophical Transactions of the Royal Society: Physical Sciences and Engineering* **349**(1689), 43–58.

Sloman, A. (1996), *Primer of Pop-11*, School of Computer Science, University of Birmingham, School of Computer Science, The University of Birmingham, Birmingham, B15 2TT, UK. (Last revised October 1997. Ftp version available in **ftp://ftp.cs.bham.ac.uk/pub/dist/poplog/**).

Sloman, A. (1998(forthcoming)), What sort of architecture is required for a human-like agent?, *in* M. Wooldridge & A. Rao, eds, 'Foundations of Rational Agency', Kluwer Academic.

Sloman, A. & Poli, R. (1996), Sim_agent: A toolkit for exploring agent designs, *in* M. Wooldridge, J. Mueller & M. Tambe, eds, 'Intelligent Agents Vol II (ATAL-95)', Springer-Verlag, pp. 392–407.

Wright, I., Sloman, A. & Beaudoin, L. (1996), 'Towards a design-based analysis of emotional episodes', *Philosophy Psychiatry and Psychology* **3**(2), 101–126.

PSI: A Theory of the Integration of Cognition, Emotion and Motivation

Christina Bartl & Dietrich Dörner

Lehrstuhl Psychologie II
Otto-Friedrich - Universität
D-96045 Bamberg

Phone: +49/951/863 1861

e-mail: christina.bartl @ppp.uni-bamberg.de
dietrich.doerner@ppp.uni-bamberg.de

ABSTRACT

In this article we describe a theory aiming at the integration of cognitive processes, emotion and motivation. The theory describes the informational structure of an intelligent, motivated, emotional agent which is able to survive in arbitrary domains of reality. This agent is „energized“ by six motives (needs for energy, water, pain-avoidance, affiliation, certainty and competence). The cognitive processes of this agent are modulated by emotional states and processes. By comparing the behaviour of Psi with human behaviour in a complex computer scenario, the model was tested against reality. Subjects were asked to regulate a dynamic system structural identical to the environment of the autonomous agent. First results show striking similarities between artificial and human behaviour as well as differences.

Keywords

Artificial Life, Cognition, Emotion, Motivation, Action Regulation.

INTRODUCTION

In cognitive science there is a focus on cognition when considering action regulation. Emotional and motivational processes, however, play a considerable role in human behaviour triggering cognitive processes. In a state of anger thinking and reasoning differs from processes under „normal“ conditions. Different emotional states even influence perception in a specific manner. – In a long lasting process of action regulation, when humans have to tackle difficult problems, neither emotions nor motives remain constant. Foreseeing that an important problem cannot be solved an individual will feel helpless and this feeling of helplessness will trigger other feelings and can change the current motive. The motive to find a solution for an intellectual task will be replaced by a motive to demonstrate „competence“ as the inability to solve the problem threatens the self-confidence of the individual.

THE PSI THEORY OF ACTION-REGULATION

A single theory of cognitive processes does not succeed in explaining human behaviour. Furthermore it is necessary to include assumptions about the dynamics of emotions and motivations. During the last years we developed a theory – the Psi theory – concerning the interaction of cognitive, emotional and motivational processes. A computer program was constructed to simulate the theoretical assumptions (see Dörner & Hille, 1995; Hille, 1997; Schaub, 1997). The Psi theory is completely formulated in terms of the theory of neuronal networks, but going into details about the inner structure would exeed the aim of this paper

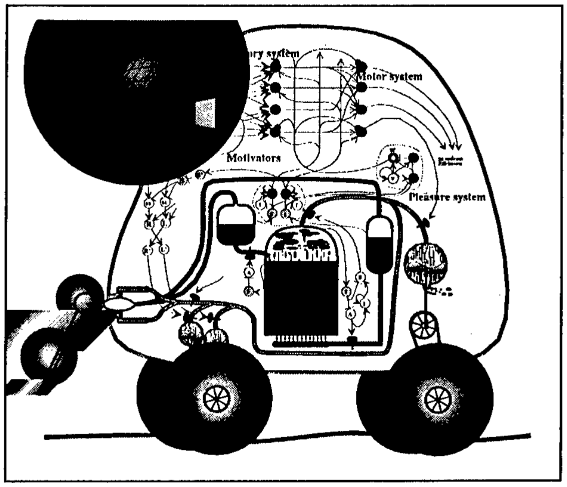


Fig. 1: **Psi as an „autonomous steam engine“.**

The Psi theory includes more than assumptions about single cognitive processes. It aims at a description of the interaction of different cognitive and non-cognitive processes. It is a theory in the tradition of „artificial life“ - research (Steels, 1993). It exists a computer program simulating the theory. The actual version of this computer program is available in internet on page http://141.13.70.49. Fig. 1 shows a possible „materialization“ of Psi as an autonomous steem engine which should care for its existential needs (water and energy). The architecture of the model will be explained below.

Motivation

Fig. 2 shows a rough sketch of Psi's internal structure. At the bottom of fig. 2 the motivational system of Psi is symbolized by a number of „watertanks“. These tanks are mechanical models of „motivators“. „Motivator“ means a system which is sensible for the level of a variable. This should be kept within certain borders (within a setpoint region) by the system. Such variables could be water or energy resources of a system, temperature of a body or any other variable important for life or welfare of a system. When a variable deviates from its set point, a motivator becomes active. In this case there is a need and the motivator will try to launch activities to restore the set point value of the respective variable.

Which motivators are necessary? First of all Psi has to care for its existence. This means that Psi needs (for instance) water and energy. And Psi should preserve its structure; it should avoid pain. Additionally to these „existential“ needs Psi has „informational“ needs, namely a need for certainty, a need for competence and a need for affiliation.

The need for certainty is satisfied by „certainty signals“. An important certainty signal is for example a correct prediction. Acting in a certain domain of reality Psi will learn regularities of its environment. Therefore it will be able to predict the outcomes of its actions and progress of events. If these predictions are correct they will be certainty signals and will fill the „certainty tank“. If the predictions are wrong or if the chain of events does not develop in the predicted way, however this means uncertainty and will decrease the level of the „certainty tank“.

The need for competence is a need for „competence signals“. Each satisfaction of a need, for instance the satisfaction of the need for water, is a signal of competence for Psi. Satisfaction of a need signifies that Psi is able to care for itself. On the other hand a longer lasting period of non-satisfaction signifies inability and therefore is an incompetence signal which empties the competence tank.

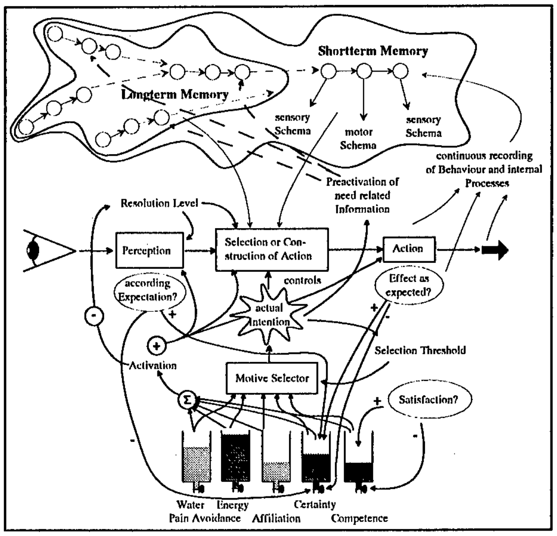


Fig. 2: **The internal structure of Psi. SeeText.**

An empty competence- and an empty certainty-tank launch specific activities. The need for certainty for instance can activate exploration or – depending on the competence (level in the competence tank) – flight. A low level of competence (it shouldn't be too low) will activate „adventure-seeking“, looking for problems the solution of which proves ones own competence.

Group integration is symbolized by the level of the „affiliation tank“. This tank will be filled up by „signals of legitimacy“ (Boulding, 1978) as for instance a smile or a clap on the shoulder. Reports of disapproval serve as signals for nonaffiliation and will empty the „affiliation tank“. – The needs for certainty and for competence are very important for the emotional regulations of Psis behaviour.

Psi's architecture of motivation allows several needs to be active at the same moment. It is therefore vitally important to equip Psi with a selection device, the Motive Selector of fig. 1. This selection device has to select one of the active motives for execution. The motive selected will become the actual intention. An *intention* is a data structure consisting of informations about the goal, about the present state and normally of more or less complete plans for achieving the goal.

The selection device works according to an expectancy – value principle; i.e. it selects the motive with the largest expectancy of success and the largest underlying need. (We call the product of expectancy of effect and amount of the underlying need the **strength** of a motive. So the selection device looks for the motive with the greatest strength.)

Action regulation, memory and cognitive processes

After an intention has been formed, Psi will „run the intention“ to achieve the respective goal. „Running the intention“ can mean different processes. When Psi has a lot of experience with the respective domain of reality its memory will often provide a complete course of action as a chain of operations or locomotions leading from the actual situation to the goal. If this however fails an inbuilt planning procedure will try to construct a course of actions by putting together single pieces of knowledge about operators and event chains. (At the moment this planning procedure is a forward-planning, hillclimbing procedure.)

If planning is impossible due to a lack of information or if planning proves to be not successfull, Psi will use trial-and-error procedures to collect information about its respective environment. Generally Psi organizes its activities according to the Rasmussen - system (Rasmussen, 1983). If possible first of all it tries its highly automatized skills, then it changes to „knowledge-based“ behaviour and the „ultima ratio“ are the trial-and-error procedures.

Psi learns by experience, learns the effects of operators in a specific domain of reality, learns goals and learns chains of events and therefore is able to predict what will happen in the future. But additionally we installed forgetting in the memory of Psi. Forgetting simply is a decay process which continuously diminuishes the strengths of the memory traces. Traces which are rather strong lose less of their strength in time than weak traces which will be destroyed rather quickly. Forgetting has a important function for Psi's cognitive processes. „Punching holes“ into sensory and motor schemata of Psi's memory makes them „abstract“, „hollow“, so that the schemata do not represent concrete images any more, but equivalence classes.

The memory system of Psi is extremely simple and (therefore) powerful. All perceptions and activities are continously recorded. This record is a kind of log of the changing environment, Psis activities and the current intentions. The memory chains representing the immediate past are very dense. Due to forgetting however, memory will consist of single episodes and activities. Memory traces combined with need satisfaction or generation (for instance pain) will be rather strong. Others are weaker and therefore more exposed to decay. Psi has a short term memory which is simply the „head“ of the record. This short term memory without any rupture continues into an episodic memory. Remnants of this eventually form the long term memory. If parts of the longterm memory are reused (in planning for instance), the strength of the respective memory trace is enhanced.

Emotions

The information processing of Psi is „modulated“. This means that all cognitive processes of Psi are „shaped“ according to certain conditions. Such conditions are for instance the strength of the actual intention, the overall amount of all the different needs, the amount of competence and others. These conditions set specific „modulators“. One of these modulators is „activation“ which depends on the strengths of the needs (roughly spoken the amount of activation mirrors, the sum of the strengths of the needs). Activation triggers some other modulators, for instance „resolution level“ and „selection threshold“. Resolution level (RL) is the degree of exactness of comparisons between sensory schemata. As most of the cognitive processes of Psi comprise comparisons between schemata this modulator is very important. Comparisons take a long time at a high level level of resolution, but they will be reliable. Under high pressure (when activation is high) the resolution level is low, comparisons don't need a long time, but the risk of „overinclusiveness“ is high. A low level of exactness will automatically produce the tendency to consider unequal objects and situations as equal. (This is due to certain mathematical reasons which will not be considered here.) Quick planning processes and a high readiness for action will be the result of a low resolution level, but the plans will be rather risky.

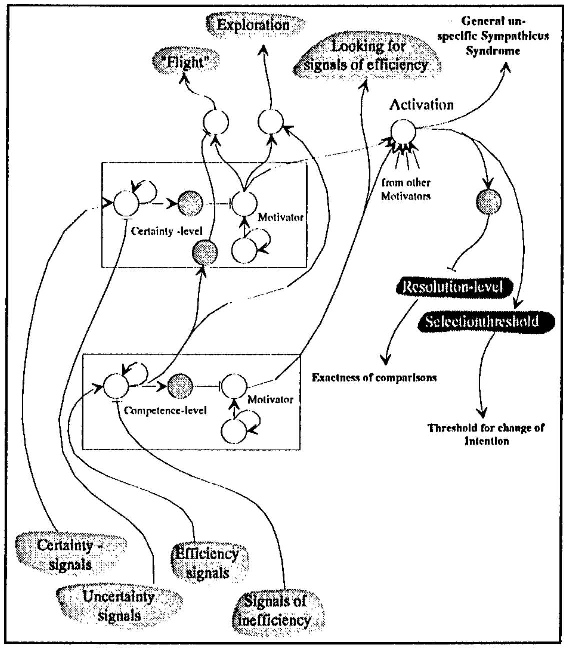


Fig. 3: **Emotional modulations. See text.**

Selection Threshold (ST) could also be called „level of concentration“. ST is the strength of the defence of the actual intention against competitors, against other intentions having the tendency to take over the command. The strength of the different motives is not at all constant in the life of Psi, but changes continuously. Because of consumption the needs for energy and for water continuously increase. But a motive can gain strength by external factors too. If for instance Psi notices in a certain situation that it is easily possible to get water, a tendency to shift to the water-intention will result as now the expectancy value for the water – motive increased. Or if an unexpected event will occur the „need for certainty“ might increase and Psi will exhibit the tendency to explore the (uncertain) environment or will have the tendency to run away and to hide. Or if for instance planning proves to be unsuccessfull, Psi's „self-confidence“ (level of competence) is endangered and Psi will exhibit the tendency to „try its strength“, to prove its competence to itself, for instance by looking for a task which is difficult enough that mastery proves competence, but not so difficult that the risk of failure is high.

If ST is high „behavioural oscillations“, i.e. a rapid change between different intentions will be hindered to a certain degree (Atkinson & Birch, 1970). A high ST prevents Psi on the other hand from using unexpectedly arising opportunities or from reacting to unexpected dangers. Is ST high, the field of Psi's perception will narrow down.

Fig. 3 gives a general impression of the emotional regulations of Psi. We describe these regulations in terms of neuronal networks (as it is realized in Psi). White circles represent activating neurons, whereas gray circles represent inhibiting neurons. The competence and the certainty - level are now represented as the activation state of neurons. Certainty signals enhance the activity of the „certainty-neuron“, whereas uncertainty - signals diminuish this activity. — Satisfaction of a need serves as competence signal and enhances the activity of the „competence-neuron“, whereas non-satisfaction decreases this activity. When the uncertainty level is low (high uncertainty) a tendency for flight or aggressive activities will be observable, depending on the competence level. With a high level of competence Psi will exhibit a tendency for aggression in uncertain situations, whereas with a low level of competence it will exhibit flight tendencies.

Activation triggers the „general unspecific sympathicus syndrome“; i.e. high vigilance and a high degree of readiness to react. Additionally it triggers RL and ST, which modulate cognitive processes, perception, planning activities, memory search. It is obvious that Psis emotions are the result of a rather complex interaction of motivational and cognitive processes together with the modulation of RL and ST.

These modulators (RL and ST) together with the need for certainty and the need for competence produce a lot of „emotional“ forms of behaviour. Psi exhibits fear (expectation of an uneasy event), anxiety („need for certainty“), anger (when unexpectedly Psi is hindered to reach a goal), surprise (unexpected event). This theory of modulations together with the specific motivational structure of Psi constitute a „subaffective“ theory of emotion. A theory, which defines emotions in non-emotional terms. To be able to monitor Psis emotions we gave a human face to Psi which alters according to Psis emotional states. Fig. 4 shows some of the facial expression of Psi in different situations.

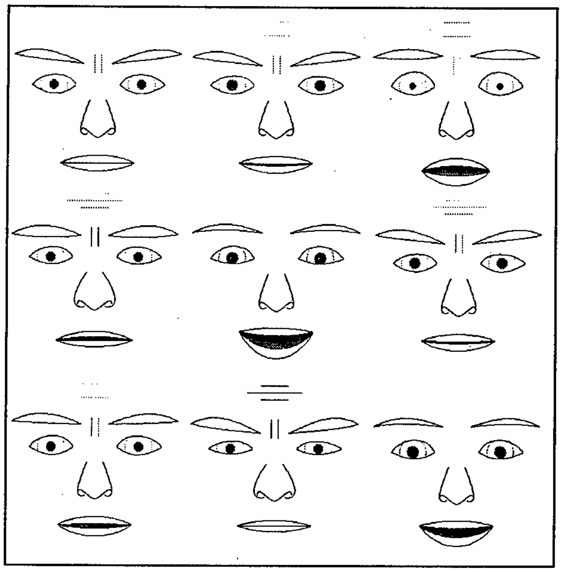


Fig. 4: **Psi's emotions. See text.**

In the upper left corner a resolute Psi can be observed. Psi has a goal and is willing to achieve it against all obstacles. In the upper right corner Psi is seized with horror, helplessly anticipating uneasy events. The middle one face shows Psi in a state of pure joy. The face in the bottom line right shows a joyfull Psi too. You will notice, however, a slight surprise-emotion in this face comparing it with the middle one face. The middle one face in the bottom line shows pain, whereas the face on the right side in the medium line exhibits a state of caution and hesitation. — All these emotions are observable not only in Psis facial expressions, but in its behaviour too[1].

Fig.4 shows what will happen, if you put Psi to a new environment. First the feeling of competence and the feeling of certainty decrease, as Psi is not able to predict what will happen and is not able to care for ist existential needs. But after some learning the respective schemata for appropriate behaviour will be established and Psi is able to cope with its „world“.

[1] The procedure for the facial expressions was programmed by Jürgen Gerdes.

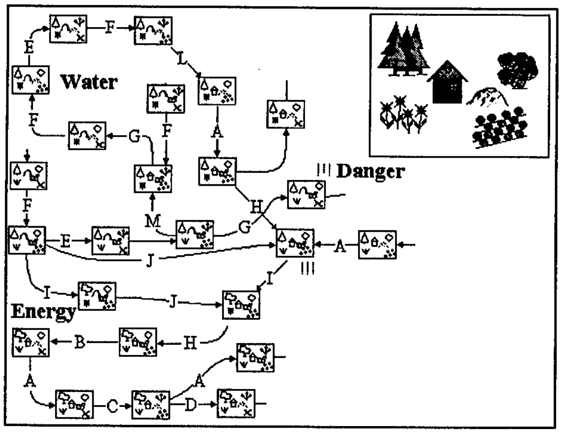


Fig. 4: **An example of the „world“ of Psi and a single „situation“.**

This „world“ is a maze-like environment composed of single „situations“. Fig. 4 shows an example of such a „world“. Psi has to learn how to move from one situation to an other one to arrive at „water“ or „energy“ - situations to satisfy its basic needs. Additionally Psi should learn to avoid dangerous situations. The „situations“ are composed of elements like houses, trees, bushes etc. In the upper right corner of fig. 4 an example of a „situation“ is visible. „To behave“ in such an environment means to manipulate the respective parts of a given situation or to move from one situation to the other one by applying the appropriate operators.

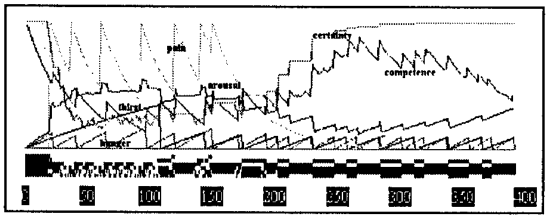


Fig. 5: **Psi's „fate“ in a new environment.**

In fig. 5 some of the internal parameters of Psi when exposed to a new environment are visible. You may observe that first Psi cannot avoid painfull situations and is not able to care for its existential needs („thirst“ for instance increases from cycle 1 to cycle 100 continually as Psi is not able to find water within this time period). But after some learning Psi becomes able to avoid painfull situations and has acquired the capabilities to care for itself.

A COMPARISON BETWEEN HUMAN BEHAVIOUR ANDTHE BEHAVIOUR OF PSI IN THE BIOLAB-GAME

The capability of understanding, explicating and predicting empirical phenomena might help to estimate the value of a theory. The study presented is examining whether the Psi-model succeeds in replicating human behaviour in a complex task.

For that aim we used the scenario BioLab to compare the behaviour of Psi with the behaviour of experimental subjects. We were interested in the similarities and differences between „artificial“ and human behaviour. Differences would possibly point out that basical assumptions of the theory have to be revised. Furthermore the comparison helps to detect the limits of the model explaining human behaviour.

In summary the behavioural test has two objectives: first the results may contribute to the evaluation of the Psi model and the underlying theoretical assumptions. Second the results can give hints to the improvement and the completition of the model of action regulation. By confronting the model with reality necessary modifications and elaborations might be detected.

The scenario BioLab

In the „Biological Laboratory for sugar-based Energy Production“ („BioLab“ factory) subjects are asked to produce certain types of molasses to generate electricity or heat. To modify the molecular structure of the molasses they can use different kinds of catalysts. Under certain conditions, however, the adding of catalysts may cause contaminations. As a result a cleaning of the reactors is necessary. Neither electricity nor heat can be produced until this work has finished. Therefore it is useful to avoid such situations.

The BioLab-system corresponds a maze formally. Subjects can move from one situation to another by using catalysts as operators. They change the structure of the molasses respectively to their actual position in the maze. The amount of operators consists of ten catalysts, some of them needing specific conditions to work. The situations consist of a combination of six dimensions each of them having two valences: either zero or one. This will lead to 64 different situations each represented by a specific combination of these digits.

It is possible to divide the structure of the maze into eight circles, each of them having the valences of the first three dimensions in common. As the eight situations within the circles are highly combined with each other, it is rather simple to move from one situation to another (see fig. 6). In order to leave a circle, it is however essential to have one specific combination of the dimensions four to six. Only this specific situation allows changing between the circles.

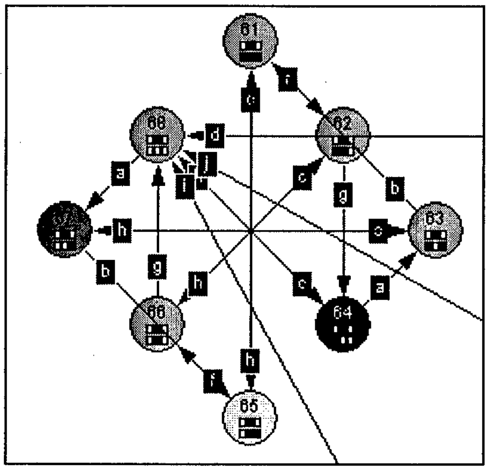


Fig. 6: **The structure of the maze consisting of eight circles, built up by eight situations (figure shows one of eight sections).**

The subjects do not know the formal structure of the maze. They have to explore the BioLab. The situations are visualized by pictures showing the molecular structure of the molasses on the screen. The situation is shown by the characteristics of the molasses in two tanks: they vary with respect to amount, colour and bubbles (see fig. 7).

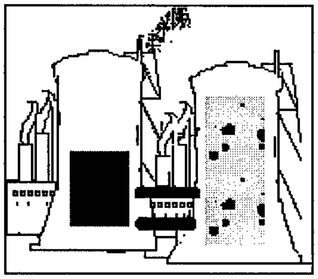


Fig. 7: **The situations of BioLab represented by the different structure of molasses in two tanks.**

To produce energy, it's inevitable to find a way from electricity to heat production and vice versa. Their need for energy is represented by two bars: one showing the actual need for electricity and the other one showing the system's need for heat. The urgency of producing electricity and heat is symbolized by the length of the bar. For example when a subject reaches a situation which provides electricity, the bar will be filled up, no matter how empty it was before. Until the reload is going to happen the electricity resources will be decremented over time.

Electricity as well as heat can be produced in each of the eight circles of the maze. To gain energy a specific combination of the dimensions four to six is essential. As a consequence of getting to situations of satisfaction several times, they will be exhausted. Therefore it is important to find alternatives and to adapt the behaviour to environmental changes.

In summary, handling the BioLab requires capacities of complex problem-solving. Subjects have to explore and regulate a dynamic system with two appetetive and one aversive aims. While they are working on the BioLab game they are coping with a problem identical to the environment of the autonomous agent Psi. Now let's have a look how efficient the laboratory is conducted and how the subjects in contrast to the Psi-model learn to use the catalysts in an effective manner.

The comparison of human and artificial behaviour: efficacy of need satisfaction and of catalysts use

The results presented rely on an experiment conducted with 12 subjects each of them playing the BioLab game for one hour. Each of the subjects had to play under two experimental conditions: first they had to think aloud, second they had to keep tacit. After half an hour of playing the experimental condition changed. Variing the sequence of the two instructions, the subjects were randomly divided into two groups. Most of the subjects were students of psychology from the University of Bamberg.

In general the task was neither too easy nor too difficult for the subjects. All of them succeeded in finding situations where energy production is possible, at least by chance. One subject succeeded in exploring the whole structure of the maze. He/she could intentionally change from one circle to another and has found a efficient way to move from electricity to heat production within the circles.

For a useful comparison between the behaviour of Psi with the behaviour of the subjects we had to parallelize parameters of environment as well as of action time. Whereas the subjects carried out about six actions per minute, Psi conducted more than sixty at the same time. For this reason only the first 360 actions of the model's behaviour protocol were evaluated.

Let us have a look upon the efficiency of managing the BioLab problem: One value representing the performance is the score achieved at the end of the run. Starting with zero, the account increases with a hundred points whenever electricity or heat is produced. Whenever the lab is contaminated, the account decreases by fifty points. Every thirty minutes the account is lowered by one point and finally every use of a catalyst costs one point either.

These statistical results illustrate that human subjects are capable of managing the lab rather good. The mean account is 1314 points after 60 minutes. The variance between the subjects, however, is huge. The subject with the best perfomance gained 2108 points, whereas the worst performance achieved 217 points. The efficiency of the model run is even lower: Psi could only manage to get 120 points in the game. The rather bad performance does not rely on a greater number of contaminations (see tab. 1). Moreover the results of the Psi model show a less effective use of catalysts and therefore a lower rate of needs satisfaction.

	Subjects			Psi
	mean	minimum	maximum	value
account of points	1314.58	217	2108	120
number of contaminations	10	2	16	8

Tab. 1: **Statistic values representing the effeciency of needs satisfaction.**

One value representing the successful use of the operators is the percentage of effective catalyses. Psi used as much catalysts as the average subjects. In contrast to the subjects, however, only 15% out of these caused the molasses to change its characteristics.

The following figure shows a boxplot about the results of the subjects and Psi. The subjects were subdivided in two groups: one of them starting with the instruction „thinking aloud", the other one tacit. The bar in the box indicates the median, within the box there are 50% of the subjects represented. The „whiskers" of the box mark the 25th and the 75th percentile of the distribution. Remarkably the performance of the Psi model would be placed within the area marked by the whiskers in the tacit group. Compared to the subjects thinking aloud Psi's performance is significantly low. Its performance is contrasted by the subject „Ellobo" who achieved the best efficacy of the whole sample.

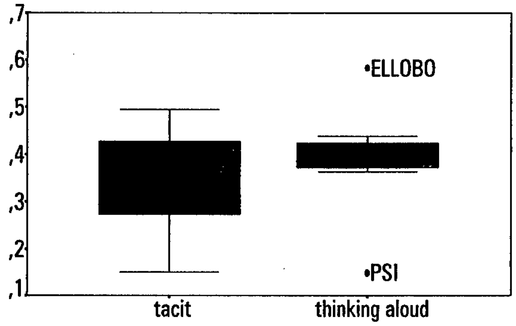


Fig. 3: **The percentage of effective catalyst use between PSI and the subjects. See text.**

First results of single-case studies

Comparing human and artificial behaviour with respect to statistical values will not be sufficient to evaluate a model. Furthermore we tried to replicate the behaviour of each individual by varing the starting parameters of the simulation. By this we created different personalities.

As long as emotional reactions and their impact on information processing are concerned, first results reveal similarities between the model's and the subjects' behaviour.

According to the assumptions of the Psi model subjects show a specific way of action organization: at the beginning they mainly apply a strategy which can be described as „trial-and-error". In the following stage, catalysts are used with respect to success or failure in the past. As a consequence catalysts leading to need satisfaction will be used more frequently in the future, whereas catalysts leading to neutral situations or without any effect will be taken less frequently. Finally catalysts producing contamination will be used more carefully.

As soon as environmental conditions are explored sufficiently, the subjects as well as Psi start making plans. Single action sequences are combined to chains. After gaining a high competence in managing the lab, people as well as our artificial system have an amount of automatisms available. The Rasmussen-system (1983) can be discovered in both: human and artificial behaviour.

Remarkably when trying to replicate the behaviour of single subjects we suceeded in modelling subjects with a rather poor performance, p.e. a quite anxious person producing contamination by the first action he/she made. As a result the subject avoided the catalyst for more than half an hour and as a consequence was not able to produce electricity.

In contrast to more successful subjects the PSI-simulation lacks the capability to reflect on its own behaviour. For this reason strategic flexibility and analogies (i.e. the adoption of learned behavioural sequences on similar situations) can not be found in the simulation runs of out artificial system but in human behaviour.

CONCLUSION

Exploring the similarities and differences of the behaviour of Psi and human behaviour, we found remarkably parallels between the behaviour of Psi and the behaviour of humans. Similar situations provide difficulities for both: humans and Psi. Moreover in comparable situations the model's emotional expression resembles to the expression of the subjects.

There are striking differences as well as similarities. For instance though the planning procedure of Psi is sometimes rather close to what is observable in human behaviour, shows striking differences to human thinking.

Mainly self-reflection is missing. Humans more or less frequently change their thinking and planning procedures by considering the records of their own thinking, analyzing the structure of these records and altering it. Psi is not able to do this. We believe that this is due to the fact that Psi is not able to speak. This „inner dialogue" is one important aspect of higher cognitive functioning in humans. Therefore Psi should be provided with natural language too in order to get the ability of an inner dialogue.

References

Atkinson, J. W. & Birch, D. (1978) (2nd ed.). Introduction to Motivation. New York: Van Nostrand.

Dörner, D. (1994). Eine Systemtheorie der Motivation. (A system theory of motivation). In: Kuhl, J. & Heckhausen, H. (1996): Enzyklopädie der Psychologie – Motivation, Volition und Handlung. Göttingen: Hogrefe, pp. 329 - 357.

Dörner, D. (1997). Motivation in Artificial and Natural Systems. In: Hara, F. & Yoshida, K. (Eds.), Proceedings of International Symposium on SYSTEM LIFE. Tokyo: The Japan Society of Mechanical Engineers & Inoue Foundation for Science, pp. 17 - 22.

Dörner, D. & Hille, K. (1995). Artificial Souls: Motivated Emotional Robots. In: IEEE Conference Proceedings, International Conference on Systems Man, and Cybernetics; Intelligent Systems for the 21st Century. Vancouver, Volume 4 to 5, pp. 3828 - 3832.

Hille, K. (1997). Die künstliche Seele. Analyse einer Theorie. (An Artificial Soul. Analysis of a theory). Wiesbaden, Germany: Deutscher Universitätsverlag.

Schaub, H. (1997). Modelling Action Regulation. Poznan Studies in the Philosophies of the Science and the Humanities, Vol. 56, pp. 97 - 136.

Boulding, K.E. (1978). Ecodynamics. Beverly Hills: Sage.

Steels, L. (1993). The Biology and Technology of Intelligent Autonomous Agents. Berlin: Springer.

Rasmussen, J. (1983). Skills, Rules, Knowledge: Signals, Signs and Symbols and Other Distinctions in Human Performance Models. IEEE - Transactions, Systems, Man, Cybernetics, SMC 13, S. 257 - 267.

Modelling an Empirical Investigation into Memory and Learning in Simple Interactive Search Tasks

Juliet Richardson, Andrew Howes and Stephen J. Payne
School of Psychology, University of Wales, Cardiff,
P.O.Box 901, Cardiff, CF1 3YG, U.K.
+44 1222 874007
RichardsonJ@Cardiff.ac.uk

ABSTRACT

In this paper we address the issue of how initial menu search experiences are encoded and then used to guide subsequent search. We report empirical data from participants searching in a menu structure in which they cannot use spatial strategies and are therefore required to use just the labels to guide themselves. We then describe two cognitive models of menu search: the AYN model which encodes recognition chunks for tried options and gradually acquires positive and negative control knowledge; and an activation-based model which increases the activation of seen and tried options and then uses these activation levels on subsequent trials to guide its search. The data from the activation-based model provides the better fit to the empirical data.

Keywords

Interactive Search, Cognitive Models, Memory, Learning, Computer Menus

INTRODUCTION

Searching through menu structures is a common method of interacting with computers: using software packages, browsing the world-wide web and searching databases are just some of the tasks that require menu search (or *interactive search*). The task of interactive search can be specified in basic terms as requiring a person to make selections in order to find a particular goal. They can either select an option[1] to move forward down a branch of the menu structure, or select an operator to move back up the menu structure (either back just one step or back to the initial starting point). The task of interactive search is therefore different from other problem solving tasks in that people initially do not know what the outcome of operators (moves) will be until they are tried.

In this paper we summarise an empirical investigation of interactive search (for a full report see Howes, Richardson and Payne, in preparation), together with two possible cognitive models of interactive search which are then assessed against the empirical data. We are especially interested in understanding how memories encoded during the initial search experience shape behaviour on subsequent searches for the same goal. In particular, how does a user learn the sequence of choices that leads to a particular goal? The delay between the time when a menu option is selected and the time when that option can be evaluated as correct or not (when the goal is achieved) makes this task more difficult than it might first appear. In many instances, incorrect paths will be explored before the correct route to the goal is found. The user must learn to distinguish those options which were tried and found to be incorrect from those which eventually led to the goal.

One of the most obvious guides as to which options to select during initial search in an unfamiliar menu is the semantics of the labels: labels which are closely related to the current goal should be better choices than those which are more distantly related to the goal (Franzke, 1994; Franzke, 1995). For example, given the goal of checking the spelling of a document in Microsoft Word, the menu header "Tools" seems like a better choice then "Insert" or "Font". However, the label semantics are rarely a sufficient guide to the correct route to a goal. In the above spell-check example, both "Tools" and "Format" might seem equally good choices to a novice user.

There have been several previous cognitive models of how people search in menu structures where the semantics are not sufficient, such as, the IDXL model developed by Rieman, Young and Howes (1996) and the model of expert search behaviour developed by Kitajima and Polson (1995). However, these have tended to focus on the initial search process and the question of how to decide which options to select. Whilst such models are candidate models of how experts and novices search during initial exploration of a menu structure they do not address the problem of how memories of that search are encoded and subsequently used: they leave open the question of how people perform a menu search task for the second, third or fourth time, or how performance improves with experience. A start has been made at addressing these questions with the AYN model (Howes, 1994).

One of the first questions that we can ask is how the initial search is encoded. The experience could be encoded just in terms of the menu labels. For example, the spell-check task might be encoded as selecting "Tools" followed by "Spelling". We term this a lexical encoding. In addition or alternatively, people might exploit the spatial structure of the menu tree and encode their search experience in terms of some spatial representation of the menu structure and the spatial location of the goal within that representation. For example, the spell-check task might be encoded as selecting an item towards the right of the menu bar and then selecting the first item under it.

In addition, it is also possible that users rehearse their choices during the initial search process. For example, at

[1] We use the terms "options", "selections", "choices" and "items" interchangeably to refer to the labels at a menu node.

any one time, users could attempt to rehearse the sequence of choices leading to their current position in the menu structure. Upon reaching the goal, the most recently rehearsed sequence would be the correct route. We would expect rehearsal of this type during search to give rise to a primacy effect (improved performance for the first items in the sequence as they will have been rehearsed for a longer time than the later items).

Alternatively, a recency effect (improved performance for the last items in the sequence) might emerge if users reflect on the actions that they have just performed when they reach the goal (e.g. Howes, 1994).

Therefore, one way to investigate the question of how initial search experiences affect subsequent learning is to look at the order in which the sequence of options leading to the goal are learnt. We use the term "effect of levels" to refer to this differential learning of options at different levels.

EMPIRICAL INVESTIGATION

In this experiment (described in full in Howes, Richardson and Payne, in preparation), we wished to investigate the order in which the choices leading to the goal were learnt, independent of factors such as semantics which may differentially affect different decisions. For example, if one choice was between two semantically plausible options and a second choice was between one highly plausible option and one that was implausible, we would expect the second choice to be learnt more readily than the first. In order to avoid potentially confusing effects of semantics such as this in our experimental data, we used menu trees constructed entirely from labels which had no semantic relationship to the goals. Thus, at each node in these menu trees the level of semantic guidance to the correct choices was the same and there should be no differential effects.

Such semantically unhelpful menu trees are not entirely unrealistic. As pointed out above, semantics are a far from perfect guide in many real-life menus and users are often faced with selecting between two (or more) equally plausible or implausible options.

In addition, we also wished to investigate how people's performance is affected if all possibility of forming spatial encodings is removed and they are forced to just use the labels. In order to achieve this we used two groups of participants. The first group performed the menu search task with randomised positioning of the labels at each node (each time a participant visited a node, the label positions might or might not be swapped around). This manipulation should prevent these participants from encoding their experience spatially. The second group of participants performed the search task in "normal" menu trees where the positioning of the labels at each node was kept constant over time. This should allow us to see what the effect is on learning and performance when participants are forming only a lexical representation of the menu tree in terms of just the labels as compared with when they can also encode spatial aspects of their experiences.

Method

Thirty-two undergraduate students from the Psychology Department at the University of Wales, Cardiff took part in this study for course credits. The experiment consisted of seven trials. On each trial the participant was asked to search for the first target in the first menu tree, followed by the second target in the second menu tree, the third target in the third menu tree and the fourth target in the fourth menu tree. In order to produce a balanced design the order of presentation of the four menu trees was manipulated in order to ensure that each menu tree was presented equally often first, second, third and fourth. The experiment was presented on an Apple Macintosh computer using a program written in MacProlog32. This program automatically recorded the choices made by participants and the time taken to make them.

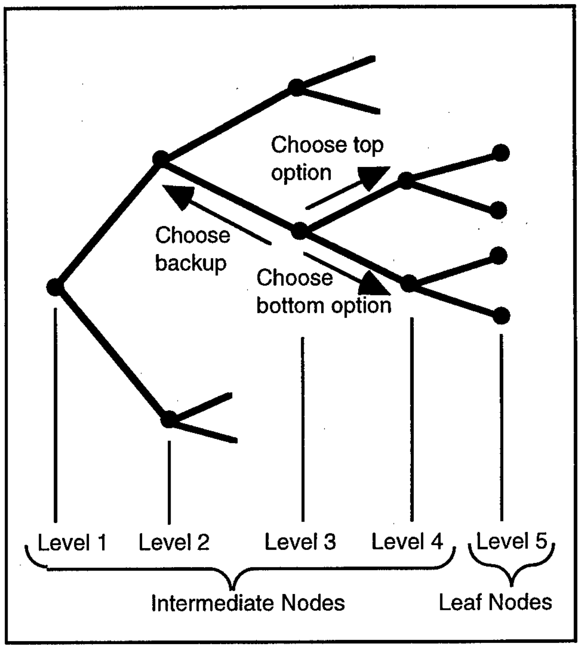


Figure 1: Illustration of the design of the trees used in the experiment.

Each menu tree consisted of five levels with binary choices between a top and a bottom label at each node, as illustrated in Figure 1. The target was one of the choices at a leaf node. At each node there were two options that could be selected to move forward down the tree and a backup option to move back up the tree (except at the top-level root node, where backup is not possible, and in the target node, where backup would allow the participant to review the choices leading to that target). Pairs of semantically related words (e.g. "Carbon" and "Charcoal") were used for the option labels at nodes in these menu trees. These label pairs were placed in different random positions for each participant. There were no close semantic relationships between the label pairs both within and between trees as determined by the experimenters' judgement.

One between-participants factor was manipulated in this experiments. Participants were randomly allocated to one of two equally sized groups. For one group, at each node, the two options were positioned randomly in the top and bottom positions on each visit. For the other group, the two options at each node appeared consistently in the same positions throughout the experiment.

Results

The main results that we are interested in modelling are the order in which participants learnt the sequence of choices leading to the target (effect of levels), and the improvement in performance over trials. Therefore, we shall only consider those analyses. (The full set of analyses can be found in Howes, Richardson and Payne, in preparation).

The effects of levels were investigated by seeing whether participants selected the correct or the incorrect option at the nodes leading to the target. For each target, there was a correct sequence of five actions that would lead directly to that target. For each of the nodes on this correct path, the percentage of correct options chosen by participants on their first visit to that node on each trial was calculated. The action taken on the first visit to each node on each trial was used because this should reflect the effects of long term memory, rather than any effects of temporary memory for recent local search sequences. This measure should therefore show how participants' memory for the correct actions developed with experience.

The data for the trial 2 levels effect are summarised in Table 1. These data were subjected to an Anova to check for main effects of level and node label positioning and for any interaction between these factors. There were no significant main effects of node positioning: F(1, 30) = 0.756, p = 0.39, nor of level F(3, 90)=0.959, p = 0.42. There was not a significant interaction between these variables F(3, 90) = 2.15, p = 0.099. However, t-tests revealed that there were significant differences between levels 1 and 3 for the consistent condition, but that there were no significant differences between any levels in the randomised condition.

The same analysis was carried out for all trials except the first. The data are summarised in Table 2 and were subjected to an Anova to test for main effects of main effects of trial, node option positioning and level and for interactions between these factors.

There was no significant main effect of level, F(3, 90) = 0.58, p = 0.63 on the total number of correct actions. However, there was a significant interaction between the positioning of node options and level, F(3, 90) = 2.72, p < 0.05. There was a significant effect of levels when the positioning of the node options was consistent: the percentage of correct choices made at levels 1 and 2 was higher than at levels 3 and 4. This primacy effect was most pronounced on trials 2 and 3. There was no effect of levels when the label positioning was random: on all trials there were no significant differences between performance at different levels.

Table 1: The mean percentage of correct choices made by participants on the first visit to nodes at levels 1, 2, 3 and 4 on the correct path on trial 2.

	Node option positioning			
	Randomised		Consistent	
Level	M	S.D	M	S.D.
1	52%	35%	72%	22%
2	58%	26%	67%	27%
3	58%	26%	48%	27%
4	61%	23%	61%	21%

Table 2: The mean percentage of correct choices made by participants on the first visit to nodes at levels 1, 2, 3 and 4 on the correct path averaged over trials 2 to 7.

	Node option positioning			
	Randomised		Consistent	
Level	M	S.D	M	S.D.
1	73%	28%	79%	20%
2	76%	22%	79%	23%
3	78%	22%	70%	26%
4	79%	24%	72%	22%

In addition, there was a significant main effect of trial, F(5, 150) = 35.26, p < 0.05 on the total number of correct choices made. Correct choices increased over trials 2 to 4 but not thereafter. There was no significant main effect of node option positioning, F(1, 30) = 0.07, p = 0.79 on the total number of correct choices. There were no other significant interactions.

Performance over trials was calculated in terms of the average number of actions taken to reach the goal on each trial. These data are summarised in Table 3. These data were subjected to an Anova to check for main effects of trial and node label positioning and for any interaction between these factors. There was no significant main effect of positioning of node options, F(1, 30) = 2.09, p = 0.16. There was a significant main effect of trial, F(6, 180) = 52.99, p < 0.01. The number of actions taken to reach the goal decreased significantly over the first four trials but not thereafter. There was no significant interaction between positioning and trial, F(6, 180) = 0.84, p = 0.54.

Conclusions

When spatial consistency was removed the correct choices at all levels within the menu structure were learnt at the same rate. In comparison, when the menu structure was spatially consistent, participants learnt the choices at the top levels first (primacy effect). This result suggests that participants in the spatially consistent condition might have been carrying out some form of spatial rehearsal whilst performing the initial search. The lack of

Table 3: The mean number of actions taken by participants to reach the goal on each trial.

	Node option positioning			
	Randomised		Consistent	
Trial	M	S.D	M	S.D
1	60.2	20.6	52.5	20.6
2	39.6	31.8	24.3	12.7
3	26.9	24.7	20.8	14.4
4	18.1	13.6	12.8	9.3
5	14.8	17.7	11.5	11.4
6	11.0	13.5	8.1	4.4
7	14.6	26.9	7.5	3.3

either a primacy or a recency effect when participants were forced to rely just on the labels to guide their search suggests that no lexical rehearsal took place.

However, even when participants had to rely just on using the labels, they could still learn to perform the task as quickly (in terms of the total number of actions taken to reach the goal) as when the label positions were left constant over time. Therefore, even though participants in the random positioning condition did not appear to be using lexical rehearsal, they were still able to learn the correct choices with practice. Possible accounts of how this might occur are discussed below in the context of two possible cognitive models of the data.

COGNITIVE MODELLING

The initial goal was to develop a cognitive model of learning in menu trees without spatial consistency (i.e. label-based learning only). Such a model can then be used as a starting point for a model of learning in the spatially consistent menu trees, where participants appeared to be using spatial rehearsal.

The main test of the model will obviously be its degree of fit to the experimental data described above. The model should therefore show a flat effect of levels (i.e. equal rate of learning of the choices on the path leading to the goal), together with improvement in performance over trials. Ideally the model should not only show the same pattern of data as the empirical participants, but also the same values. For example, its performance (in terms of the number of actions taken to reach the goal) should improve over the first four trials only but not thereafter.

Two models of label-based interactive search are: (1) The AYN model (Howes, 1994) which encodes chunks for tried items, uses this knowledge to limit the search space on subsequent trials and learns that the most recently selected item is correct when it finds that it is on the right path. (2) An activation-based model which boosts the activation levels of the representations of tried and seen items and then makes decisions based on the relative activation levels to guide its selections.

COGNITIVE MODEL 1: AYN

The first model of interactive search that we describe is the AYN model (Howes, 1994). AYN acquires two types of knowledge as it interacts with a menu structure: *recognition* knowledge and *control* knowledge.

The recognition knowledge consists of episodic chunks that are encoded for every combination of goal, menu and action that the model experiences, regardless of whether the action in question leads to the goal or not. AYN also acquires recognition knowledge that the goal has been achieved. This recognition knowledge supports identification of the menu trees that have been previously visited, which selections made and which goals visited.

AYN uses its recognition knowledge to help guide search in the menu structure during both initial exploration and subsequent searches. A set of rules determines how the model applies this knowledge: (1) if the goal has not yet been achieved then avoid recognised selections; (2) if the goal has been achieved and there is a recognised selection then it should be applied; (3) if there are no recognised selections and the goal has been achieved then a backup operator should be applied. These rules help limit the size of the search space.

AYN also acquires both positive and negative control knowledge through its exploration of the menu structure. This knowledge determines which menu selections lead to the goal and which lead to dead-ends. In AYN working memory is bounded to store only the previous action. Thus, when the goal is achieved AYN only learns positive control knowledge for the selection immediately preceding the goal. On the next trial, when AYN reaches the selection known to be right (i.e. the one before the goal), it learns positive control knowledge for the immediately preceding selection that led to it. In this way positive knowledge is passed back up the structure in a *final-first* way until positive knowledge has been learnt for all the selections leading to the goal.

AYN acquires negative control knowledge in a similar way for selections that lead to dead-ends. In fact, the AYN model was altered slightly from the version reported by Howes (1994) in order to get it to learn negative knowledge from backing up, rather than from cancelling and returning to the start state. AYN was altered so that it learnt that a particular move was "bad" either if that move led directly to a dead-end or if that move led to a node where both options were rated as "bad".

The AYN model was run fifty times (for seven trials on each run) over a five-level binary menu tree (i.e. the same structure as that used in the experiments) to generate the data. The data generated from the model should therefore be in a form that is comparable with that obtained empirically.

Results

The effect of levels on each trial was calculated in terms of percent correct selections made on the first visit to each of the nodes on the correct path. The data for trial 2 only are summarised in Table 4 and Figure 2. The 95% confidence interval was calculated for the empirical mean obtained at each level in the menu (see Grant, 1962, for a discussion of this type of analysis). These confidence intervals are shown in Figure 2. None of the means generated by the AYN model fell inside the confidence interval at any level. At each level, the means generated by the AYN model were higher than those of the experimental participants.

The data for the effect of levels averaged over all trials are summarised in Table 5. The correlation between the percentage of correct choices made at each level on each trial by the AYN model and by the experimental participants was calculated. The correlation was fairly poor, $r = 0.747$, $r^2 = 0.559$.

Table 4: The mean percentage of correct choices made by AYN and the activation-based model on the first visit to nodes on the correct path on trial 2.

	AYN		Activation-based model	
Level	M	S.D.	M	S.D.
1	72%	45%	52%	50%
2	76%	43%	56%	50%
3	68%	47%	54%	50%
4	100%	0%	48%	50%

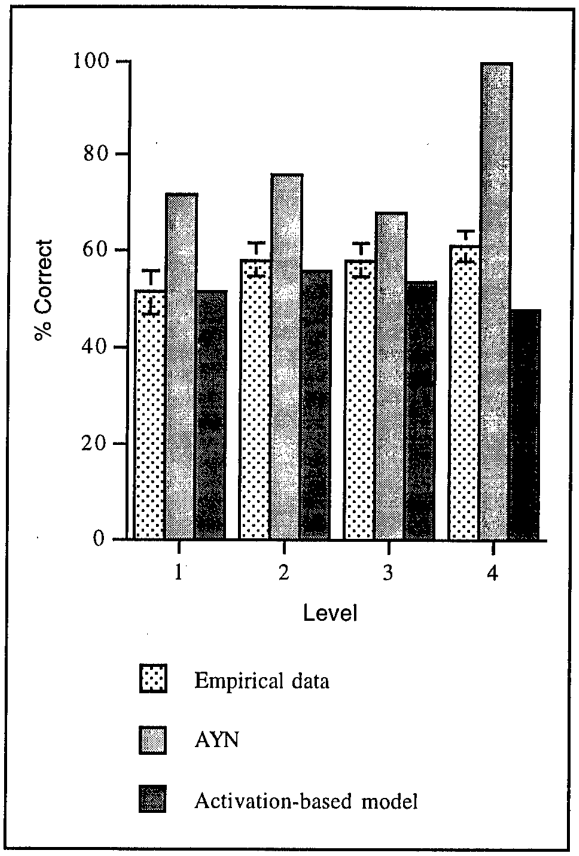


Figure 2: The percentage of correct selections made on the first visits to nodes on the correct path on trial 2 by the empirical participants, the AYN model and the activation-based model

The number of actions taken by the AYN model to reach the goal on each trial was calculated. These data are summarised in Table 6 and Figure 3. The 95% confidence interval was calculated for the empirical mean obtained on each trial, as shown in Figure 3. The means generated by the AYN model fell outside the confidence interval on trials 1, 3, 4 and 5. On each of these trials, the means generated by the AYN model were lower than those of the experimental participants. The correlation between the data generated by the AYN model and the empirical data was calculated. The correlation was very good, r = 0.991, r^2 =0.982.

Table 5: The mean percentage of correct choices made by AYN and the activation-based model on the first visit to nodes on the correct path averaged over trials 2 to 7.

	AYN		Activation-based model	
Level	M	S.D.	M	S.D.
1	90%	20%	83%	29%
2	93%	14%	87%	26%
3	95%	8%	84%	31%
4	100%	0%	84%	29%

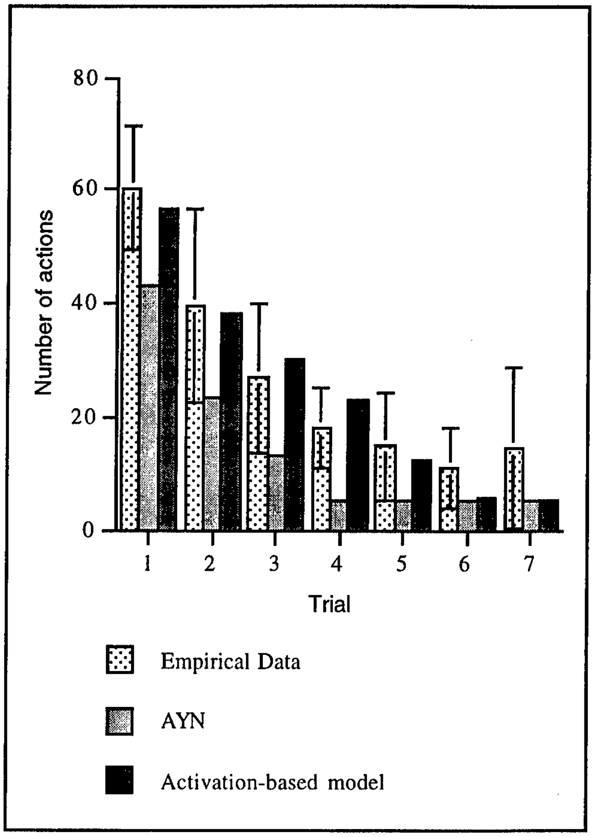


Figure 3: The number of actions taken to reach the goal on each trial by the empirical participants, the AYN model and the activation-based model

COGNITIVE MODEL 2: ACTIVATION-BASED MODEL

The second cognitive model of interactive search that we consider is a simple activation-based model which makes more refined judgements than the AYN model. This model does not just distinguish tried from untried options, but makes four classifications of options: untried; seen and possibly tried; definitely tried; and very recently tried. Most importantly, this model does not acquire any form of AYN-like control knowledge. Instead it simply uses the relative activation levels to determine which choices are correct and which are incorrect.

Table 6: The mean number of actions taken by AYN and the activation-based model to reach the goal on each trial.

	AYN		Activation-based model	
Trial	M	S.D	M	S.D
1	43.3	26.4	56.4	38.9
2	23.6	21.6	38.0	32.8
3	13.2	15.8	30.2	35.7
4	5.2	0.6	23.1	31.5
5	5	0	12.5	21.2
6	5	0	5.6	2.7
7	5	0	5.1	0.9

In this activation-based cognitive model, when an option is seen its activation is boosted by 10 units, and when an option is selected its activation level is boosted by a further 40 units (unseen options have an activation level of zero). Every time a move is made (selection of an option or selecting backup) the activation of all other options decays by 1%. Therefore with time the activation levels of previously tried and seen options decrease. There are 110 decay cycles between trials (to simulate the intervening tasks in the experiments).

The model assesses the activation levels of the options that it encounters in order to infer whether options have been seen or tried before. It then uses these inferences to determine which action to select. If the activation level of an option is less than 1 unit, then the model infers that that option has never been seen before (= untried). If the activation level is between 1 and 20 units then it infers that the option has definitely been seen before and could possibly have been tried some time ago as well (= seen-and-possibly-tried). If the activation level is between 20 and 40 then it infers that the option has definitely been tried before (= definitely-tried), and if the activation level is above 40 then it assumes that the option was tried very recently (= very-recently-tried). The model uses its assessments of the activation of the possible options at a node, together with knowledge of whether the goal has already been achieved or not, in order to decide which action to take.

If the goal has not yet been achieved, the model uses a simple search algorithm similar to that of the AYN model. It avoids options that are assessed as being definitely-tried or very-recently-tried and selects those that are assessed as being untried or seen-and-possibly-tried. At a node with two options which are untried or seen-and-possibly-tried, it prefers to select the untried option. At a node with one option that is definitely-tried or very-recently-tried and one that is untried or seen-and-possibly-tried, it selects the untried or seen-and-possibly-tried option. At a node with only definitely-tried or very-recently tried options, or at a node that is a dead-end, it selects backup. In this way the model searches efficiently through the menu structure to reach the goal.

Once the goal has been achieved, the model again uses a search algorithm based on that of the AYN model. It prefers to select an option that has been assessed as being definitely-tried before, but not very-recently-tried. If not it will select an option that is assessed as seen-and-possibly-tried. It does not select untried options or very-recently-tried options. It also backs up from deadends.

The model was run fifty times over a five-level binary menu tree, for seven trials on each run, to generate the data.

Results

As before, the effect of levels on all trials was calculated in terms of the percentage of correct choices made on the first visit to each of the nodes on the correct path. The data for trial 2 only are summarised in Table 4 and Figure 2. For this model, the means for the percentage of correct choices made at levels 1 and 2 on trial 2 fell within the 95% confidence intervals for the empirical means. The mean percentage correct choices for levels 3 and 4 fell below the confidence interval: the model made fewer correct choices at these levels, on average, that the experimental participants.

The data for the effect of levels averaged over all trials are summarised in Table 5. The correlation between the percentage of correct choices made at each level on each trial by the model and by the experimental participants was calculated. The correlation was good, r = 0.917, r^2 =0.840.

The number of actions taken to reach the goal on each trial was calculated. These data are summarised in Table 6 and Figure 3. The means generated by the model fell within the 95% confidence interval for the empirical means on all trials. The correlation between the data generated by the model and the empirical data was calculated. The correlation was very good, r = 0.964, r^2 =0.930.

CONCLUSIONS

The data generated by the AYN model provided a good fit to the shape of the empirical practice data, although its performance was higher, as would be expected given its 100% accurate all-or-nothing recognition. However, for the effect of levels, the correlation of the AYN data to the empirical data was not as good: AYN showed a recency effect in the learning of the choices on the correct path (i.e. better performance for the last item), whereas no such effect was seen in the empirical data. In addition, the overall level of correct selections made by AYN at the nodes on the correct path was much higher than that of the empirical participants.

The activation-based model gave a very good fit to the empirical data for learning based on labels alone. The correlation between its data and the empirical practice effect data was similar to that seen for AYN. However, unlike the AYN model, the curve that it generated did not differ in absolute value from the empirical data. The data from the activation-based model also provided a reasonable fit to the empirical levels effect data. Its correlation to this data was much higher than the AYN model. In addition, although the curve that it generated had a slightly different shape to the empirical data, the absolute values were not different for two of the four means.

The activation-based model is therefore able to learn the correct menu choices at the same speed and in approximately the same pattern as empirical participants without recourse to either rehearsal or the use of AYN-like control knowledge. Learning occurs simply through the gradual increase in activation of the correct choices relative to other choices.

A slight, but important difference between the activation-based model and the empirical data is that the experimental participants showed a slight (but non-significant) recency effect whereas the model showed an almost completely flat levels effect. However, further trials of the model showed that when the delay between trials is reduced, the model begins to show a recency effect similar to that of the experimental participants. It would be interesting to see whether the small recency effect exhibited by the participants alters with the delay between tasks in a similar way. There is some evidence in straightforward recognition tasks that delay affects recency effects in this way (Wright, Santiago, Sands, Kendrick and Cook, 1985).

GENERAL DISCUSSION

The simple activation-based model provided a better fit to the empirical data for searching a spatially inconsistent menu structure than the AYN model. Importantly, the activation-based model learns the correct path without manifesting a recency effect. This is due to the fact that it doesn't acquire explicit control knowledge. Instead the activation levels of each of the correct choices gradually increases relative to the other choices. On average this rate of relative increase is the same for all of the choices leading to the goal and so a flat effect of levels was observed.

In addition, this result showed that making a simple all-or-none distinction between tried and untried options (as AYN does) led to better performance than was seen empirically. Instead it seems likely that in reality menu users might occasionally be unsure as to whether a particular item has been selected before or merely seen. Errors will therefore arise when users select items that have merely been seen before and not tried. Such uncertainty is akin to a feeling of mere familiarity for a menu item and can be contrasted with definite recollection that an item has been tried before (see Jacoby, 1991, and Mandler, 1980, for an account of the distinction between familiarity and recollection and Payne, Richardson & Howes, in preparation, for an account of the role of familiarity in guiding menu search behaviour). There is currently some debate as to whether familiarity and recollection are indeed separate processes or just end-points on a continuum (see for example, Dodson & Johnson, 1996; Jacoby, 1991). However a simplified version of the single-process model of recognition can be seen as analogous to the decision process underlying the activation-based model. This simplified model assumes that there is a single quantity (activation levels, in our case) underlying different recognition judgements. If the activation level is above a certain criterion, an item will seem merely familiar, whereas if the activation level is above another, higher, criterion the item will be recollected. The activation-based model can therefore be thought of as preferring "recollected" selections over "familiar" ones. It occasionally makes errors by selecting, on the basis of their familiarity, items that had only been seen before and not actually tried. This model therefore had a lower overall level of performance that was not significantly different from that of the empirical participants.

However, this activation-based model only accounts for the data obtained in the situation where participants had to rely on the labels alone and could not exploit the spatial consistencies within the environment. In other words, this model only simulates the possible *lexical* encodings that a person might form whilst navigating through a menu structure, it does not account for any *spatial* representations that might be constructed. As shown in the experiment reported earlier, when spatial consistency was provided, people seemed to perform some form of spatially-based rehearsal. The activation-based model should therefore be extended to model this type of performance, perhaps by adding another "layer" which rehearses spatial location whilst searching through the menu structure.

There are several other possible avenues of development for the activation-based model. One of the first changes to explore might be the effect of different functions for the rate of decay. For example, research shows that the rate of forgetting might be governed by a power law (see Anderson, 1995, for an account). Another possibility is to explore the effects of adding in features such as associations and spreading activation between the activated representations. However, the evolution of this model to date has been driven largely by the goal of modelling empirical data, and any future architectural developments will therefore be made only in response to empirical data that challenge the model.

Finally, both the AYN model and the activation-based model as described here do not use the semantic plausibility of the menu items to guide their search. Other experiments that we have carried out suggest that people do use semantic plausibility, in conjunction with recognition memory, to determine which choices to select in a menu (Payne, Richardson & Howes, in preparation). Both models have, in other versions, been altered to use the semantics of the labels to guide their choices. For both models the effect of this is effectively to limit the search space to just the subset of the menu labels that are judged to be semantically plausible.

ACKNOWLEDGEMENTS

The research reported in this paper was funded under the ESRC's Cognitive Engineering Initiative, Grant No. L127251029.

REFERENCES

Anderson, J. R. (1993). *Rules of the mind.* London: Lawrence Erlbaum Associates.

Anderson, J. R. (1995). *Learning and memory: An integrated approach.* Chichester: John Wiley and Sons, Inc.

Dodson, C. S. & Johnson, M. K. (1996). Some problems with the process-dissociation approach to memory. *Journal of Experimental Psychology: General, 125*, 181-194.

Franzke, M. (1994). *Exploration, acquisition and retention of skill with display-based systems.* Ph.D. Thesis, Department of Psychology, University of Colorado, Boulder.

Franzke, M. (1995). Turning research into practice: characteristics of display-based interaction. *Proceedings of the Conference on Human Factors in Computing Systems* (New York, NY.), Association for Computing Machinery, 421-428.

Grant, D. A. (1962). Testing the null hypothesis and the strategy and tactics of investigating theoretical models. *Psychological Review*, *69*, 54-61.

Howes, A. (1994). A model of the acquisition of menu knowledge by exploration. In B. Adelson, S. Dumais and J. Olson (Eds.) *Proceedings of Human Factors in Computing Systems CHI'94* (Boston, MA.), ACM Press, 445-451.

Howes, A. Richardson, J. & Payne, S. J. (in preparation). Strategies and representations for interactive search tasks.

Jacoby, L. L. (1991). A process dissociation framework: Separating automatic from intentional uses of memory. *Journal of Memory and Language*, *30*, 513-541.

Kitajima, M. & Polson, P. G. (1995). A comprehension-based model of correct performance and errors in skilled, display-based, human-computer interaction. *International Journal of Human-Computer Studies*, *43*, 65-69.

Mandler, G. (1980). Recognizing: The judgement of previous occurrence. *Psychological Review, 87*, 252-271.

Payne, S. J., Richardson, J. & Howes, A. (in preparation). Strategic use of familiarity and plausibility in display-based problem solving.

Rieman, J., Young, R. M. & Howes, A. (1996). A dual-space model of iteratively deepening exploratory learning. *International Journal of Human-Computer Studies*, *44*, 743-775.

Wright, A. A., Santiago, H. C., Sands, S. F., Kendrick, D. F. & Cook, R. G. (1985). Memory processing of serial lists by pigeons, monkeys and people. *Science*, *229*, 287-289.

LICAI+: A Comprehension-Based Model of The Recall of Action Sequences

Muneo Kitajima
National Institute of
Bioscience and Human-Technology
1-1 Higashi Tsukuba Ibaraki 305, JAPAN
Tel: +81 (298) 54-6731
E-mail: kitajima@nibh.go.jp

Rodolfo Soto and Peter G. Polson
Institute of Cognitive Science
University of Colorado
Boulder, CO 80309-0345, USA
Tel: +1 (303) 492-5622
E-mail: {soto, ppolson}@psych.colorado.edu

ABSTRACT

This paper presents a model of occasional use of functions of an application by an experienced user of an environment like Windows 95 or the MacOS. We have developed a simulation model, LICAI+, that assumes that users store episodic records of correct steps discovered by exploration or told to them during training. They then use the application display and their goal as retrieval cues in attempts to recall these episodes later. The model predicts, and supporting data show, that tasks that violate the label-following strategy are not only hard to learn by exploration but also difficult to remember even if the correct steps have been previously presented.

Keywords

cognitive model, learning by exploration, label-following strategy, LICAI+

INTRODUCTION

Experienced users of an environment like Windows 95 or the MacOS are *occasional* users of many applications (e.g., a graphics package). Furthermore, many functions of a frequently used application like a word processor are only used occasionally (e.g., constructing and editing a table). Thus, a large majority of the *different* tasks undertaken by skilled users are performed infrequently (Santhanam & Wiedenbeck, 1993).

Such patterns of occasional use should constrain the design of usable computer systems. Ideally, such systems should consistently support learning by exploration. At a minimum, they should facilitate memory for action sequences learned by demonstration or by being looked up in a manual. The ease of recalling infrequently performed functions can be a major determinate of usability. This is not a novel claim. For example, the designers of the Xerox Star had very similar insights (Bewley, Roberts, Schroit, & Verplank, 1983; Smith, Irby, Kimball, Verplank, & Harslem, 1982). This paper presents a theoretical model of recall of tasks that have been done once or a few times and data supporting the model.

LICAI+ is a model of recall of occasionally used action sequences. LICAI+ assumes that users store episodic records of correct steps discovered by exploration or told to them during training. They then use the application display and their goal as retrieval cues in attempts to later recall these episodes. The resulting model of the recall process is similar to models of text recall (Wolfe & Kintsch, submitted).

LICAI+ is an extension of LICAI[1] (Kitajima & Polson, 1996; 1997) which is a model of the processes involved in comprehending task instructions and using the resulting goals to guide successful exploration. Both LICAI and LICAI+ are based on Kintsch's (1986; in press) construction-integration theory of text comprehension. LICAI+ adds to LICAI the processes involved in encoding and successfully retrieving encodings of correct actions. LICAI+ assumes that successful performance of occasionally performed tasks involves a mixture of recall of episodes of correct actions and problem solving if recall fails. The model is related to Ross' (1984) and Rickard's (1997) models of skill acquisition.

Following a general description of the LICAI+ model, we present a theoretically motivated analysis of recall of occasionally performed action sequences. Readers interested in a more detailed descriptions of the LICAI model should consult (Kitajima & Polson, 1995; 1996; 1997). In support of the LICAI+ model and our theoretical analysis we compare our simulation results with data reported by Franzke (1994; 1995) and Soto (1997). In conclusion, we describe design implications of our results. We demonstrate that *both* ease of learning by exploration and good recall are supported by similar attributes of an interface.

DESCRIPTION OF LICAI+

LICAI+ simulates skilled Mac users in an experiment where they are taught novel tasks using a new application, Cricket Graph III. The task instructions are very explicit but do not contain any information about how to perform the task. Then, at some later time ranging from several minutes to a week, they are tested for retention of these skills when given the task descriptions and the displays generated by the application as retrieval cues. Users attempt to perform each task by exploration and/or recalling an action sequence. However, hints are given by the experimenter if users cannot discover correct actions by themselves.

[1] LICAI is an acronym of the LInked model of Comprehension-based Action planning and Instruction taking. When LICAI is pronounced [li kai], the pronunciation represents a two-kanji Japanese word, 理解, meaning 'comprehension.'

LICAI simulates comprehension of task instructions and hints, the generation of goals, and the use of these goals to discover correct actions by exploration. LICAI+ adds to LICAI processes that encode successful actions and retrieve them after a delay.

Goal Formation

LICAI's action planning processes contain limited capabilities to discover correct actions by exploration. These processes are controlled by *goals* generated by comprehending task instructions and hints. LICAI assumes that goal-formation is a specialized form of the normal reading process in which task specific strategies generate inferences required to guide goal formation. LICAI's goal-formation process is derived from Kintsch's (1988; in press, Chapter 10) model of word problem solving.

Kintsch's model takes as input a low-level semantic representation of problem text, the *textbase*, and processes it sentence by sentence. The result is a *problem model.* Construction of the problem model makes extensive use of comprehension schemata which elaborate the original text representation with problem domain specific inferences.

LICAI incorporates comprehension schemata that transform relevant parts of the textbase for the task instructions and hints into goals that control the action planning process. Propositions that describe actions on task objects in the textbase are recognized and further elaborated by specialized task domain schemata to generate a more complete description of a task. For example, consider a graphing task in which the user was given the instruction, Plot a variable named 'Observed' as a function of a variable named 'Serial Position.' LICAI transforms this task description into the propositional representations of two sentences. 1) Put 'Observed' on the y-axis, and 2) Put 'Serial Position' on the x-axis. The representations of the last two sentences are then transformed into *task goals* that control the action planning process. Terwilliger and Polson (1997) demonstrated that users actually perform this transformation.

In the studies described in this paper, experimenters gave hints of the form 'perform a specific action on a specified screen object' (e.g., pull-down the **Options** menu). LICAI requires that these text or verbal descriptions of an action on an object have to be transformed into a goal, a *do-it goal,* that specifies a specific object on the screen and/or legal actions on that object. Specialized comprehension schemata carry this transformation. See Kitajima and Polson (1997) for extensive descriptions of comprehension schemata.

Action Planning

The heart of LICAI is the action planning processes. LICAI assumes that successful action planning involves linking propositional representations of a goal (e.g., create a new graph), the screen object to be acted on (e.g., the **Graph** menu), and an action to be performed on that object (e.g., press and hold). The most critical of the three links is the link between the goal and the correct screen object. This link can be retrieved from memory or generated by an exploration process.

Skilled Users

Kitajima and Polson (1995) developed a version of the action planning process used by *skilled* users of an application. This model represents an arbitrary sequence of actions required to perform a task as hierarchical goal structure that is retrieved from long-term memory and used to generate the actions. A task is decomposed into a sequence of task goals. *Task goals* refer to actions (e.g., edit) on a task object (e.g., graph title). Each task goal is linked to an ordered sequence of one or more *device goals.* Each device goal specifies a unique object on the screen (e.g., the **Options** menu, the graph title) and the state of the object (e.g., highlighted) after it has been acted on. Thus, skilled users retrieve the critical links between goal and screen object from memory. However, Kitajima and Polson (1995) did not describe how such goal sequences are learned or how they are retrieved from memory.

New Users

When a *new* user of an application attempts to perform a task for the first time, Kitajima and Polson (1997) assumed that they have a task goal but not the device goals. LICAI can simulate exploration by generating the correct actions for a novel task without the device goals if the task goal can be linked to correct screen objects by LICAI's action planning processes.

A task goal is a proposition with two arguments describing a task action and a task object (e.g., hide legend). If a correct object on the screen has a label representing either one of these concepts (e.g., a menu labeled "hide"), the representation of the object will be linked to the task goal. LICAI will retrieve the correct actions (e.g., move the cursor to the object and press-and-hold) on this object from long-term memory, completing the necessary links to generate actions. We and numerous other researchers have called this linking process *the label-following strategy* (Franzke, 1994; Franzke, 1995; Kitajima & Polson, 1997; Polson & Lewis, 1990;. Rieman, Young, & Howes, 1996). Thus, the critical links can be generated to mediate successful exploration. The label-following strategy is the only method that LICAI has for learning by exploration. If there is no direct link between the task goal and the correct object, users must be given a hint.

LICAI+'s Encoding and Recall Processes

LICAI already incorporates a model of encoding and recall of goals based on the Kintsch and Welsch (1991) model of text recall. They assumed that the textbase is stored in episodic memory during the comprehension process. The strength in episodic memory of a given element of the textbase is determined by the number of cycles it stays in working memory and the activation levels it achieves during each cycle. LICAI+ generalizes this model to the encoding and recall of successful actions. LICAI+ also incorporates assumptions from the Wolfe and Kintsch (submitted) model of story recall that enables us to compute predicted recall probabilities.

Encoding Process

LICAI+ assumes that encoding and storage of a successful action is just a special case of the comprehension process. The model "comprehends" the results of a successful action during training. A comprehension schema creates a representation of the successful action which is stored in memory during the comprehension process.

There are two forms of this encoding. The first includes the task goal, the object acted on, and results of the action if the label-following strategy can discover the correct action. The second case is defined by the failure of the label-following strategy. The experimenter gives a hint which is transformed into a do-it goal by the instruction comprehension processes. A do-it goal specifies an action on a screen object (e.g., Pull-down the **Options** menu). The do-it goal is included in the encoding of the successful action in this second case.

LICAI+'s goal formation, action planning, encoding, and retrieval processes are implemented as special cases of Kintsch's (1988; in press) construction-integration theory of text comprehension. Each process is modeled by one or more iterations of a general construction-integration cycle.

The following is a description of the encoding and recall cycles. See Kitajima and Polson (1997) for detailed descriptions of the remaining processes.

The construction phase of the encoding process generates a network of propositions that contains the following representations:

1) the task goal,
2) the do-it goal (if a hint was given),
3) the acted-on object,
4) its label (if the acted-on object is labeled),
5) salient changes in the display state caused by the action (e.g., menu dropped),
6) the display caused by the action (e.g., a pull-down menu),
7) a special encoding node that links the nodes 1, 2, 3, 4, and 5 with the strengths defined by an analyst.

In addition, the fundamental linking mechanism assumed by the construction-integration theory, the argument overlap mechanism, is applied to connect any two propositions in the network sharing arguments. Figure 1 illustrates a network generated for encoding a step of pulling down the **Legend** menu. This action caused a pull-down menu to appear with menu items, **Hide**, **Show**, **Move**, and **Arrange**.

The integration phase of the encoding process is performed using a spreading activation process. The nodes in the network can be partitioned into sources of activation, targets of activation, and links between sources and targets. In the encoding process, the representations of screen objects, the task goal, and the do-it goal serve as sources of activation. In Figure 1, these nodes are shaded. The encoding node is the target. The results of the integration of the network are stored in episodic memory.

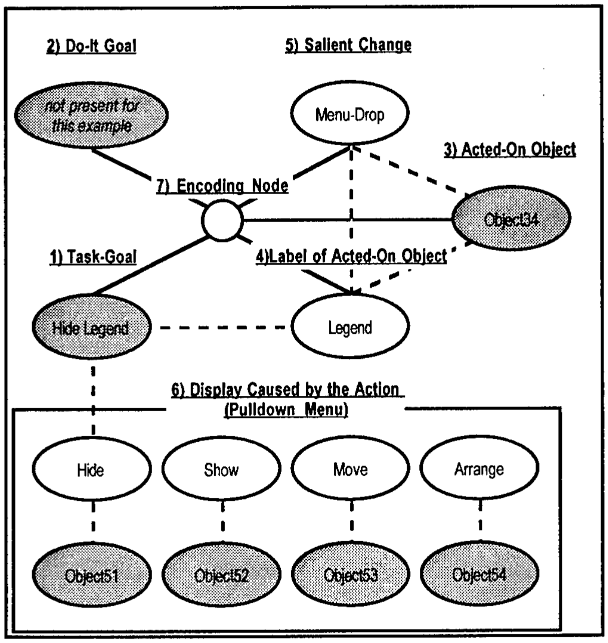


Figure 1. A diagram showing the propositional network generated by the construction subprocess in the encoding process. The dotted lines represent the argument overlap links. The solid lines connecting nodes, 1 through 5, with the encoding node, 7, are special links defining the encoding process.

At the end of training, episodic memory contains the nodes representing the textbase for the task instructions and hints, and the nodes participated in encoding processes for the correct steps. The strengths of links between these nodes are determined by the pattern of activation levels achieved in respective integration processes for text comprehension and encoding.

Recall Process

The recall process of LICAI+ assumes that users employ the task goal and the current display representation as retrieval cues. The recall process retrieves nodes in episodic memory that are linked to these cues. Nodes from episodic memory are sampled with replacement until the model retrieves an encoding of a step or retrieves a do-it goal (i.e., the action planning representation of a hint).

The predicted sampling distribution for retrieving nodes from episodic memory for a given set of retrieval cues is calculated by using a sampling probability matrix. This matrix is a fully interconnected matrix generated from the original episodic memory network. Following Wolfe and Kintsch (submitted), the sampling probability matrix is generated by two steps: 1) dividing each link strength in the episodic memory network by the maximum link strength, 2) for any two nodes linked by an indirect path, assigning the product of the strength values of the link segments in the path to their link strength.

Any nodes that are directly linked with the retrieval cues in the sampling probability matrix are retrievable. The probability of retrieving a retrievable node in a single memory sampling trial is proportional to its relative link strengths with the retrieval cues.

Sampling is with replacement, and sampling terminates on retrieval of one of the step encodings or a do-it goal. These assumptions enable us to calculate the recall probability distribution for step encodings and do-it goals (recall targets).

Action Planning After Recall

LICAI+ attempts to act using the retrieved step encoding or the hint. If the step encoding or the hint generates the correct action, the model successfully recalls the current step. However, there are no explicit order cues in the encoding of each step, so the model can retrieve steps out of order or retrieve hints that don't apply to the current display. In this case, the retrieval process fails, and the model has to explore the interface again as on the training trial. The exploration will succeed in performing the correct action if the label-following strategy works for this step.

AN ANALYSIS OF RECALL OF OCCASIONALLY PERFORMED TASKS

The basic claim of LICAI+ is that how a step in a task is learned, by exploration or with hints, determines how that step is encoded and retrieved. Thus, we distinguish between label-following (LF) steps or tasks, and non-label-following (NLF) steps or tasks where the label-following strategy fails for lack of linking shared concepts.

Franzke (1994; 1995) and many others have shown that LF steps are rapidly discovered and "accurately" recalled. However, it is hard to distinguish between rediscovery and recall of a step after one training trial because both recall and discovery processes can have similar latency distributions.

Soto (1997), in an analysis of a large number of different graphing tasks using Cricket Graph III, showed that NLF tasks have some LF steps, usually toward the end of their action sequences. The task 'hide legend' is a good example. The first two steps (pull-down the **Options** menu, and select **Show Graph Items...**) are NLF steps. No menu label matches the task goal. The third step (clear the check box labeled by Legend) is an LF step. The last step (click OK) is a highly over-learned action that closes a dialog box and terminates the action sequence.

Rodriguez (1997) and Soto (1997) found that the first NLF step in the hide legend task is the source of the difficulties that users have with this task. Almost all users required a hint to complete the first step. Franzke (1994; 1995) found a highly significant interaction for number of hints between number of targets (screen objects) for possible actions on the screen and LF versus NLF steps. There are many targets for possible actions on the first step of any task. Thus, we would expect first steps to be especially problematic. Once users are given the hint "pull-down the **Options** menu" in the hide legend task, there are only 7 menu items on that menu.

We have used two versions of the hide legend task in the simulation described in the following sections. The first version was a simulation of performing the hide legend task using Cricket Graph III, Version 1.5.3 described above. We will refer to this as the NLF scenario. The other version of the simulated task used a hypothetical version of Cricket Graph III that added a **Legend** menu to the menu bar. The items on this menu were **Show**, **Hide**, **Move**, and **Arrange**. This version of the hide legend task requires two steps (select **Hide** from the **Legend** menu) using this hypothetical interface. We will refer to this simulation as the LF scenario. Our discussion will focus on recall of the first step for each of the two versions.

SIMULATION

A Mathematica program was developed implementing processes incorporated in LICAI+ and simulating responses from Cricket Graph III for correct actions in the hide legend task. Training was simulated by assuming that each step was performed correctly with hints given for the first NLF step. The following processes are simulated for the training: the comprehension process that generates goals and comprehends hints, storage in episodic memory during comprehension, retrieval of goals from episodic memory, and action planning, encoding of successful actions, and storage in episodic memory.

Representations of the task instructions, hints, and interface displays were coded and input to the simulation. The simulation also incorporated extensive knowledge about the basic Macintosh interface conventions for each screen object. For example, the **Options** menu item affords pull-down, and the **Options** menu item causes menu-selection, and so on. Other knowledge about actions, including moving and dragging the mouse pointer, and single- and double-clicking the mouse button, etc., was incorporated into the model.

Simulation of Training

Training on each of the scenarios for the hide legend task was simulated in several encoding conditions as described below. At the end of training, episodic memory included nodes representing the task instructions, the hint (for the NLF scenario), the acted-on object and its label for each step, and the display generated by the application. The link strengths of nodes in episodic memory are proportional to the activation level of these nodes obtained in the encoding cycle.

Encoding Bias

In encoding cycles, we manipulated the relative strengths of the links between the rest of the network and the links between the network and the task and do-it goals. The motivation for such manipulations is a fundamental property of the action planning process. The action planning process will *not* work unless the links between the current task, or do-it goal, and the rest of the network are much stronger than the rest of the links in the network. These strong links cause a goal to dominate the integration subprocess. This subprocess selects the object to be acted on and the action to be performed on each step of the task. Manipulating relative strengths of the links between the goal and the rest of the network enables us to explore the hypothesis that the goal may dominate *both* action planning and encoding processes.

Encoding processes have been simulated under three conditions. In task goal biased encoding condition (TG), we generated a network by multiplying by a factor of 4 the strengths of links from the task goal. The strengths of the links from the do-it goal were not changed. In Figure 1, three links from the task goal (hide legend) are strengthend by a factor of 4. In do-it goal biased encoding condition (DIG), the strengths of the links from the do-it goal were multiplied by a factor of 4, and those from the task goal remained unchanged. In the neutral encoding condition (N), no multiplication factor was applied. The NLF scenario was simulated using the TG, DIG, and N conditions. The LF scenario was simulated for the TG and N conditions since hints are not required and there is no do-it goal for the LF scenario.

Simulation of Recall

The recall cues are the task instruction and the representation of task goals used in the action planning process in training trial, and the initial display for the first step. In each simulation, nodes in the episodic memory that match the representations of the cues were identified, and then the probability distribution of retrieving the recall targets were calculated. The recall targets were two encoding nodes for the LF scenario, and the do-it goal and four encoding nodes for the NLF scenario.

Recall after LF training

The probabilities of recalling the encoding of the first step for the LF scenario for TG and N bias conditions are given in Table 1. In the LF scenario, the encodings of the first and second steps are linked to the task goal. In the TG condition, the probabilities of recalling the encoding for each of the two steps was nearly equal since the task goal dominated the encoding process, reducing the influence of the application display. Thus, the model retrieved the representation of the first step a little more than 50% of the time. In the remainder, the model retrieved representation of the second step blocking the successful retrieval of the first step.

Correct performance of both steps is mediated by the same task goal, and the encodings are linked strongly to the common task goal in the TG condition. One implication of these results is that the encoding of a multi-step LF task will not reliably be retrieved by the combinations of task goal and display cues on each step. Thus, correct performance will depend on a mixture of successful recall and the label-following strategy. However, by lessening the biasing on the task goal in the N encoding condition, the display cues made a much stronger contribution to the encoding process and significantly increased the probability of correctly recalling the encoding of each step.

Table 1. Probabilities of recalling the do-it goal or the encoding of first step for the LF and NLF scenarios. TG, N, and DIG stand for task goal biased, neutral, and do-it goal biased encoding condition, respectively.

	LF Scenario		*NLF Scenario*		
	TG	*N*	*TG*	*N*	*DIG*
Probability of recalling the do-it goal	N/A	N/A	.027	.253	.618
Probability of recalling first step encoding	.551	.736	.251	.446	.177
Total	.551	.736	.278	.698	.795
Predicted Hints	N/A	N/A	.722	.302	.205

Recall after NLF training

The probabilities of recalling the encoding for the first step and the do-it goal for the NLF scenario in the TG, DIG, and N bias conditions are given in Table 1. For the NLF scenario, the row labeled *Total* gives the probability of correctly performing the first step. LICAI+ cannot perform the first step without recalling the encoding or the do-it goal. The entries for Predicted Hints are, 1–*Total.*

Manipulation in the NLF scenario of the bias has a huge impact on recall performance. In the TG biasing condition, the probability of recalling the do-it goal is small. The task goal dominates the encoding process and the do-it goal has very weak, indirect links to the task goal. The task goal does have links to all four encodings of each step. The probabilities of recalling each step encoding are almost equal, .251, .227, .180, and .315, respectively.

In the N encoding condition, both the recall probabilities for the do-it goal and the first step encoding increased compared with the TG encoding condition. The reason is the same as the LF case. The display cues become more effective in recall process. Included in these cues is the label for the **Options** menu which is directly linked to the do-it goal. Thus, the initial display is a more effective retrieval cue for both the encoding of the first step and the do-it goal.

On the other hand, in the DIG condition, all links involving the concept Option are very strong. This enhances the effectiveness of the representation of the **Options** menu as a retrieval cue and strengthens the representation of the do-it goal in episodic memory, making it easier to retrieve.

COMPARISONS WITH USER PERFORMANCE

Franzke (1994) and Soto (1997) have done studies relevant to evaluating LICAI+'s recall predictions. For NLF steps, the model predicts that users will require a hint to successfully perform the step if they fail to recall the correct step encoding or hint. We used the best available measure of recall, proportion of subject

Table 2. Proportion of times at least one hint was required for steps categorized by link type, training (exploration) and recall trial (short or long delay). From Franzke (1994).

Link Type	*Training*	*Short Delay*	*Long Delay*
Exact Match	.07	.00	.14
Synonym	.08	.02	.18
Inference	.42	.07	.29
No Link	.88	.05	.60

requiring a hint on a task or step. However, this variable does not provide an unambiguous measure for evaluating the recall predictions for LF steps and tasks. Both successful recall and the label-following strategy can generate correct actions within 10 seconds.

For LF steps and tasks, LICAI+ predicts that no hints should be required during training or on recall trials. However, Rieman (1996) and Rieman, Young, and Howes (1996) found that users will explore an interface before taking the initial correct action predicted by the label-following strategy. This initial exploratory behavior can lead to long latencies and hints on LF steps that are outside the scope of LICAI+.

Description of Available Experimental Data

We first present experimental data from Franzke (1994) and Soto (1997) focusing on the proportion of hints required on training and recall trials.

Description of Franzke (1994)

Franzke (1994) had four groups of 20 participants create a graph and then perform 9 editing tasks on the graph using one of four graphics applications, Cricket Graph I or III, or one of two versions of EXCEL. During training, participants did the task by exploration, receiving hints when necessary. Half the participants in each group were tested for retention after a 5 minute delay (short delay), and the remainder were tested after a 7 day delay (long delay).

Franzke classified each step in each task into one of four categories according to the relationship between the task goal for each step given in her instructions and the label of the object to be acted on for that step. Her exact match and synonym categories are examples of LF steps. In her third category an inference is required to link the correct object and the task goal. In the fourth category (no link) there is no meaningful link between the screen object and task goal. The latter two categories are both examples of NLF steps.

The results relevant to LICAI+ from Franzke's (1994) experiment are shown in Table 2. The table shows the proportion of times that at least one hint was required on a step, with the steps categorized by link type, training (exploration) and recall trial (short or long delay).

Description of Soto (1997)

Soto (1997) performed a study replicating and extending Franzke's results. Soto's 19 participants were trained on a series of 33 graph editing tasks using Cricket Graph III and were tested for retention after a 2 or a 7 day delay. All participants were experienced Macintosh users who had not used a graphing application. Editing tasks were carried out on three types of graphs: histograms, pie charts, and bar charts. The 11 histogram editing tasks and the first of the 11 bar and pie chart editing tasks were used as warm-up tasks, and these data are not included in the results described below.

Table 3. Observed proportions of tasks requiring at least one hint as a function of task type and training and delay. From Soto (1997).

	Session 1		*Session 2*	
Task Type	*Training*	*Short Delay*	*Long Delay*	*Short Delay*
LF/C	.01	.00	.00	.00
LF/U	.19	N/A	.12	N/A
PL/C	.84	.26	.46	.11
PL/U	.58	N/A	.29	N/A

Four out of the 10 experimental pie and bar chart editing tasks were unique (U) to that graph type and occurred once during training and testing. An example is "stand out a pie slice." Six of the tasks were common (C) to both graph types and occurred twice during training and recall sessions. An example is 'hide legend.' The delay between the two presentations of the common tasks averaged about 7 minutes. In Soto's data analysis, the second occurrence of a common task was treated as a recall trial with a short delay. His participants had no trouble recognizing the second occurrence even with a change in graph type.

Soto classified his editing tasks into three categories. Label-following (LF) tasks required acting on objects whose labels were semantically related to the goal. Thus, all steps in these tasks were equivalent to Franzke's direct match and synonym step types. Direct-manipulation (DM) tasks required acting on the task object (e.g. pie slice) mentioned in the task goal. These data are not discussed as it is beyond the scope of this version of LICAI+. Poorly-labeled (PL) tasks did not support either label-following or direct-manipulation violating the label-following strategy. Occasionally, a task supported label following as well as direct manipulation (e.g., 'Change the graph title to "Year of Production"'). For this reason, the tasks were classified based on the method used by the subject, rather than on a priori criteria.

Soto's analysis is by task rather than by the step level. The typical PL task has one or two initial NLF steps. Soto's findings and Franzke's (1994) results suggest that the initial NLF step has the largest impact on users' performance. Previously, we summarized Franzke's result showing that there is an interaction for the number of hints needed between LF versus NLF and the number of possible targets for action on a screen. The difficulty of

NLF steps increases dramatically as a function of the number of targets.

Comparison With LICAI+'s Predictions

Training Performance

LICAI+ predicts perfect performance for both training and recall trials at all delays for LF steps. If we use the proportion of users requiring hints as our measure, a large majority of Franzke's (1994) results (shown in Table 2) and Soto's (1997) findings (shown in Table 3) support this prediction. The largest deviation that we know of is in the data from LF/U, Soto's condition where 19% of the participants required hints on the training trial.

The model makes equally strong training performance predictions for tasks and steps that do not support the label-following strategy (NLF tasks). LICAI+ predicts that these tasks and steps cannot be learned by exploration without hints or information looked up in a manual or help system. However, this prediction for NLF tasks is not sound. The observed proportions of tasks or steps requiring at least one hint ranges from less than .5 to .9 in different conditions of the Franzke and the Soto data.

However, the pattern of deviations in both the Franzke and the Soto data is instructive and supports the claim that the LF-NLF distinction is a useful design heuristic. LICAI+ makes incorrect predictions for learning by exploration in NLF tasks because of the model's simple exploration process. First, the model cannot perform exploratory activities like pulling down a menu to see if any items on that menu link to the tasks goal. Experienced Macintosh users carefully explore menus (Rieman, 1996) and act upon matching labels uncovered during such explorations.

Second, users seem to be able to use elimination strategies when dealing with a small number of screen objects like the items on a menu. For example, when participants are given the hint to pull down the **Options** menu in the hide legend task, they correctly select **Show Graph Items...** by a process of elimination. The other items on this menu are more specific and clearly have nothing to do with the hide legend task. LICAI+ can perform this step if it is given the knowledge that 'show is the opposite of hide' and that 'the legend is a graph item.'

The above arguments suggest that an interesting test of the model would be to consider NLF tasks in which the first two steps violate the label-following strategy. 'Hide legend' is such a task. Rodriguez (1997) shows that 100% of his subjects required hints to be able to perform this task. Franzke (1994) found that approximately 90% of the participants required hints for steps where there was no link between the task goal and the correct object's label.

Recall at Short Delays for NLF Tasks

LICAI+ predicts that successful performance on recall trials is possible only when users retrieve a hint or an encoding of a step from episodic memory. However, the model does not make predictions about the effects of delay. We have assumed that LICAI+'s recall predictions apply to delays of one or more days.

Franzke's (1994) and Soto's (1997) results show that immediate recall of NLF steps is quite good. Franzke (1994) found that about 90% of NLF steps can be recalled after a 5 minute delay (see Table 2). About 75% of Soto's PL tasks were performed correctly, without a hint, after a short delay (See Table 3).

Recall at Long Delays for NLF Tasks and Steps

LICAI+ predicts that successful recall performance can vary from .722, to .205 as a function of the encoding bias for NLF tasks and steps. Franzke's and Soto's results at long delays are hard to interpret because of the results from training trials for NLF tasks. Users' learning by exploration is better than that predicted by LICAI+. Thus, contrary to the predictions of the model, users will be able to discover the correct action on a recall trial even if they fail to recall a hint or encoding of the step.

We reanalyzed both Franzke's no link and inference steps at the long delay shown in Table 2 and Soto's recall data from his PL conditions shown in Table 3 at the long delay. We made the assumption that the probability of requiring hints on recall trials, $P_{require\_hint}$, is just the probability of failing to recall a hint or step encoding, P_{fail_recall} , times the probability of failing to discover the correct action by exploration, $P_{fail\_exploration}$, assuming that the two events are independent. If we assume that $P_{fail\_exploration}$ estimated by the probability of requiring hints on the training trial, P_{fail_recall} can be estimated by $P_{fail\_recall} = P_{require\_hint}/P_{fail\_exploration}$.

The estimated values of P_{fail_recall} for Franzke's no link steps is .68, and .69 for the inference steps. These values are close to the predicted value for the TG condition shown in Table 1.

The estimated values of P_{fail_recall} for Soto's poorly labeled tasks at a long delay is .50 for the unique tasks and .55 for the common tasks. These results suggest that the task goal has a strong influence on the encoding process but that it is not as strong as the 4:1 bias assumed in computing the predictions for the TG conditions shown in Table 1.

CONCLUSIONS AND IMPLICATIONS FOR PRACTICE

We have asserted that most users are occasional users of many applications, and they routinely use only a small fraction of the functionality of their frequently used applications. A model of routine cognitive skill is not a good description of users' actual patterns of use. The action sequences for occasionally performed tasks are generated by a mixture of recall of previous episodes of use and of problem solving processes that attempt to reconstruct missing action knowledge. Performance of these tasks is more like the reconstructive processes involved in recalling a story rather than the execution of a rule-based representation of a routine cognitive skill.

LICAI+ is a model of occasional users. This model suggests the partitioning of all steps executed in performing a task into two categories: steps that support the label-following strategy and those that do not. Steps and tasks that support the label-following strategy can be performed by exploration. We know that users have strong preferences for learning by exploration (Carroll, 1990; Rieman, 1996), which the label-following strategy supports.

Experienced users can make effective use of manuals (Rieman, 1996) to perform tasks that are not supported by the label-following strategy. However, users will have continued trouble with steps not supported by label following (NLF steps). These steps once correctly performed with the assistance of hints are difficult to remember over long delays (2 or more days). We estimate that the probability of recall failure is at least .5.

The data from the short delay recall conditions also suggests a possible limitation of empirical usability tests. Test users will have trouble with the initial versions of common tasks that don't support the label-following strategy. Second and third versions of these tasks that are given to test-takers later in a session will be performed correctly, and evaluators may incorrectly infer that there are no problems with the interface for these later versions.

In summary, the theoretical and empirical results presented in this paper and in numerous other studies demonstrate the wide applicability of the label-following strategy. It supports rapid learning of all kinds of applications, not just walk-up-and-use applications like automated teller machines. We have shown in this paper that label following is also a major contributor to the usability of occasionally performed tasks.

REFERENCES

Bewley, W.L., Roberts, T.L., Schroit, D., & Verplank, W.L. (1983). *Human Factors Testing in the Design of Xerox's 8010 'Star' Office Workstation: Case Study D: The Star, the Lisa, and the Macintosh.*

Carroll, J.M. (1990). *The Nuremberg funnel: Designing minimalist instruction for practical computer skills.* Cambridge, MA: MIT Press.

Franzke, M. (1994). *Exploration, acquisition, and retention of skill with display-based systems.* Unpublished Dissertation, University of Colorado, Boulder, Department of Psychology.

Franzke, M. (1995). Turning research into practice: Characteristics of display-based interaction. *Proceedings of human factors in computing systems CHI '95* (pp. 421–428). New York: ACM.

Kintsch, W. (1988). The role of knowledge in discourse comprehension: A construction-integration model. *Psychological Review*, **95**, 163–182.

Kintsch, W. (in press). *Comprehension: A paradigm for cognition.* Cambridge University Press.

Kintsch, W., & Welsch, D.M. (1991). The construction-integration model: A framework for studying memory for text. In W. E. Hockley and S. Lewan-dowsky (Eds.), *Relating theory and data: Essays on human memory* (pp. 367–385). Hillsdale, NJ: Erlbaum.

Kitajima, M., & Polson, P.G. (1995). A comprehension-based model of correct performance and errors in skilled, display-based human-computer interaction. *International Journal of Human-Computer Studies,* **43**, 65–99.

Kitajima, M., & Polson, P.G. (1996). A comprehension-based model of exploration. *Proceedings of human factors in computing systems CHI '96* (pp. 324–331). New York: ACM.

Kitajima, M., & Polson, P.G. (1997). A comprehension-based model of exploration. *Human-Computer Interaction*, **12**, 345–389.

Polson, P.G., & Lewis, C. (1990). Theory-based design for easily learned interfaces. *Human-Computer Interaction*, **5**, 191–220.

Rickard, T.C. (1997). Bending the power law: A CMPL theory of strategy shifts and the automatization of cognitive skills. *Journal of Experimental Psychology: General,* **126**, 288–311.

Rieman, J. (1996). A field study of exploratory learning strategies. *ACM Transactions on Computer-Human Interaction,* **3**, 189–218.

Rieman, J., Young, R.M., & Howes, A. (1996). A dual space model of iteratively deepening exploratory learning. *International Journal of Human-Computer Studies,* **44**, 743–775.

Rodriguez, M. (1997). A detailed analysis of exploratory behavior of new users of a graphics application. Institute of Cognitive Science Technical Report. University of Colorado, Boulder.

Ross, B. (1984). Remindings and their effects in learning a cognitive skill. *Cognitive Psychology,* **16**, 371–416.

Santhanam, R., & Wiedenbeck, S. (1993). Neither novice nor expert: the discretionary user of software. *International Journal of Man-Machine Studies,* **38**, 201–229.

Smith, D.C., Irby, C., Kimball, R., Verplank, W.L., & Harslem, E. (1982). *Designing the Star User Interface: Case Study D: The Star, the Lisa, and the Macintosh.*

Soto, R. (1997). Properties of Tasks that Determine Success in Learning By Exploration and Recall. Unpublished Master Theses.

Terwilliger, R.B., & Polson, P.G. (1997). Relationships between users' and interfaces' task representations. Proceedings of human factors in computing systems CHI '97. New York: ACM.

Wolfe, M., & Kintsch, W. (submitted). An overview of the construction-integration model.

Modeling Human Error within a Cognitive Theoretical Framework

Daniela K. Busse and Chris W. Johnson
Dept. of Computing Science, University of Glasgow
17, Lilybank Gardens, Glasgow G12 8RZ
Tel: +44 141 339 8855 ext. 2918
Email: [bussedljohnson]@dcs.gla.ac.uk

ABSTRACT

Current cognitive user models enable interface designers to describe, analyze and predict aspects of user cognition. However, none of the major cognitive user models such as ICS, MHP, or CCT tackle the human error aspect of cognition explicitly. The represented operator performance is constrained to be error-free, expert performance. This paper argues that usability and design analysis will greatly benefit from representing a cognition-based error model within a cognitive architecture, such as ICS. The Netscape Internet browser acts as a case study throughout. The resulting approach is shown to aid the analysis of human error. Reasoning about potential error causes as well as the generation of design recommendations can thus be grounded in cognitive theory.

Keywords

Human Error, Netscape, Cognitive User Modeling, ICS

INTRODUCTION

Integrating Error Models and Cognitive Architectures

Cognitive architectures seek to represent the building blocks of human cognition. They provide the basis for cognitive user models, which strive to represent some aspects of the user's understanding, knowledge, or cognitive processing. These models can then contribute to our understanding of the cognitive limitations of an operator performing a task, for example the effects of cognitive load on user performance (Barnard and May, 1993; Ashcraft, 1994).

Erroneous task performance highlights precisely these limitations of human cognition. It is surprising, therefore, that the major cognitive user models do not explicitly tackle issues associated with erroneous performance based on cognition. They strive to represent error-free performance, assuming expert performance in some perfect context (see for instance Simon, 1988; Grant and Mayes, 1991; Booth, 1991). This idealizes real-life conditions of task performance.

User error can point to problems in human-system interaction that need to be resolved in order to enhance the system's usability. Human error taxonomies aid the prediction and detection of error classes. They can thus be exploited for error prevention and recovery mechanisms (Reason, 1990; Taylor, 1988). Those can then be incorporated into the interface design.

On the other hand, stand-alone human error theories highlight possible sources of erroneous performance without providing a language in which to express these error tendencies when applied to human cognitive task performance. This paper will use a cognitive architecture as a vehicle for expressing not only expert task performance but also the more realistic error-prone thought and action sequences processed by the human operator. By doing this, the error modeling capability implicit in the comprehensive ICS cognitive architecture is made the focus of inquiry into the underlying cognition of user performance. Such explicit modeling of erroneous performance can thus help to communicate user cognition analyses, and to ground design decisions in a cognitive theoretical framework.

As a running example, error modeling will be applied to tasks concerning the use of Netscape Navigator™. This example is appropriate because it represents a mass-market application where errors frequently lead to high levels of frustration during common tasks (Johnson, C., 1997).

Interacting Cognitive Subsystems (ICS) and Reason's Model of Human Error

We will use Interacting Cognitive Subsystems (ICS) (Barnard and May, 1993) to illustrate the modeling of human error within a cognitive architecture. ICS provides a comprehensive account of human cognition. It has proved powerful in explaining cognitive phenomena such as the stability of users' mental models during dual task interference effects (Duke, et al. 1995). It has been applied to real-life systems and tasks, such as cinematography (May and Barnard, 1995). Alternative cognitive user models, such as Task Analysis for Knowledge based Descriptions (TAKD) (Johnson, P. et al., 1994), User Action Notation (UAN) (Hartson et al., 1990), or Soar (Newell, 1990) might have been used. However, they lack the level of detail in ICS's representation of cognitive processes, or, in the case of Soar, the inherent constraints these have to satisfy (Wilson et al., 1988; Kjaer-Hansen, 1995). ICS was designed to provide a theoretical framework within which to place user cognition. It attempts to "satisfy the need for applicable theory" (Barnard, 1987). ICS, therefore, bridges the gap between theory-oriented cognitive architectures and task-oriented cognitive user models (Grant and Mayes, 1991; Simon, 1988).

Reason's taxonomy of human error (Reason, 1990) represents a conceptual classification of error, as opposed to a contextual or a behavioural one. The latter, exemplified for instance by Hollnagel's (1991) classification of error phenotypes, does not lend itself to the in-depth analysis of the underlying cognitive sources of error. For instance, a behavioural error category might include errors that exhibit the same surface characteristics without sharing the same cognitive basis.

An Interactive System: Netscape Navigator

According to user population estimates, the Internet is gaining roughly 150,000 new users per month, joining 20 million existing Internet users (Pitkow and Recker, 1994). Internet browsers facilitate global communication by providing supporting hypertext navigation. Familiarity with such browsers, and therefore their usability constitutes a prerequisite for taking part in this novel information exchange. Maximizing this usability therefore represents a continuous concern for designers of successively modified versions of Internet browsers. The Netscape Interface (see Figure 1) will be used for illustration throughout this paper.

Content and Structure of this Paper

The following section will take a closer look at the ICS architecture and Reason's theory of human error. The modeling capacities of ICS will be illustrated by a representation of an error-free user performance. Reason's error classification scheme will then be introduced. Readers familiar with ICS and Reason can move straight to the third section, where the benefits of this combined modeling approach are pointed out. ICS is used as a framework within which Reason's classification of human error can be expressed.

A COGNITIVE ARCHITECTURE AND A HUMAN ERROR MODEL

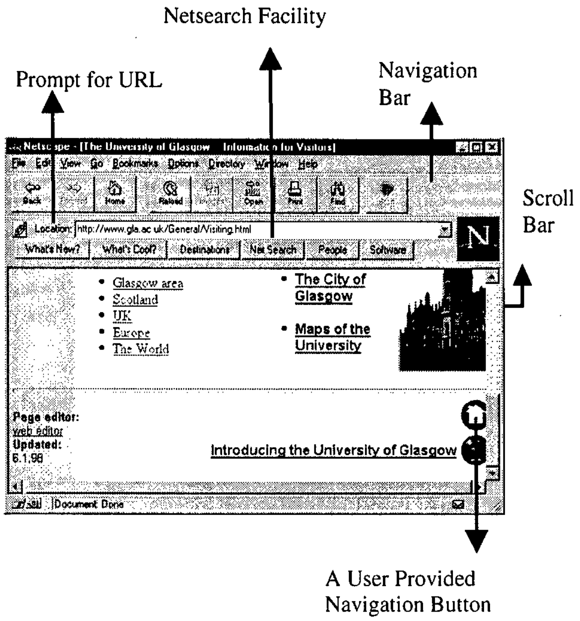

Netsearch Facility
Prompt for URL
Navigation Bar
Scroll Bar
A User Provided Navigation Button
Netscape - [The University of Glasgow Information for Visitors]
File Edit View Go Bookmarks Options Directory Window Help
Back Home Reload Open Print Find
Location: http://www.gla.ac.uk/General/Visiting.html
What's New? What's Cool? Destinations Net Search People Software
Glasgow area
Scotland
UK
Europe
The World
The City of Glasgow
Maps of the University
Page editor:
web editor
Updated:
6.1.98
Introducing the University of Glasgow
Document Done

Figure 1. The Netscape Internet Browser

This section describes Barnard's ICS model and Reason's human error taxonomy. This provides the framework in which the representation of erroneous operator interaction can be placed.

Interactive Cognitive Subsystems (ICS)

Cognition is represented in ICS as the flow of information between a number of different subsystems, and the processing performed on this data. Each of the subsystems has associated with it a unique mental code in which it represents the information it receives and processes. It will transform its data output into the corresponding mental code of the subsequently receiving subsystems. Each subsystem can receive several input streams and achieve a blending of these data streams under certain circumstances as described below (May and Barnard, 1995). Each subsystem also has at its disposal a local image store. This serves as an episodic memory buffer of infinite size. A copy of any input the subsystem receives will automatically be copied to the local image store, before being further processed.

The nine subsystems can be grouped into four categories. Figure 2 presents an overview.

Modeling a Netscape Task in ICS

Figure 3 illustrates how the error-free performance of a task of locating an object (an Up-Arrow, such as shown in the visual subsystem) is modeled in ICS in terms of

Sensory subsystems:	
VIS	visual: hue, contour etc. from the eyes
AC	acoustic: pitch, rhythm etc. from the ears
BS	body-state: proprioceptive feedback
Effector subsystems:	
ART	articulatory: subvocal rehearsal & speech
LIM	limb: motion of limbs, eyes etc.
Structural subsystems:	
OBJ	object: mental imagery, shapes etc.
MPL	morphonolexical: words, lexical forms
Meaning subsystems:	
PROP	propositional: semantic relationships
IMPLIC	implicational: holistic meaning

Figure 2. The Cognitive Subsystems

information flow between the subsystems, and thus the different resources that are employed. Visual information concerning the target arrives at the visual subsystem and is copied into the local store. It is then transformed into object code (1). The propositional subsystem has generated a representation of the target of the location task (by conferring with its local buffer) and transforms this into object code (2). This is sent to the object subsystem, and can there be blended with the incoming structurally encoded visual information (3). The matching representation can be sent back to the propositional subsystem – the target has been located.

Thus, Figure 3 illustrates how human mental processing underlying error-free performance can be represented within ICS. In the case of erroneous performance, however, usability designers might resort to an error classification scheme in order to analyse this particular instance of user behaviour. The following section will introduce one such taxonomy. We will then go on to show how a more detailed, cognitive analysis can be based on initial error classification, and thus provide a further perspective on user behaviour.

Reason's Classification of Human Error

Reason (1990) investigated the more general underlying error production mechanisms within human cognition and produced a conceptual classification of error types which is widely referred to in research into error modeling (Green, 1985, Rasmussen, 1983; Rouse and Morris, 1987; De Keyser, 1989). He bases his error classification skill-based slips and lapses on the one hand, and rule- and knowledge- based mistakes on the other (see also Norman, 1981, and Rasmussen, 1983).

Reason furthermore asserts that instances of his three basic error types are indirect results of what he calls the 'underspecification' of cognitive operations. In case of an ambiguity of the situational requirements, the cognitive system defaults to contextually appropriate, high frequency responses. This idea of default assignments features in most other cognitive theories, such as Bartlett's (1932) theory of schemata, and is well backed up by empirical evidence.

This scenario particularly lends itself to being expressed in the 'cognitive language' provided by ICS. The limitations of human cognition in the face of information overload, or cognitive strain, is built into ICS as the architectural constraint of subsystems not being able to process simultaneously inputs which belong to distinct configurations. Using ICS might help expressing the details of Reason's 'underspecification' more precisely.

Skill-based Slips and Lapses

Slips and lapses are error types that these manifest themselves as actions or states that deviate from the current intention due to execution failures (slips) and/or storage failures (lapses). Slips and lapses are observed at the skill-based level of performance, and originate from either the omission of attentional checks (inattention) during the routine action sequence or making an attentional check at an inappropriate moment (overattention).

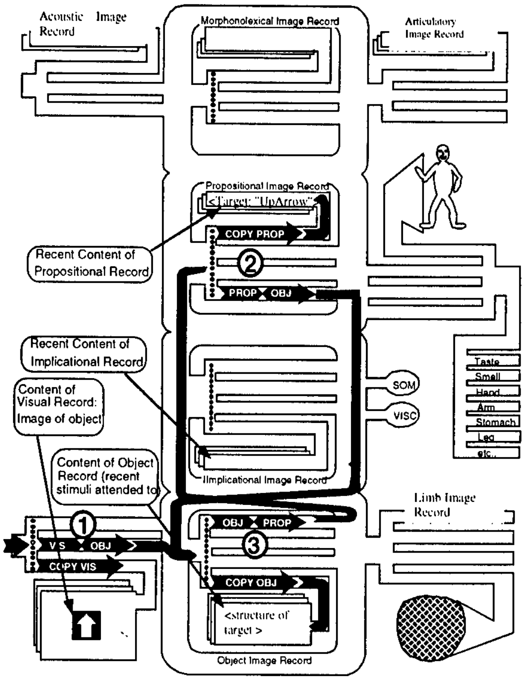

Acoustic Image Record
Morphonolexical Image Record
Articulatory Image Record
Propositional Image Record
<Target: "UpArrow">
COPY PROP
Recent Content of Propositional Record
2
PROP
OBJ
Recent Content of Implicational Record
Content of Visual Record: Image of object
SOM
VISC
Taste
Smell
Hand
Arm
Stomach
Leg
etc.
Content of Object Record (recent stimuli attended to
Implicational Image Record
Limb Image Record
1
OBJ
PROP
3
V S
OBJ
COPY VIS
COPY OBJ
<structure of target >
Object Image Record

Figure 3. Processing associated with the task of locating an icon on Netscape

A *slip* caused by inattention occurs in particular when current intention is to deviate from common practice. For instance, entering a well-known URL of a website constitutes a routine task. If the URL is changed and the user, although aware of that change, still happens to enter the old URL, then this is a typical example of an action slip.

A *lapse* might arise from what Reason calls 'Reduced Intentionality'. For instance, if selecting a link on the current site results in a considerable delay for this site to be loaded, the users might become distracted, and then experience disorientation upon facing the loading site. This can be seen as one of Reason's described reduced intentionality states, such as a 'what-am-I-doing-here' experience (see below).

Skill based errors such as these contribute to the sources of user frustration when accessing the World Wide Web (as described in more detail in Johnson, C., 1997). These errors need to be taken into account in future design decisions. Applying Reason's categorization of error helps to identify error classes and presents a step towards dealing with the underlying usability problems of the system.

However, error taxonomies such as Reason's typically confine themselves to broad error categories such as slips and lapses. A more detailed, lower level description of such classes might aid the further investigation of its instances. Thus, the design process might be tuned more finely to the usability needs pointed to by the user error.

Cognitive modeling techniques such as ICS can provide a more precise vocabulary to augment the general descriptions of error taxonomies. Examples of this lower level modeling of classes of human error are given below.

Rule-based Mistakes

Mistakes are apparent in actions that may run according to plan, but where the plan is inadequate to achieve its desired outcome. For any task, rules must be selected by the cognitive system which describes methods to reach a given (sub)goal. The selection occurs according to certain criteria. These include best match, specificity, and rule strength. Rule strength is defined to be the number of times a rule has performed successfully in the past. Occasionally, rule strength might override the other factors resulting in misapplications of otherwise 'good' rules to inappropriate situations.

As an example, an animated icon at the bottom of a page, near the contact information is quite often the mail-me icon (commonly found are self-folding envelopes, self-writing letters, or moving mailboxes). A corresponding rule will be formed and strengthened over several successful applications. In the case of a home-page icon being animated and located at a similar position in the screen layout, this rule might be applied and could lead to non-intended actions such as clicking on the icon when intending to mail the author of the page.

Such error classes can be predicted as increasingly adding to usability deficiencies as the use of animated icons accelerates in web page design (Nielsen, 1997). By being able to predict these errors, preventative measures can be taken and further user frustration (Johnson, C., 1997; Ramsay et al., 1998) can be curbed.

USING ICS TO EXPRESS REASON'S ERROR TYPES

In this section, we will examine more closely the modeling of errors as identified by Reason's taxonomy within the ICS architecture.

Commonly occurring errors and usability problems when interacting with Internet browsers' interfaces gave rise to numerous design guidelines and principles[1]. Interface design issues such as the use of counter-intuitive icons and download delays are all well known to aggravate usability problems (see for instance Nielsen, 1996; Johnson, C., 1997; Ramsay et al., 1998). Rarely, however, are the errors resulting from those usability problems described in detail, or even analyzed in terms of underlying psychological factors (Johnson, C., 1998). Expressing such errors within a cognitive model will allow us to investigate and reason about their underlying psychological causes. The model is thus used as a tool for reasoning about user error on a further, more detailed level.

Analysis of Errors and their Underlying Cognition

High download latency of web pages was identified as major source of frustration and decreased satisfaction with the downloading site and also as attenuating user performance (Ramsay, Barabesi and Preece, 1998; Johnson, C., 1997). For instance, as introduced above, if selecting a link on the current site results in a considerable delay for this site to be loaded, the users might become distracted, and then experience disorientation upon facing the loading site.

This disorientation can be classed as the effect of a phenomenon which Reason termed 'Reduced Intentionality'. If a delay occurs between the formulation of an intention to do something and the time for this activity to be executed, the intention needs to be periodically refreshed. Other cognitive processes such as secondary intentions will otherwise claim the workspace resources. This mechanism can lead to lapses in the form of reduced intentionality states, the above described surprise and disorientation.

The cognitive processes underlying this scenario can be represented in ICS as shown in Figure 4.

[1] See for instance Yale C/AIM WWW Style Manual (URL: "http://info.med.yale.edu/caim/manual/index.html" current at 08.12.1997) or The Ten Commandments of HTML (URL:"http://www.visdesigns.com/design/commandments.html" current at 08.12.1997

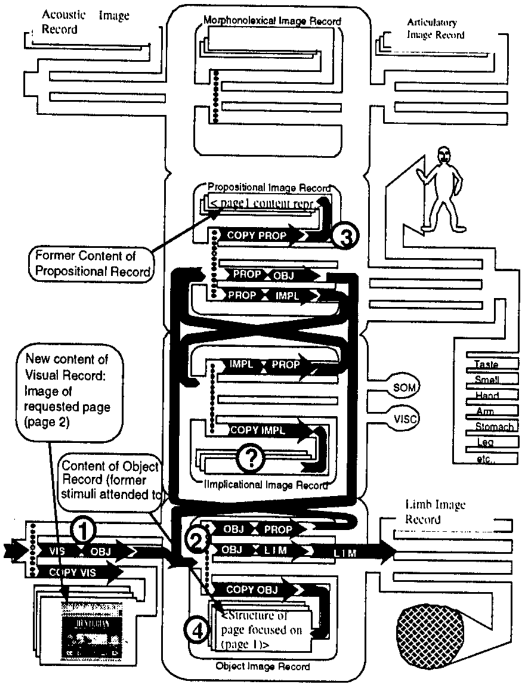

Acoustic Image Record
Morphonolexical Image Record
Articulatory Image Record
Propositional Image Record
< page1 content repr.
COPY PROP
3
Former Content of Propositional Record
PROP OBJ
PROP IMPL
New content of Visual Record: Image of requested page (page 2)
IMPL PROP
SOM
VISC
Taste
Smell
Hand
Arm
Stomach
Leg
etc.
COPY IMPL
?
Implicational Image Record
Content of Object Record (former stimuli attended to)
Limb Image Record
1
2
OBJ PROP
VIS OBJ
OBJ LIM
LIM
COPY VIS
COPY OBJ
<Structure of page focused on (page 1)>
4
Object Image Record

Figure 4. Reduced Intentionality: A Lapse

After processing the goal hierarchy for selecting a link, the cognitive system shifts its focus back onto the current page (3 and 4). If novel external (1) and the current internal input are not coherent, and thus cannot be blended (2), a decision must be made as to which of those to accept as valid input. The longer the delay, the stronger the influence of the novel input grows, with it eventually replacing the internal propositionally influenced representation (3). The recognition of this mismatch will lead to a lapse as described above.

By modeling the underlying mechanisms of manifestations of attenuated performance, such as user error, and the causes of decreased satisfaction within ICS we can shed some light on the processes fundamental to the production of the user error as mediated by the described usability problems.

Reasoning about Alternative Analyses of Error Causes

Misinterpreting user interface icons is a common source for user error in interactive systems (Norman, 1988, 1993). However, the mistake might be grounded in varying cognitive processes, and not stem from one kind of cognitive mechanism alone.

Typically, user interface design manuals and textbooks stress the importance of intuitiveness of the icons chosen (Preece, 1994) and thus identify 'counter-intuitiveness' as a source of faulty identification of icons. However, further insight into the source of such user error can be obtained by investigating it in greater detail. As will be shown below, mistaking for instance a mail-me button with a homepage icon can be modeled in respect to two differing underlying cognitive mechanisms.

Unless these two different causes are considered these designs might misdiagnose an important problem in user utilization of icons. Using a cognitive architecture to reason about the potential underlying cognitive error production processes allows designers to investigate the detected usability problem in a systematic way.

The above described user error could according to Reason's scheme be classified as a slip termed 'Perceptual Confusion'. In perceptual confusion, something that looks like the proper object, is in the expected location, or does a similar job is accepted as a match for the proper object. These slips could arise because, in a routine set of actions, it is unnecessary to invest the same amount of attention in the matching process. Thus acceptance criteria concerning the expected input might degrade, and result in rough and ready matches.

The processing carried out can be modeled in ICS as shown in Figure 5.

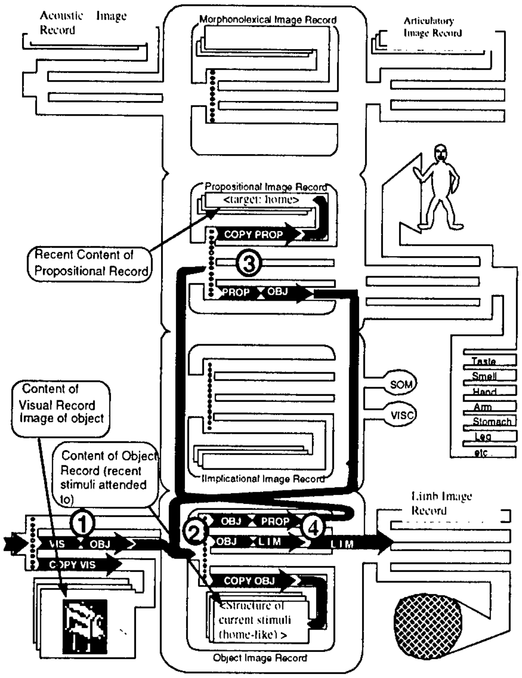

Acoustic Image Record
Morphonolexical Image Record
Articulatory Image Record
Propositional Image Record
<target: home>
COPY PROP
Recent Content of Propositional Record
3
PROP OBJ
SOM
VISC
Taste
Smell
Hand
Arm
Stomach
Leg
etc.
Content of Visual Record Image of object
Content of Object Record (recent stimuli attended to)
Implicational Image Record
Limb Image Record
1
2
OBJ PROP
4
VIS OBJ
OBJ LIM
LIM
COPY VIS
COPY OBJ
<Structure of current stimuli (home-like) >
Object Image Record

Figure 5. Perceptual Confusion: A Slip

The visual data is received at the visual subsystem (1), sent to the object subsystem for the recovery of a structural description (2), and finally interpreted by the propositional subsystem (3). A loop is entered in order to maintain a stable cognition. The resulting interpretation on the propositional level influences the further view of the object. If, however, the object subsystem receives ambiguous visual information, it will make use of its local image record and fill in the assumed missing information. This principle of ICS resembles closely what Reason describes as the cognitive system's reaction to underspecification of a mental operation as described above.

The data thus acquired from the image record of the object subsystem might also fit in with the propositional interpretation of what is perceived, and thus stabilize in the cognitive system. If the assumption underlying the choice of what data is used to eliminate the underspecification is wrong, however, the representation of what is thought to be perceived will also be incorrect. The wrong icon will be chosen, and the information necessary for a mouse click sent to limb subsystem (4).

This represents one possible underlying cause of the described error. However, the same manifestation of user behaviour might also point towards a second, different underlying cognitive mechanism. Employing Reason's taxonomy, the mistaking of an icon can be classed as a perceptual slip as modeled above. On the other hand, it could also be classed as a rule based mistake. Using ICS to model the underlying cognition of the error provides a means to further investigate the behaviour trace and its associated usability problem.

Thus, the error described above could be classed as a rule-based mistake as opposed to a slip. Identifying the home-icon might well be based on rules that are utilized by the cognitive system in order to discriminate different sets of icons. Features which positively discriminate icons fulfilling one function from those fulfilling another might be listed in the set of conditions which when matched cause to fire the rule. Indiscriminative features in icons might thus lead to a rule wrongly being fired.

This can be modeled in ICS (see Figure 6) similar to the modeling approach applied to the perceptual confusion approach, but this time with the implicational subsystem playing the major role in accepting information augmented wrongly by the propositional subsystem and its local image store. Thus for the goal 'press home button', a subgoal hierarchy can be formulated as 'if locate home button, move cursor to click on it', and 'if object has X features, it is the home button'. By mistaking the icons on a propositional level, the mail-me button might be clicked instead.

The examples elaborated above show clearly how one overt form of user error can stem from several different 'errors' within the cognitive processing taking place. This M:N relationship between cause and error might have gone undetected if systematic error modeling within a cognitive architecture had not taken place, this helps analysts to explicitly consider the detailed causes of usability problems.

Generating Design Recommendations

Since underspecification proved the major source of error in the above example, once for perceptually and then for semantically discriminative features of the icon, this should be targeted by designers to remedy misidentification of icons. Thus, two functionally dissociated sets of icons should not share the same superficial perceptual features.

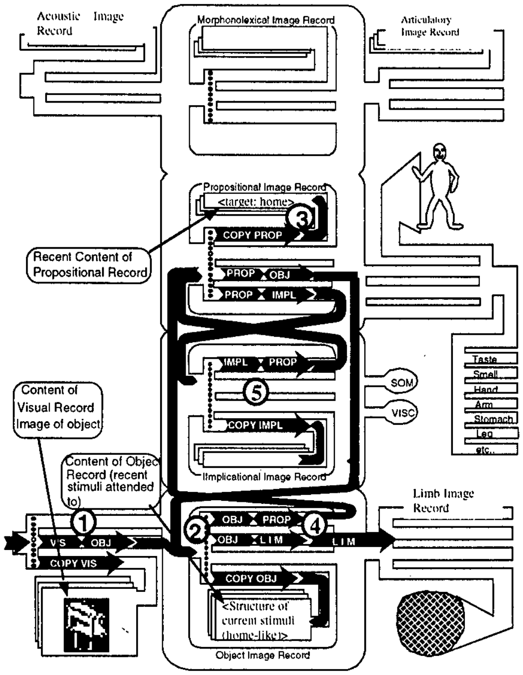

Acoustic Image Record
Morphonolexical Image Record
Articulatory Image Record
Propositional Image Record
<target: home>
3
COPY PROP
Recent Content of Propositional Record
PROP OBJ
PROP IMPL
IMPL PROP
5
SOM
VISC
Taste
Smell
Hand
Arm
Stomach
Leg
etc..
Content of Visual Record Image of object
COPY IMPL
Content of Objec Record (recent stimuli attended to)
Implicational Image Record
Limb Image Record
1
V S OBJ
COPY VIS
2
OBJ PROP
4
OBJ LIM
LIM
COPY OBJ
<Structure of current stimuli (home-like)>
Object Image Record

Figure 6. Rule Strength: A Mistake

Features commonly used to discriminate one set of icons from another should be taken into account when designing future sets (Moyes, 1995). These feature considerations should not limit themselves to ambiguity concerning structural characteristics of icons, but also to features such as those mentioned in the examples earlier. This included as discriminative features of mail-me buttons not only their shape and internal composition, but also for instance the location of the icon on the screen, and characteristics commonly unique to mail-me buttons such as animation as present in self-folding envelopes, self-writing letters, or moving mailboxes.

The important point to highlight here is that the modeling approach described does present a method for providing a grounded rationale for design decisions, and can guide the designer in making informed choices when faced with design alternatives.

Another example of how this modeling technique can aid the generation of design decisions is introduced as this section progresses.

Johnson (C., 1997) describes how download latency of web pages affects the usability of the World Wide Web. The effects range from user dissatisfaction with time investment to the psychological devaluation of the anticipated page (Ramsay et al., 1998). Consider the following scenario of user error resulting from download latency: After having selecting a link on the current site, a delay in downloading might lead to attention being focused on reading the current page. An intention to scroll down the page just before the new page is downloaded might lead to the scrolling action being carried out on the new page instead.

This scenario fits Reason's description of 'behavioural spoonerisms', namely slips based on interference errors. As defined above, a slip is an action that deviates from intention due to failure in the execution stage of processing operations. An interference error occurs, when two concurrent actions compete for control over cognitive processing and a transposition of actions within the same sequence takes place. For instance, intending to speak and perform an action at the same time can lead to inappropriate blends of speech and action. In our example, waiting for the new page to load, and scrolling the old page can be seen as two concurrent actions interfering and leading to an execution failure, the scrolling of the new page.

This can be modeled in ICS very similarly to the skill-based example of reduced intentionality. Only this time the focus is not on the delay but on the shift of focus back to the current page. A 'mental model' of the current page will be constructed (or reactivated). The unexpected appearance of the new page might lead to a blending of representation and the action included in one cognitive configuration carried out as part of a secondary one.

As a consequence, future browser designers should beware of the error-inducing character of non-interrupted browser functionality when downloading a site. Alternatively, browser functionality should only be available to the current site accessed. A clear distinction should be made when transferring functionality to the downloading site to alert users to the new context. This design flaw in Internet Browsers has not received much attention. We hypothesize that it may become increasingly important as the interweaving of the user population of the Internet grows and the World Wide Web becomes an increasingly common tool for communication and information exchange. Detailed, error-oriented cognitive analysis of such design problems can help to predict future generations of interface problems.

CONCLUSION AND FURTHER WORK

Cognitive user modeling enables engineers to gain a deeper understanding of the complexities of human task performance. Current techniques typically constrain this performance to be idealized, error-free and often at an expert level. However, human error during performance represents a major source of insights into the workings and limitations of operator cognition, and therefore into usability problems. By being based on cognitive models, the possibility of representing erroneous performance is inherent in these techniques. Few modeling techniques to date explicitly represent human error precisely, as embedded in cognitive theory. This paper showed the adoption of Reason's error taxonomy and Barnard's ICS for the systematic representation of operator error within a theoretical cognitive framework. The utilization of such a combined approach was illustrated to benefit several areas of application. User error can be described more precisely by linking it to its underlying cognition. Analysis can reach beyond surface categorization, and it is made possible to reason about the actual causes of error. As a consequence, an informed choice concerning competing design options is facilitated. This paves the way for usability design that takes full advantage of the insights expressed in cognitive theory.

Embedding human error modeling into a cognitive theoretical framework helps to express designers' understanding of the error sources. Communication of their reasoning, based on expertise and experience, is illustrated in this paper by using Reason's taxonomy and ICS. Further work might also take issues such as 'learnability' and level of complexity into account in the choice of the cognitive architecture employed. More easily learnable cognitive modeling techniques will further lend themselves for integration into the design process.

ACKNOWLEDGEMENTS

This work was supported by UK EPSRC Grant No GR/L27800.

REFERENCES

Ashcraft, M.H., *Human Memory and Cognition*, 2^{nd} ed., Harper Collins College Publishers, 1994

Barnard, P.J., Cognitive Resources and the Learning of Human-Computer Dialogs. In: Carroll, J.M., (ed.) *Interfacing Thought: Cognitive Aspects of Human-Computer Interaction*, ch.6. MIT Press, Cambridge, MA, 1987

Barnard, P.J., May, J., Cognitive Modeling for User Requirements. In Byerley, P.F., Barnard, P.J., & May, J. (eds) *Computers, Communication and Usability* (North Holland Series in Telecommunication) Elsevier: Amsterdam (1993).

Bartlett, F.C., *Remembering: A Study in Experimental and Social Psychology.* Cambridge: Cambridge University

Press, 1932

Booth, P.A., Modeling the User: User-System Errors and Predictive Grammars. In Weir G.R.S. and Alty, J.L. (eds.) *Human-Computer Interaction and Complex Systems*, Academic Press Ltd., 1991

De Keyser, V., Human Error, *Recherche*, 1989, Vol. 20, No. 216, p.1444

Duke, D.J., Barnard, P.J., Duce, D.A., and May, J., *Syndetic Modeling,* Amodeus-2 Technical Report ID/WP49, 1995

Grant, A.S. and Mayes,J.T., Cognitive Task Analysis? In Weir G.R.S. and Alty, J.L. (eds.) *Human-Computer Interaction and Complex Systems*, Academic Press Ltd., 1991

Green, R.G., Stress and Accidents, *Aviation Space and Environmental Medicine* 1985, Vol. 56, No. 7, pp.638 - 641

Hartson, H.R., Siochi, A.C., Hix, D. The UAN: A User-Oriented Representation for Direct Manipulation. *ACM Transactions on Information Systems*, 8(3), pp. 181-203, July 1990

Hollnagel, E., The Phenotype of Erroneous Actions: Implications for HCI Design. In: Weir, G.R.S. and Alty, J.L., (Eds.), *Human-Computer Interaction and Complex Systems*, Academic Press, 1991

Johnson, P., Diaper, D., and Long, J.B. Tasks, Skills, and Knowledge: Task Analysis for Knowledge based Descriptions. In: B. Shackel (ed.) *Human-Computer Interaction - INTERACT 1994.* Amsterdam: North Holland.

Johnson, C.W. Electronic Gridlock, Information Saturation and the Unpredictability of Information Retrieval Over the World Wide Web. In: Palanque, P. and Paterno, F. *Formal Methods in Human Computer Interaction: Comparison, Benefits, Open Questions.* Springer Verlag, Berlin, 1997

Johnson, C.W. *Why CHI (Computer-Human Interaction) Has Failed to Improve the Web*, at URL: "http://www.dcs.gla.ac.uk/~johnson/papers/web98.html" current at 02.02.1998

Kjaer-Hansen, J., Unitary Theories of Cognitive Architectures. In: Hoc, J., Cacciabue, P., Hollnagel, E., (Eds.), *Expertise and Technology: Cognition and Human-Computer Interaction.* LEA, Inc, 1995

May, J. and Barnard, P. (1995) Cinematography and Interface Design. In Nordby, K., Helmersen, P.H., Gilmore, D.J. and Arnesen, S.A. (eds.) *Human-Computer Interaction: Interact '95.* Chapman and Hall: London pp.26-31

Moyes, J., *Putting Icons in Context: The Influence of Contextual Information on The Usability of Icons,* PhD Thesis, Glasgow University, Computing Science Dept, 1995

Newell, A., *Unified Theories of Cognition.* Cambridge, MA: Harvard University Press, 1990

Newell, A. and Simon, H.A., *Human Problem Solving.* Englewood Cliffs, NJ: Prentic-Hall, 1972

Nielsen, J. Alert Box for May 1996: *Top Ten Mistakes in Web Design* at URL: "http://www.useit. com/alertbox /9605.html" current 08.12.1997

Norman, D.A. Categorization of action slips. *Psychological Review*, 1981, 88, 1-15.

Norman, D.A., *The Psychology of Everyday Things.* New York: Basic Books. 1988

Norman, D.A., *Things that make us smart*, Reading, MA: Addison-Wesley, 1993

Pitkow, J.E. and Recker, M.M., *Results from the first World-Wide Web User Survey*, 1994, at URL: " http://www.gvu.gatech.edu/user_surveys/survey-01-1994/survey-paper.html" current 02.12.1997

Preece, J., *Human-Computer Interaction*, Addison-Wesley, 1994

Ramsay, J., Barabesi, A., and Preece, J., A Psychological Investigation of Long Retrieval Times on the World Wide Web. In: *Interacting with Computers.* Special edition on CHI and Information Retrieval, 1998, in press

Rasmussen, J. Skills, rules, knowledge: signals, signs and symbols and other distinctions in human performance models. *IEEE Transactions: Systems, Man and Cybernetics*, 1983, SMC-13, 257-267.

Reason, J., *Human Error*, Cambridge University Press, 1990

Rouse, W.B., Morris, N.M., Conceptual Design of a Human Error Tolerant Interface For Complex Engineering Systems, *Automatica*, 1987, Vol. 23, No. 2, pp. 231-235

Simon, T., Analysing the Scope of Cognitive Models in HCI: A Trade-Off Approach. In: Jones, D.M. and Winder, R. (eds.) *People and Computers IV, Proceedings of the Fourth Conference of the British Computer Society.* Cambridge University Press, 1988

Taylor, J.R., Using Cognitive Models to make plants Safer: Experimental and Practical Studies. In: Goodstein, L.P., Andersen, H.B., and Olsen, S.E. (eds.) *Tasks, Errors and Mental Models,* Taylor and Francis, London, 1988

Wilson, M.D., Barnard, P.J., Green, T.R, and Maclean, A. Knowledge-Based Task Analysis for Human-Computer Systems. In: Van Der Veer, G., Green, T.R., Hoc, J., and Murray, D.M. (eds.) *Working with Computers: Theory versus Outcome* Academic Press, 1988

Automatic, action driven classification of user problem solving strategies by statistical and analytical methods: a comparative study

M. Fjeld, S. Schluep & M. Rauterberg
Institute of Hygiene and Applied Physiology (IHA)
Swiss Federal Institute of Technology (ETH)
Clausiusstr. 25, CH-8092 Zurich
{fjeld, schluep, rauterberg}@iha.bepr.ethz.ch
www.iha.bepr.ethz.ch/pages/forschung/mmi/mmi.htm

ABSTRACT

We have recorded the behaviour of several users solving the same tasks with an interactive database program and were able to identify several distinct strategies. Since the number of users exceeds the number of strategies, multiple users will have a strategy in common. Our aim was to find groups of users sharing the same strategy. Following each of the three methods (correlation, intersection, and exclusion) we define a metric among task solving sequences. For multiple users, we represent these measures by a matrix system, in order to find groups of users with common behaviour. Direct interpretation or multi dimensional scaling of such matrices indicates distinct user groups. The common denominator for each group can be interpreted as a strategy. A few distinctive solution strategies were found to exist.

Keywords

Mental models, observable behaviour, plan recognition, user strategies, statistical analysis, repetitive behaviour

1 MODELLING APPROACH

Humans express themselves in many ways. One of these ways is everyday problem solving. We will focus on problem solving in the domain of human computer interaction. In particular, we will examine how multiple users solve various tasks with a relational database application.

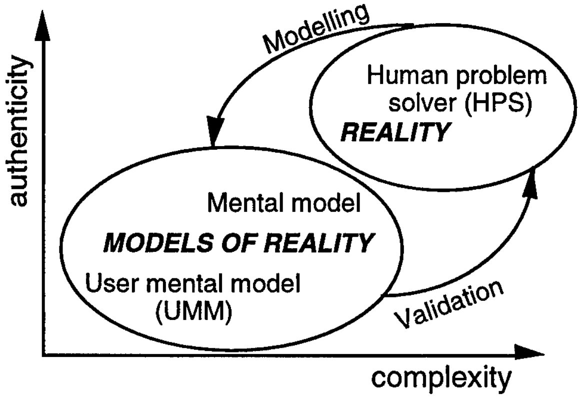


Fig. 1: A scheme showing the differences between models of reality and real humans (HPSs). Models are meant to represent objects and processes existing in reality.

It is hard to grasp how human problem solvers (HPS) really express themselves, since the persons we study are live beings. Nevertheless, a mental model (see Fig. 1) may give us an idea of the real HPS. Since we are interested in computer mediated, everyday task solving, we introduce a special case of mental models, called user mental model (Tauber, 1985) (UMM; see Fig. 1). UMMs can bring understanding about the strategies people use when solving specific problems. UMMs can be represented in many ways, using plain text, Petri nets or state-transition vectors. We choose the latter representation to elaborate UMMs based on observable task solving behaviour.

In general, we observe a lot of task solving behaviour that is not strictly *task related.* If we study one user solving a task, we are hardly able to single out the successful *strategy* from the *remaining behaviour.* One approach may be to study many users solving the same task. Since they all solve the same problem, we suppose that their common behaviour is what was required to solve the task. If there are several successful strategies, some users may have one strategy in common, other users a second one.

Successful strategies are most often defined by the given task-system combination. For users to accomplish a task, they must follow one of these strategies. As soon as a successful strategy has been accomplished, user behaviour is finished.

Which strategy a user prefers, as well as other kinds of user behaviour can tell us something about the particular HPS; for instance how the successful strategy was acquired. Given a behavioural task solving sequence, we want to separate the *strategy* (which is more related to the task-system combination) from the *remaining behaviour* (which is more related to the HPS). In the rest of this paper, *strategy* will mean one (of many), possibly error free, task solving behavioural sequences.

The aim of our work is to find which strategies are needed to solve a given task. We are looking for automatic methods to find these strategies. Under certain conditions, strategies may also be obtained by protocol analysis (Ericsson and Simon, 1984). Protocol analysis implies manual inspection of video and verbal utterances in addition to logfiles. With simple tasks, this work can be overcome. For more complex tasks, protocol analysis becomes cumbersome. Semi-automatic generation of process models was studied by Ritter and Larkin (1994). Motivated by their work, we wish to suggest further principles for automatic recognition of user strategies and plans.

In this paper, human perception and verbalisation will not be considered as part of the problem solving. Hence, purely based on observable task solving behaviour, we set out for automatic methods, applicable with simple as well as with complex tasks. We only consider protocol analysis as a mean to validate the automatic methods we elaborate.

2 SYSTEM DESCRIPTION

The system we study is a relational database program with 153 different dialogue states. The possible transitions of the system are represented by a state-transition vector space. A state-transition-vector (STV) summarises a subject's task solving behaviour for one task. It has length n, where n is the total number of transitions (n=978) for the complete database program. Each STV element tells how many times a certain transition was activated to solve the task.

Since the order of activated transitions is not contained in the STV, the order of user behaviour is only partly conserved. It is stored in an implicit form, given by the system dialogue structure and is embedded in the structure of the STV.

To reduce complexity, it is possible to replace each STV element >1, by 1. We call the result binary-state-transition-vector (B-STV). It tells us which transitions were activated, but nothing about repetition.

3 TASK DOMAIN

An empirical investigation was carried out to compare different types of expertise (Rauterberg, 1992). For the reconstruction of UMMs we used logfiles of six novice and six expert users, all solving the same task. The task was to find out how many data records there are in a given database consisting of three file. An example UMM of a task solving process, based on one of the experts, is presented in Rauterberg et al. (1997). In that example, 15 *different* transitions (number of positive STV elements) were activated to solve the task. However, since some of them were activated repeatedly, the *total* number of activated transitions (the sum of STV elements) is 25.

4 INTERPRETING BEHAVIOURAL SEQUENCES

Studying an STV of one user can tell us which system states the user passed by, which transitions that were triggered in those states and how many times that happened. Different users working with the same system are directly comparable, since their behavioural sequences only differ by the value of the vector elements.

5 BASIC QUESTIONS AND METHODOLOGY

First, we want to find out how the behavioural sequences of two users can be related. A classical method is that of correlation. An alternative is to look for analytical methods. The user STVs can be represented by ellipses as in Fig. 2. The area of an ellipse corresponds to the sum of the STV element values. Intersection area can be understood as symmetric similarity between two user STVs. Exclusion areas can be understood as the asymmetric difference between two user STVs.

Based on such considerations, we raise the following questions and suggest corresponding methods as answer:

1) What is the proximity between two behavioural sequences? Method suggested: *correlation.*

2) What do two behavioural sequences have in common (similarity)? Method suggested: *intersection* (Fig. 2).

3) What do two behavioural sequences not have in common (difference)? Method suggested: *exclusion* (Fig. 2).

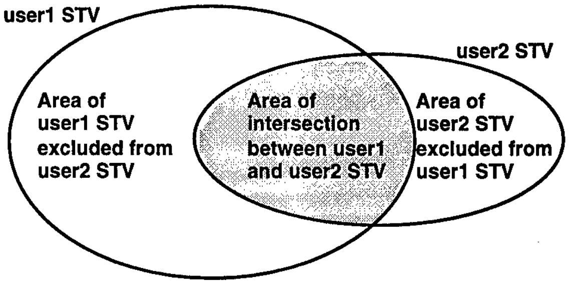

user1 STV
user2 STV
Area of user1 STV excluded from user2 STV
Area of intersection between user1 and user2 STV
Area of user2 STV excluded from user1 STV

Fig. 2: Intersection area and exclusion areas between user1 and user2 STV.

For each method, we elaborate a metric (Table 1). The order of the metric may be symmetrical (the metric applied from user1 STV to user2 STV is the same as the metric applied from user2 STV to user1 STV) or asymmetric (the metric applied from user1 STV to user2 STV is *not* the same as the metric applied from user2 STV to user1 STV). Based on the metrics applied between all the user STVs, we then apply a grouping algorithm.

With each group suggested by the grouping algorithm, a strategy may be approximated. The procedure is to create a STV with a maximum number of non-zero elements common to all the users of the group.

In the following presentation, we will proceed from more statistically based to more analytically based methods.

Table 1: The three suggested methods and their characteristics. CORR means a standard correlation method, the other metrics are defined by Formula 1,2 and 3.

Method	Metric name	Metric nature	Grouping algorithm
Correlation	CORR	Statistical	Statistical
Intersection	$M^{IS}_{p,q}$, $M^{BIS}_{p,q}$	Analytical	Statistical
Exclusion	$M^{EX}_{p,q}$	Analytical	Analytical

5.1 CORRELATION METHOD

In this method the metric between user STVs is the degree of proximity. The metric values are analysed by multi-dimensional-scaling (MDS, Systat, 1989) to indicate groups of users.

5.1.1 METRIC

Correlation is one way to measure the proximity between behavioural sequences. We apply Pearson correlation as a measure for proximity between two STVs. By this procedure, we get an mxm (m=12) diagonal dominant symmetrical matrix with possible values between minus one, via zero (no proximity) and one (equality). For Fig. 3 the observed values are between -0.003 and 0.948 (without considering the diagonal elements).

5.1.2 GROUPING ALGORITHM

The correlation matrix is interpret by MDS, giving the plot of Fig. 3. We have chosen to apply two dimensional MDS to allow visual interpretation of the plots.

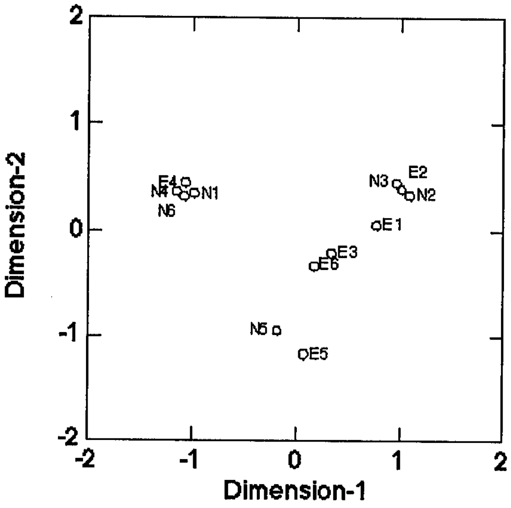


Fig. 3: MDS (r=1, Kruskal, Mono) plot with a Pearson correlation matrix gives RSQ=0.870.

5.1.3 OUTCOME

From the plot in Fig. 3 we see how the users may be grouped: {N1, N4, N6, E4}, {N2, N3, E1, E2, E3, E6} and {N5, E5}. Some of these user STVs may well consists of parts of several strategies in addition to the successful one.

According to the proportion of variance (RSQ=0.870), MDS explains some of the variance of the user data, but a significant part remains unexplained.

5.2 INTERSECTION METHOD

This method is based on the observation that if two users followed the same strategy, that strategy will belong to the intersection of the two users STVs. The order of the an intersection metric is symmetric, since both user STVs have the same in common. These metric values are analysed by MDS to indicate groups of users.

5.2.1 METRIC

Similar behaviour is measured by summing up the smaller STV elements of the two user STVs, thus considering the number of activated transitions common to both users.

It is reasonable to *normalise* the degree of intersection by the smaller of the sums of the STVs elements (which would be the maximum possible value for the intersection).

Formula 1:

$$M_{p,q}^{IS} = \frac{\sum_{i=1}^{n} \min\left(e_{p,i}, e_{q,i}\right)}{\min\left(\sum_{i=1}^{n} e_{p,i}, \sum_{i=1}^{n} e_{p,i}\right)}$$

where:

$M_{p,q}^{IS}$: Intersection metric between user p and q

i : Summing Index STV elements

n : STV length, upper summing limit

$e_{p,i}$: STV element i for user p

$e_{q,i}$: STV element i for user q

We may ignore repetitive behaviour, using B-STVs instead of STV. Results based on B-STVs are called *binary*.

Formula 2:

$$M_{p,q}^{BIS} = \frac{\sum_{i=1}^{n} \min\left(e_{p,i} \cdot e_{q,i}, 1\right)}{\min\left(\sum_{i=1}^{n} \min\left(e_{p,i}, 1\right), \sum_{i=1}^{n} \min\left(e_{p,i}, 1\right)\right)}$$

where:

$M_{p,q}^{BIS}$: Binary intersection metric between user p and q

i : Summing Index B-STV elements

n : B-STV length, upper summing limit

$e_{p,i}$: B-STV element i for user p

$e_{q,i}$: B-STV element i for user q

By this procedure, we get an mxm (m=12) symmetrical matrix with elements based on STVs (Formula 1) or B-STVs (Formula 2). The elements take possible values between zero (no similarity) and one (equality). For Fig. 4, based on STVs, the observed values are between 0.078 and 0.929 (without considering the diagonal elements). For Fig. 5, based on B-STVs, the observed values are between 0.182 and 0.882 (without considering the diagonal elements).

5.2.2 GROUPING ALGORITHM

We interpret the symmetrical exclusion matrix by MDS, obtaining plots like Figs. 4 and 5. The users seem to represent three groups, {N1, N4, N5, N6, E4}, {N2, N3, E1, E2, E5} and {E3, E6}. E3 and E6 may as well be combinations of several strategies.

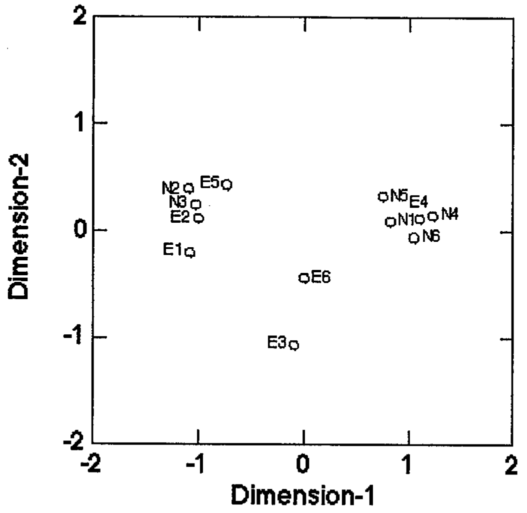


Fig. 4: MDS (r=1, Kruskal, Mono) plot with a normalised intersection matrix gives RSQ=0.975.

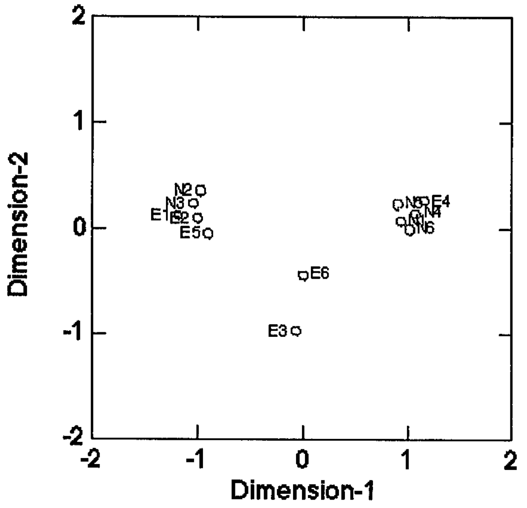


Fig. 5: MDS (r=1, Kruskal, Mono) plot with a binary normalised intersection matrix gives RSQ=0.995.

5.2.3 OUTCOME

According to the RSQ of Fig. 4 (RSQ=0.975) and of Fig. 5 (RSQ=0.995), we can explain most of the variance among user data. However, the binary based plot of Fig. 5 (RSQ=0.995) is slightly better than that of Fig. 4 (RSQ=0.975). That is surprising, since the method ignores information about repetitive behaviour. Maybe such information is redundant in the context of this method.

5.3 EXCLUSION METHOD

This method is based on the exclusion as a metric of difference. Exclusion among two users is always given by two areas. The area of one user STV (user 1, Fig. 2) excluded from the area of a second user STV (user 2, Fig. 2), is not the same as the area of the second user STV excluded from the area of the first one. Since the two exclusion areas are asymmetric, the method does not allow for MDS as grouping algorithm.

5.3.1 METRIC

This method measures the difference between two STVs by estimating how much of one user STV (column index in Table 2) is excluded from a second one (row index, Table 2).

Formula 3:

$$M_{p,q}^{EX} = \sum_{i=1}^{n} \left| \min\left(e_{p,i} - e_{q,i}, 0\right) \right|$$

where:

$M_{p,q}^{EX}$	: Exclusion metric between user p and q
i	: Summing Index STV elements
n	: STV length, upper summing limit
$e_{p,i}$	: STV element i for user p
$e_{q,i}$	: STV element i for user q

Following this procedure for all users, we get an mxm asymmetrical matrix (Table 2), where each element is a measure of exclusion (Formula 3). Since there were six novices (N1-N6) and six experts (E1-E6), m is 6+6=12.

Table 2: Numerical representation of exclusion matrix.

E6	6	43	47	51	70	50	47	35	73	5	171	0
E5	17	15	14	69	48	67	23	7	64	21	0	24
E4	9	44	47	56	70	55	47	35	73	0	171	8
E3	17	41	41	68	77	62	28	28	0	21	162	24
E2	17	15	16	68	81	67	19	0	66	21	143	24
E1	20	16	19	73	85	72	0	7	54	21	147	24
N6	3	41	44	28	48	0	47	30	63	4	166	2
N5	3	39	42	41	0	33	45	29	63	4	132	7
N4	2	41	42	0	55	27	47	30	68	4	167	2
N3	16	11	0	68	82	69	19	4	67	21	138	24
N2	18	0	15	71	83	70	20	7	71	22	143	24
N1	0	41	43	55	70	55	47	32	70	10	168	10
	N1	N2	N3	N4	N5	N6	E1	E2	E3	E4	E5	E6

5.3.2 GROUPING ALGORITHM

The grayscale representation (Fig. 6) of the exclusion matrix (Table 2) is generated by Mathematica (Wolfram, 1991) *ListDensityPlot* with the negative, inverted exclusion matrix as input. We use the negative matrix to obtain a consistent plot. Fig. 6 is only meant as a visualisation of Table 2, and is not an exact mapping. Since division by zero is not defined, the diagonal elements of Table 2 were directly mapped to the darkest graytone. Fig. 6 shows to what degree a *column* user STV is excluded from a *row* user STV. Darker matrix elements correspond to lower degree of exclusion.

To interpret the degrees of exclusion in Table 2, we suggest an iterative *predictor-corrector* algorithm. The *corrector* is an estimator for the threshold value so that only considering exclusion measures between that value and zero will give the predicted number (*predictor*) of user groups. The stop criterion for the iteration method is that the number of user groups given by the corrector, equals the value of the predictor. Research on converge criteria is part of our future work, so now we simply assume convergence. For each iteration the corrector is modified in order to meet the stop criterion, according to the following rules: If we consider too few exclusion relations (i.e. the corrector is too close to zero), the number of groups will be higher than the predictor. If we consider too many exclusion (i.e. the corrector is too far from zero), many or all of the users will be related by exclusion statements, and the number of groups will be lower than the predictor. We give our predictor the value predictor=3. By visual inspection of Fig. 6 it appears reasonable to consider the darkest matrix elements only. Since these elements have numerical values equal to or below 8 (Table 2), we choose the initial value of the corrector to be 8.

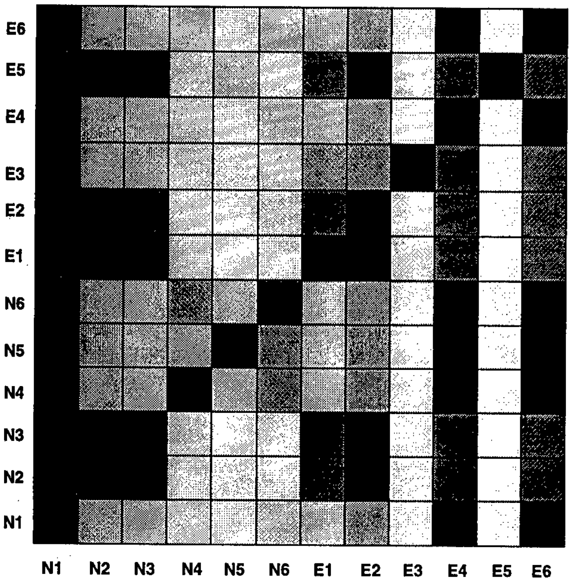


Fig. 6: Grayscale representation of exclusion matrix. Darker elements mean higher exclusion of *column* user STV from *row* user STV.

Diagonal elements are ignored, since each STV is fully similar to itself.

Since small differences indicate similarity, we can derive (based on Table 2) four similarity relations (Table 3).

Table 3: We can derive these four similarity relations.

Similarity relation	User STVs of each relation
1	N1 ∈ N4, N5, N6, E6
2	E4 ∈ N4, N5, N6, E6
3	E6 ∈ N4, N5, N6, E4
4	E2 ∈ N2, N3, E1, E5

All users that are related by an similarity relation are defined to belong to one group. Since the three first similarity relations (Table 3) are interrelated, this gives one group. The remaining, fourth similarity relation (Table 3) gives a second group. Users not appearing in any similarity relation define a separate group.

5.3.3 OUTCOME

Hence, the algorithm gives the following groups: {N1, N4, N5, N6, E4, E6}, {N2, N3, E1, E2, E5} and {E3}. We assumed that the number of groups should be three, so the stop criterion has already been met. If our prediction had not been met, we would have to try with a higher or lower corrector (according to the above mentioned rules) and go back to the start of the predictor- corrector algorithm. This algorithm is repeated until the stop criterion is met (convergence).

6 DISCUSSION

In order to validate the outcome of these three automatic methods, we performed a protocol analysis (Ericsson and Simon, 1984) of the task. This is manual work, based on analysis of video and verbal utterances in addition to logfiles. This is mostly feasible for simple tasks, where users basically follow one or a few strategies. This analysis showed that there are three distinct strategies solving the task. We call these strategies S1, S2 and S3. Table 4 shows the users according to their successful strategy.

Table 4: Manual protocol analysis of the task shows three distinct strategies and gives information about which user succeeded by which strategy.

Strategy	Users according to strategy
S1	N1, N4, N5, N6, E4, E6
S2	N2, N3, E1, E2, E5
S3	E3

The strategies are represented as STVs and have the same qualitative interpretation as the STVs of the users (N1-N6) and (E1-E6). We see that the correlation method and intersection method do not correspond fully with the outcome of the protocol analysis. The exclusion method, however, gives exactly the same results. So, the exclusion method is the best one with our combination of system, task and users behaviour. In the future, we want to find out how the different methods, especially the exclusion method, perform with other, more complex tasks.

We have seen that for a relatively simple task, the method which is purely analytical (exclusion method) is the best one. Measured by the RSQ-values, the intersection method is better than the correlation method, which is purely statistical. This indicates that in our context, statistical methods offer less explaining power than the analytical methods for strategy and plan recognition.

7 CONCLUSION AND FUTURE PERSPECTIVES

We have acquired results for one task only. To make our methods more reliable, we need to evaluate several tasks. For each task, we will validate our methods by manual protocol analysis.

We also plan to study learning experiments, in order to recognise the acquisition process of strategies.

8 REFERENCES

Ericsson, K. A., & Simon H. A. (1984). Protocol analysis, verbal reports as data. The MIT Press.

Rauterberg, M. (1992). An empirical comparison of menu selection (CUI) and desktop (GUI) computer programs carried out by beginners and experts. *Behaviour and Information Technology 11*, pp. 227-236.

Rauterberg, M. (1996). A Petri net based analyzing and modelling tool kit for logfiles in human computer interaction. In (Yoshikawa, H., & Hollnagel. E., eds.) *Proceedings 'Cognitive Systems Engineering in Process ControlSCEPC'96*. Kyoto University, pp. 268-275.

Rauterberg, M., & Fjeld, M., & Schluep, S. (1997). Parallel or event driven goal setting mechanism in Petri net based models of expert decision behaviour. In (Bagnara, S., & Hollnagel, E., & Mariani, M., & Norros, L., eds.) *Proceedings of CSPAC'97*. CNR, Roma, pp. 98-102.

Ritter, F. E., & Larkin, J. H., (1994). Developing Process Models as Summaries of HCI Action Sequences. *Human Computer Interaction* 9, pp. 345-383.

SYSTAT Inc. (1989). SYSTAT®: The system for statistics. pp 93-166. SYSTAT program version 7.0.1 for PC.

Tauber, M. J., (1985). Top down design of human-computer systems from the demands of human cognition to the virtual machine - an interdisciplinary approach to model interfaces in human-computer interaction. In Proceedings of the IEEE workshop on languages for automation (Palma De Mallorca (E) June 28-29), pp. 132-140.

Wolfram, S. (1991). Mathematica®, A system for Doing Mathematics by Computer. 2nd Ed., AddisonWesley, pp. 164, 395, 819. Mathematica program version X3.0.1.1 for Silicon Graphics IRIX.

A Psychological Model of Air Traffic Control and Its Implementation

Cornelia Niessen, Sandro Leuchter, & Klaus Eyferth
Centre for Man-Machine-Systems Studies
Technical University of Berlin
TIB 4/5-3, Gustav-Meyer-Allee 25
D - 13355 Berlin
{niessenlleuchterleyferth}@zmms.tu-berlin.de

ABSTRACT

In this paper, we describe a model of en-route air traffic controllers' cognitive activities in a dynamic man-machine system. The implementation of the model MoFl (*Modell der Fluglotsenleistungen*) is based on a production system in the programming language ACT-R (Adaptive Control of Thought - Rational, Anderson, 1993).

KEYWORDS

ACT-R, dynamic mental representation, air traffic control

INTRODUCTION

For various reasons, it can be useful to have a computer model of the operator's cognitive skills (see e.g., Opwis & Spada, 1994). The implementation of complex psychological assumptions

- can provide a more detailed and explicit description of every cognitive process involved than a verbal description,
- can test a theoretical framework by showing if the anticipated effects can be reproduced,
- can serve as a framework for generating hypotheses that support the empirical work, and
- can be used to analyse and predict the effects of future technological changes on the operator's cognitive activities in complex man-machine systems. These insights into the consequences affecting cognitve performance can be helpful for future system design or training concepts.

On the basis of a broad empirical work - interviews, simulation experiments, memory tests, and a card sorting task with experienced and less experienced en-route air traffic controllers and of theoretical considerations, the interdisciplinary research group "En-route Controller's Representation" (EnCoRe) constructed a model MoFl (*Modell der Fluglotsenleistungen*) of the cognitive activities of experienced en-route air traffic controllers. The air traffic control domain serves here as an example to model cognitive processing during control of complex and dynamic situations. The focus has been on issues concerning problems inherent to dynamic situations: mental representation of the changing situations, and the context-dependent flexible coordination of concurrent cognitive tasks. In comparison to other research (Freed & Johnston, 1995, Bass et al., 1995) in our approach we concentrated on modelling the cognitive abilities of air traffic controllers rather than perceptual and motor skills. According to the rate at which traffic situations changes, and the cognitive task of air traffic controllers, perceptual and motor skills were only treated in order to ensure a realistic model - environment interaction.

The implementation of the model is based on a production system in the programming language ACT-R 3.0 (Adaptive Control of Thought - Rational, Anderson, 1993). As programming environment, ACT-R includes a broad and detailed theoretical framework of human cognition. For the most part, ACT-R is suitable for modelling the cognitive performance of en-route air traffic controllers. But, for some aspects of dynamic situations ACT-R does not provide convincing solutions.

The aim of this paper is to present the construction and the implementation of the model. This includes the principles of construction and implementation of our model, and the discussion of two special issues concerning the cognitive architecture of ACT-R: "dynamic representation" and "executive control". This paper is divided into three sections:

- short description of the air traffic control task
- the framework for the implementation: the cognitive architecture ACT-R
- description of the psychological assumptions of the model and its implementation

THE AIR TRAFFIC CONTROL TASK

On the basis of different sources of information (e.g., radarscreen, flight strips, head-phone communication with pilots), air traffic controllers have to control complex, dynamic, and time-constraint traffic situations in order to diagnose risky relationships between aircraft and to solve potential conflicts. Therefore, they have to perceive, comprehend, and anticipate multiple characteristics of many aircraft while new incoming aircraft create new traffic relationships for evaluation. It's a common assumption, that in complex technological systems of a dynamic nature operators develop a mental representation of the task environment with which they interact. Diagnosis, decisions on future cognitive activities and actions are based on these insights into current and anticipated structures of the changing situation. Air traffic controllers express with the term *picture* (e.g., Whitfield & Jackson, 1982; Falzon, 1982) what is often described as *situation awareness* (e.g., Endsley, 1995; Flach, 1995): a mental representation of the current and future traffic situation.

By modifying the framework of cognitive task analysis (the "decision ladder", Rasmussen, 1986), extensive interviews with seven experienced controllers provided a first explorative functional analysis of main tasks used to build up and maintain this mental *picture* of the traffic situation.

According to verbal reports of the air traffic controllers, the diagnosis of potential conflicts between aircraft contains stages, which are characterized by an increasing restriction and specification of the problem space. These stages are: *observing* the whole situation, *analysing* the parameters of selected aircraft, and *anticipation.* In the first step (*observation*) the operator monitors the whole situation in order to get a quick overview of the whole traffic situation. The goal of conflict detection demands selection strategies during radar-screening to structure the representation (see e.g. Amaldi & Leroux, 1995). According to the verbal reports, experienced controllers classify the aircraft on the basis of these signals (proximity, vertical movement, etc.) into two groups: those aircraft which have to be further analyzed (*analysing the parameters*) and anticipated (*anticipation*) in order to check for future conflicts, and those which are separated safely during that moment. The initial steps towards intervention and conflict resolution could be described according to Rasmussen´s stages (define task, fomulate procedures, and execute).

In order to model the air traffic controller's *picture* and the processes used to build up and to maintain this mental representation of the changing traffic situation, experiments provided a more detailed analysis of the following topics:

- information selection and recall,
- relational structure of the representation, and
- anticipation and conflict management.

The experimental work with real time simulation was based on a realistic simulation system of the control task called "En-route Controllers Representation - Programmable Airspace Simulation" (EnCoRe-PLuS) (Bierwagen, 1996). This system simulates air traffic control scenarios providing radar screen runs, electronic flight strips, and head-phone communication with a ghost-pilot; it also allows the user to set up experimental procedures and to keep logfiles of all system activities.

The results of this empirical work led to the conceptualization and the implementation of a model that describes the cognitive activities of air traffic controllers. The implementation of the model is connected with a modified version of EnCoRe-PLuS. EnCoRe-PLuS provides a real-time simulation environment. Predefined traffic builds up a simulation scenario that interacts with the model:

- The model can actively access new information about the changing traffic situation and can integrate it to its representation of the current situation.
- The model is informed about events within the task environment (e.g., incoming aircraft)
- The model can intervene with the traffic environment in order to solve conflicts.

MODELLING MENTAL PROCESSES OF EXPERIENCED OPERATORS DURING CONTROL OF A DYNAMIC MAN-MACHINE SYSTEM

For modelling mental processes of experienced air traffic controllers during control we have used the production system ACT-R 3.0. ACT-R provides a suitable framework: 1. as a psychological framework of human cognition, it also describes an environment for implementation, 2. ACT-R is based on explicit and very detailed assumptions about the cognitive architecture, and 3. as an environment for implementation, it is available in the public domain at no costs. In addition ACT-R has been applied to modelling a great number of problem solving tasks and is still in progress (e.g., ACT-R Perceptual - Motor Layer, RPM).

Even within such a framework, the conceptualization and implementation of mental processes in dynamic environments, as in the case of air traffic control, demand additional assumptions about three aspects of the dynamic task environment. 1. The continous changes of the situation. These changes do not allow fixed sequences of cognitive processing, they rather call in a cyclic update of varying relations as a basis of situational awareness. 2. The necessity to predict future states of the situation in order to predict potential conflicts. Such predictions alter the goals of ongoing control activities. 3. The demands to coordinate and to sequence simultanious requirements of the control task.

Widely used concepts for adaptive control of complex task enviroments (e.g., Anderson, 1993; Rasmussen, 1986; Hacker, 1978) concentrate on rather static tasks and on invariant goal structures. For example the cognitive architecture of Anderson's ACT-R does not take into account that in dynamic situations the operator has to continuously update her or his mental representation. In addition, such production systems are directed by a fixed goal hierarchy. But in the case of the changing and complex situation requirements, the controller has to coordinate the cognitive activities. This coordination is context-dependent: it does not follow a pre-defined goal hierarchy.

Recently there are some promising attempts to formulate cognitive architectures that deal with the specific demands of a dynamic task environment. For example, as a conceptual neighbor to ACT-R and SOAR, a new computational framework, the executive - process interactive control (EPIC), is proposed for this kind of human performance (Meyer & Kieras, 1997a,b; Meyer et al., 1995). Perceptual, cognitive, and motor processors have been built up for modelling cognitive processes during the performance of multiple concurrent tasks. The perceptual processor provides a continously update of the task environment. Within the cognitive processor, concurrent tasks can be scheduled by flexible executive processes that control relative task priorities. Also the architecture for human representation in complex system, "Man Machine Interactive Design and Analysis System" (MIDAS), promises a modelling environment that provides an updateable mental representation of the task environment and flexible scheduling of multiple task performance (Corker & Smith, 1993).

The implementation of the model "MoFl" (*Modell der Fluglotsenleistungen*) is based on ACT-R 3.0. The basic assumption is that cognitive skills are composed of production rules. A production rule is a modular piece of knowledge. Combining these rules into a sequence represents complex cognitive processes. ACT-R includes two kinds of knowledge representation: declarative and procedural knowledge. The basic units in declarative memory are so-called working memory elements (WMEs). A WME is an object with identity. It has named slots that can be filled with Lisp objects or references to other WMEs. References to other WMEs can be interpreted as relations, so that a semantic net with WMEs as nodes and references for relations is spread out. ACT-R defines an object-oriented structure for declarative memory. Every node in the net is an object of a certain class. A class is declared by naming all slots an object of this class will have. Subclassing is possible. Every WME has an activation level. It is manipulated by the programming environment. A special structure within the declarative part of the memory is the goal-stack. WMEs can be pushed onto and popped from this structure. The topmost WME is the current goal.

Production rules are the procedural part of memory. They consist of a condition and an action part. Conditions and actions refer to WMEs. The application of a production rule is realized by a simple pattern-matching mechanism. In order to support goal-directed performance, the first condition of every production rule must match the current goal. If all conditions of a production rule are true, then the action part is executed. Possible actions are: manipulation of the goalstack (push and pop), creation and deletion of WMEs, and modification of the slots of already retrieved WMEs. An ACT-R run consists of the continous application of production rules.

The prioritizing of processing is controlled by the activation parameter in ACT-R as well as by the current goal. A production is applied if it fires. A rule can fire if all conditions are fulfilled. Typically the fastest production will fire. The speed of application is mainly computed by the time it takes to retrieve the condition WMEs.

Activation signifies the current relevance of a WME for the processing of information. Sources of activation are the encoding process, execution of a production (addition of new WMEs), and creation of a goal node. The more activated a WME is, the faster it is retrieved. This means that if various WMEs match the pattern of a production rule, the most activated WME is retrieved. If various production rules can be applied, that production rule fires that retrieves the most activated WMEs. A WME can only get retrieved if its activation is above a certain level. But in the case of air traffic control there are three cases in which an inactive WME also has to be retrieved. In the first case, the controller has to update his mental representation continuously. Empirical work showed that controllers reduce the problem space by paying attention to meaningful signals for conflict detection during radar-screening. Because of these signal features, aircraft become focal. That means that they are attention demanding objects, therefore highly activated. Aircraft without these features are *extrafocal* (less activated). For these extrafocal aircraft there is no further demand for processing and they become inactive. But, in contrast to ACT-R, these inactive WMEs have to be retrieved in order to update them. Second, activation is increased not only by the encoding process. It is also guided by the encoding of signal features of aircraft. The third case concerns the context-dependend coordination of a goal. The high activation level of a goal that targets the solution of a detected conflict between aircraft can be decreased, it may be put aside for a while if there is enough time remaining for the solution. But at a certain point, activation has to increase suddenly in order to retrieve this WME and to apply the appropriate production rule in order to solve the conflict. Otherwise the both *inactive* aircraft will collide.

Additional features of ACT-R are learning mechanisms to adjust WME and production parameters, partial matching, and the aggregation of production rules. These features are not used in our model.

THE MODEL

In this section, the psychological assumptions, based on experimental work and theoretical considerations, and the implementation of the main components and functions of the model MoFl are summarized.

MoFl describes three main cycles of information processing, (i.e., *monitoring*, *anticipation*, *problem resolution*) operating on different parts of the situation representation, called the *picture* (see Figure 1). The coordination of these processes is driven by *control procedures. Monitoring* and *anticipation* are diagnostic processes (conflict detection), *problem resolution* is the preparatory step for intervention by the controller.

The Monitoring Cycle: Data Selection and Update

The *monitoring cycle* includes data selection procedures and the regular update of aircraft features. In an experiment on data selection, 36 en route controllers had to control familiar and unfamiliar dynamic airspace situations. In order to investigate information selection, data of aircraft on the radar screen and the flight-strip-system were masked, but could be unmasked by moving the pointer of the mouse to the respective location.

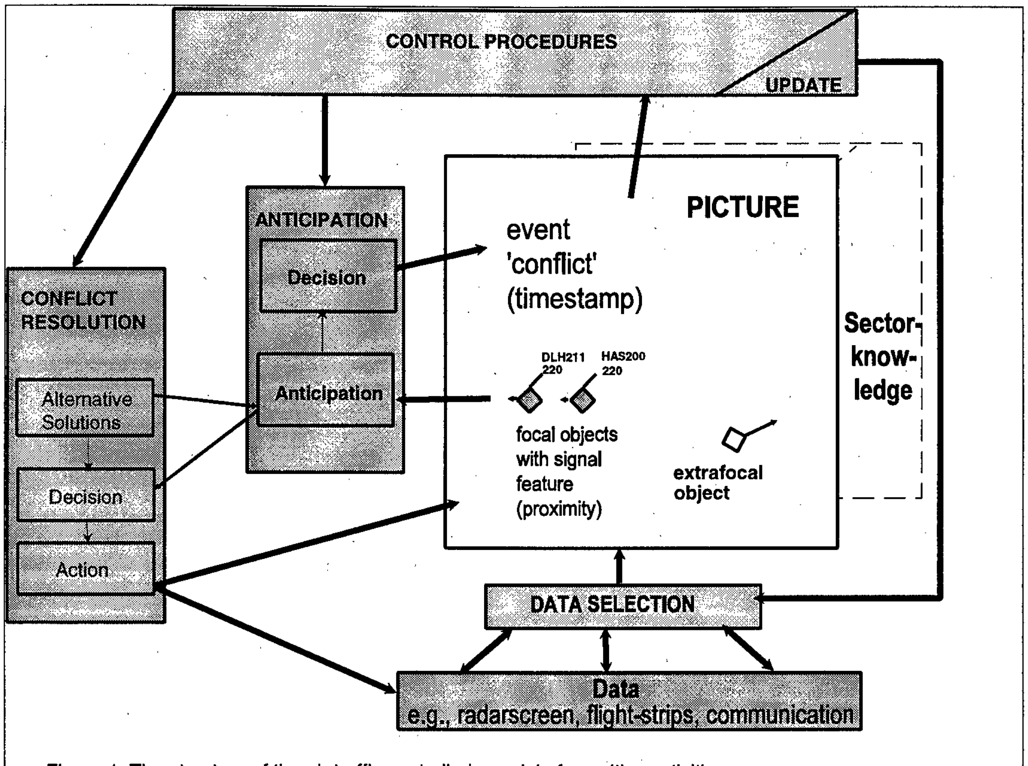


Figure 1: The structure of the air traffic controller's model of cognitive activities

Frequencies and durations of the unmasking were recorded. The data showed, that the representation of the current traffic situation was build up under considerable reduction of information. The controller selects relevant features of aircraft, especially identification codes, the horizontal and vertical positions of objects, and flight directions. In addition, our interviews and the literature indicate that the controller searches for meaningful signals in order to detect conflicts during radar-screening. These are aircraft features like vertical movements, proximity to other aircraft or to points in airspace where conflicts frequently occur (e.g., Niessen et al., 1997; Amaldi & Leroux 1995).

According to these signal features, aircraft become *focal* (highly activated), that means that they are attention demanding objects. Aircraft without such features are *extrafocal* (less activated). In the dynamic environment of air traffic control, objects have to be updated continuously. There is a relationship between the semantics of objects and the frequency of updating: focal, attention-demanding objects demand a higher monitoring frequency than extrafocal objects. This assumption has been supported by results of a memory test: positions of extrafocal (inrelevant) aircraft were reproduced back in time, whereas positions of attention demanding objects (e.g., conflictions, and climb or decend) were reproduced correctly (for similar results, see Boudes et al., 1995). This bias indicates, that there is an interaction between the semantics of objects and the updating frequency: the more the current position of aircraft demands attention the better they were reproduced.

The communication between the controller and the task environment, and the data selection were implemented as follows: Communication between MoFl and EnCoRe-PLuS is realized by socket communication. Two ways of communication are provided:

- *asynchronous communication*: Special events in the task environment, like pilot-initiated radio communication or signals suddenly appearing on the radar-screen, are announced to MoFl by EnCoRe-PLuS. After every application of a production rule, a Lisp function hooked to the ACT-R specific production-cycle-hook, checks for new messages and triggers appropriate Lisp call-back functions that create new WMEs for further processing.
- *synchronous communication*: MoFl identifies an internal demand for new information about a specific object within the task environment or the internal control-flow suggests to update aircraft information. This demand is fulfilled by an active request to the simulation environment. The response is integrated into the *picture* by call-back functions.

If the data selection procedures are triggered, appropriate goals are put onto the goal-stack to enable the following processing sequence:

1. *choose aircraft*: according to aircraft focality and state of the *picture*, decide which aircraft has to be updated.
2. *make an information request*: according to the state of the object which is going to be updated, choose which information has to be requested, and trigger

the appropriate Lisp function. The response of EnCoRe-PLuS is handled by a call-back function that generates a goal.

3. *take new information into the picture*: This goal is processed by a production that modifies the WME representing this information.
4. *test new data for signal features*: the updated WME is tested for changes of signal features such as changing flightlevel (vertical movement), or proximity to other aircraft.

Anticipation

The next step in diagnosis consists of an *anticipation cycle* which operates on the focal objects. For each attention-demanding (*focal*) aircraft or aircraft relationships, a future state is anticipated seperately. The goal of the anticipation cycle is to create new cognitive processing information about aircraft. Depending on the results of anticipation, aircraft with signal features can then be represented as *events*. An event reflects the type of relation between aircraft or relations between aircraft and airspace features in future time and space. The anticipation allows to decide (*decision*) if the future trajectories of aircraft result either in a conflict, in a safe separation, or the demand for more monitoring. In an experiment on conflict-management, different types of clearcut and potential conflicts were varied in a 70 minutes traffic scenario according to the *Eurocontrol Air Space Model* (EUROCONTROl, 1994). The EUROCONTROL classification has two dimensions: 1. different tracks (same, opposite, crossing), and 2. level- or climb/ decend-flight. 36 controllers had to detect and to solve the conflicts. The data showed that controllers did not differentiate between conflicts (separation minimum: 5 nautical miles) and potential conflicts (10 nautical miles): they intervened in all cases. This indicates that conflict detection is not based on a calculation but on fuzzy estimation. The controllers always chose the safer way by overestimating the risk.

We assume that, if a conflict is detected, the event *conflict* includes an estimation of the time remaining for conflict solution (*timestamp*). Relations which have proved to be safe, are no longer in the focal part of the *picture* and become extrafocal at this time. This indicates that there is almost no demand for cognitive processing, except for updating. If the operator is not sure about the potential conflict, the event *monitoring* becomes *focal*, indicating both a higher frequency of monitoring and also a high demand for further anticipation. This distinction of aircraft relationship has been supported by the results of a card sorting task with 18 air traffic controllers. As expected the controller showed a tendency to classify traffic scenarios on the basis of anticipation.

The anticipation cycle is implemented by sequenced production rules testing four questions:

1. Are aircraft on the same airway, or on crossing airways?
2. Have aircraft the sàme altitude or is at least one in climb or descend?
3. Simulation of the future movement of aircraft using *velocity leaders*. A velocity leader is an graphical arrow element on the radar screen showing the estimated movement of aircraft for a certain lapse of time. Will there be a violation of the separation criterion (*anticipation*)?
4. How certain was this simulation? Certainty is measured by the time remaining for the violation of the separation criteria. In addition the latest time for conflict solution is calculated (*timestamp*).

According to this sequence focality of aircraft-WMEs is modified, or events are created.

The Picture

The resulting *picture* is characterized as a representation of objects, events, and objects with reference to other objects, and / or airspace structure. Objects with signal features are represented focally, objects without these features extrafocally. In addition, events which indicate the meaning of aircraft relations in future time and space are represented focally. Within the air traffic control domain, the term *picture* describes the idea of a global mental representation of the current and future traffic situation in working memory. From a psychological perspective, we assume the *picture* as an analogous non-symbolic mental representation of the situation. There is some empirical evidence that experienced controllers anticipate future states of aircraft without calculating the trajectories. This indicates that they build up a non-metric, analogous representation of the situation. In assuming such an analogous representation, we follow Craik's (1943) and Johnson-Laird's (1983) basic ideas of a functional internal model that parallels processes of the external world.

The *picture*

- is understood as an active knowledge-based construction of meaningful relations between elements of a situation, and not as an addition of perceptions.
- is incomplete with regard to the content of information and is temporary. The representation is build up by schemata in order to serve current functions, and is not stored in long term memory.
- can be manipulated by drawing inferences, by making predictions, by understanding phenomena, by deciding what further processing or action to take, and by controlling the execution.

The implementation emulates the *picture* as the totality of the cognitively available objects at a given time, their features, and their perceived and infered relations in actual and future time and space in terms of WMEs. Since it is not possible to model an analogous representation of space on digital computers, the implementation's *picture* is a semantic net of airspace objects, anticipated events, and inferenced actions that are represented as WMEs. Some of these objects have spatial positions that make it possible to define them by positions. More sophisticated operations such as retrieval by distance to other airspace objects have to be emulated.

We used the object-oriented features of ACT-R to define the structure of the *picture* (see, Figure 2). Every airspace object has a position on the radar screen. Derived classes are *airways*, *sector boundaries*, and *aircraft* which have additional slots including callsign, speed, and altitude. Aircraft are specialized to *incoming*, *changing altitude*, and *near to another airspace object* (proximity). For every class, instances are generated and modified as WMEs in working memory by data selecting productions during the monitoring cycle. Events represent infered knowledge about aircraft. All events refer to aircraft objects. Instances are generated by production rules in the anticipation and conflict resolution module. They belong to the event-subclasses: *monitoring*, *conflict*, and *resolution*. Conflicts can be *crossing* or *chain*. Conflict events have an additional slot that holds a reference to the conflict partner.

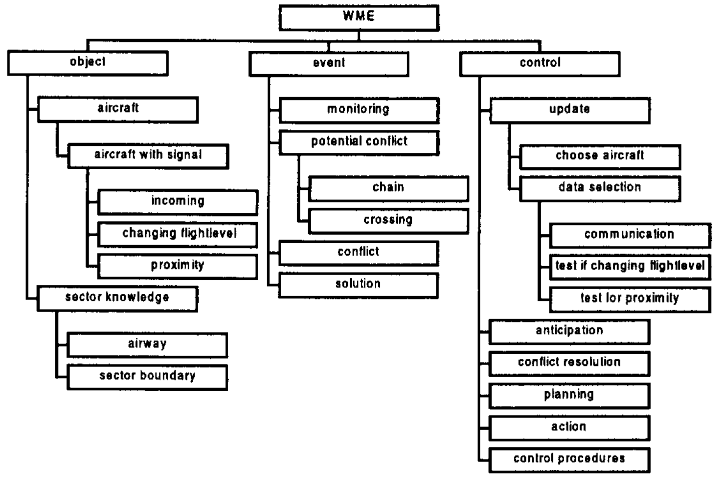


Figure 2: Simplified class hierarchy for the working memory elements

Conflict Resolution

If conflicts are detected, the *problem resolution cycle* initiates several steps to prevent an impending conflict. The controller has to select the most urgent conflict in order to generate or recall solutions (*alternative solutions*). Next, the operator has to check that the solution does not generate new follow up conflicts (*decision*). We assume that the controller checks by running a mental simulation of the solution (as in the *anticipation cycle*). The results of this model are executed (*action*).

The implementation uses a predefined set of standard solutions fitting certain types of conflicts. To use this set the class of the conflict is determined by production-rules. According to this classification some solutions are generated from the standard solution set. The production rules of the simulation in the *anticipation cycle* are triggered by goals indicating the solutions that have to be taken into account. If a solution does not produce follow-up conflicts a solution-WME is generated. A solution consists of a sequence of actions that have to be executed by the model. The time remaining for the first intervention of the sequence is stored in the solution-WME. To execute an intervention sequence Lisp functions interact with the task environment EnCoRe-PLuS.

Control Procedures

The multitude of represented objects, relations, and features within the *picture* demands that the controllers prioritize the processing at any one time. The coordination of the above describes modules (data selection and update, anticipation, conflict resolution, and action) is driven by *control procedures*. We assume that the different processing components cannot be interrupted. The controller has to switch between them: for example, between the solution of a conflict and further monitoring (update including data selection). On the basis of the state of the *picture*, control procedures select the most important and most urgent processing demand.

In ACT-R, Anderson postulates a hierarchical goal structure that directly reflects the task dependency in the environment. To model this hierarchy of goals, several WMEs can be pushed onto the goalstack, a special structure within working memory. Processing is controlled by the current goal, which is the first element of the goalstack. The current goal spreads activation among its neighbors in the semantic net. The system focusses only on this top goal at this time. But, because of the dynamic task environment of air traffic control, there is no fixed hierarchical goal structure. Therefore, the continuously changing situation demands another prioritizing of the processing of simultaneously on-going events at any particular time. In addition, time contraints in this context force a flexible and appropriate selection of the most relevant demand for processing. In order to model this contextualized scheduling of processing, we had to postulate a different concept. Our assumption is that the scheduling of processing is determined by the state of the whole mental representation of the traffic situation.

Several tasks are active at every moment. Every task is done by one of the modules *data selection*, *anticipation*, or *conflict resolution*. The superior control procedures module has to build up an ad hoc process flow depending on the current structure of the *picture*. To achieve this, we assume that the modules cannot be interrupted and are exclusive. The process flow is done by meta productions in the *control procedures* module that trigger a module with an object or event as parameter. In order to trigger a module and make it not interruptible, we introduced a new class of WMEs. These *control*-WMEs are the only ones that get onto the goalstack.

The start of every module is a *top level production*. It is triggered by a *top level goal*. This kind of production will push new subgoals onto the goalstack that will trigger other productions of that module. Every production has to clean the goalstack by popping its trigger-WME. When a module is finished the goalstack should then be clean. The productions of the *control procedures* are triggered by the *controlflow*-goal, which has no parameter. This goal is never popped. Thus when the goalstack is "clean" it is on top of the goalstack and thus the current goal triggers the *control procedures*-module again. Processing radio communication when a plane announces that it is going to enter the sector, is the only reason to interupt a module, make a mark in the working memory, and continue the module. The mark has a high priority so that it will be processed soon.

The meta production rules of the *control-flow*-module for the air traffic controller model use this prioritizing rules:

1. if a solution-WME exists in the *picture* and it is time to solve, then do *action* on this solution, else
2. if a conflict-WME exists and it is time to do, then *conflict resolution*, else
3. if a monitoring-event or an aircraft-WME with a signal (*incoming*, *changing altitude*, or *proximity*) exists in the *picture*, then do *update* and *anticipation* on this WME, else
4. if an aircraft-WME exists, then do *monitoring* on it.

Every solution-WME and every *conflict*-WME has a slot, where it represents when it is supposed to happen. The control productions use a function, that compares this ideal time with the current time. It fires the appropriate action according to a predefined bias.

If the current goal is *controlflow*, only the meta-productions are able to fire. They match patterns against the *picture* according to the prioritization scheme listed above. The chosen action will generate a new *control*-WME (CF) of the appropriate subclass. It refers to the detected aircraft-WME or event-WME. The goalstack consists now of (*controlflow*,CF). This triggers the toplevel production for CF. It will produce new control-WMEs probably refering to the detected WME, pop CF, and put the new control-WMEs onto the goalstack. They trigger new sublevel productions that all pop their trigger. When the module for CF is finished, the goalstack is (*controlflow*), meaning that only the meta-productions are able to fire.

The model deals well with the dynamic environment by using this control scheme. If another task needed interruptible modules, the control procedures would have to be triggered after every production cycle within the module, and the controlflow WMEs would have to be stored in the *picture*, when they are inactive. The meta productions would then trigger the most important controlflow-WME or generate a new one.

CONCLUDING REMARKS: EVALUATION OF THE MODEL

The construction and implementation of the above described model is based on a broad experimental work. Early in 1998 we will evaluate our model with empirical data. Three simulation experiments with experienced air traffic controllers are planned in order to investigate time parameters of conflict detection, the content of the *picture*, and the distribution of activation within the controller's *picture*. These data will be compared to the results of model simulation runs using the same task environment.

ACKNOWLEDGMENTS

This research project has been sponsored by the Deutsche Forschungsgemeinschaft, Ey 4/16-2 , Fr 375/48-2. We gratefully acknowledge the contributions of Thomas Bierwagen (DFS) to the conceptions of our model.

REFERENCES

Amaldi, P. & Leroux, M. (1995). Selecting Relevant Information in a Complex Environment: The Case of Air Traffic Control. In: Norros (Eds.), *5th European Conference on Cognitive Science Approaches to Process Control* (pp. 89-98). Finland: VTT Automation.

Anderson, J.R. (1993). *Rules of the Mind.* Hillsdale: Lawrence Erlbaum.

Bass, E.J., Baxter, G.D. & Ritter, F. (1995). Creating Models to Control Simulations: A Generic Approach. AI & Simulation, pp 18-25.

Bierwagen, T. (1996). Programmsystem EnCoRe-PLuS. Über die Möglichkeiten der programmierbaren Luftraumsimulation. *ZMMS-Forschungsbericht, 96-2*. Technische Universität Berlin.

Boudes, N., Amaldi, P., Cellier, J.M. & Leroux, M. (1995). Forseeing Judgement in an Informationally Rich Environment: The Case of Air Traffic Control. In: Norros (Eds.), *5th European Conference on Cognitive Science Approaches to Process Control* (pp. 76-88). Finland: VTT Automation.

Corker, K.M. & Smith, B.R. (1993). An Architecture and Model for Cognitive Engineering Simulation Analysis: Application to Advanced Aviation Automation. Presented at AIAA Conference on Computing in Aerospace, San Diege, CA. available at http://george.arc.nasa.gov/af/aff/midas/www/AIAA_Final.txt

Craik, K.J.W. (1943). *The Nature of Explanation.* Cambridge: University Press.

Endsley, M.R. (1995). Toward a Theory of Situation Awareness in Dynamic Systems. *Human Factors, 37(1),* 32-64.

Eurocontrol (1994). The Eurocontrol Airspace Model: An Introduction for Potential Users. Eurocontrol-Workshop "Cognitive Aspects in ATC", May 1996.

Falzon, P. (1982). Display Structures: Compatibility with the Operator's Mental Representation and Reasoning Process. *Proceedings of the 2nd European Annual Conference on Human Decision Making and Manual Control*, pp 297-305.

Flach, J.M. (1995). Situation Awareness: Proceed with Caution. *Human Factors, 37(1),* 149-157.

Freed, M. & Johnston, J.C. (1995). Simulating Human Cognition in the Domain of Air Traffic Control. In M.T. Cox & M. Freed (Eds.), 1995 AAAI Symposium on Representing Mental States and Mechanism. Menlo Park, CA: AAAI Press.

Hacker, W. (1978). *Allgemeine Arbeits- und Ingenieurspsychologie*. Bern: Hans Huber.

Johnson-Laird, P.N. (1983). *Mental Models*. Cambridge, Massachusetts: Harvard University Press.

Meyer, D.E. & Kieras, D.E. (1997a). A Computational Theory of Executive Processes and Multiple-Task Performance: Part 1. Basic Mechanisms. *Psychological Review, 104(1)*, pp 3-65.

Meyer, D.E. & Kieras, D.E. (1997b). A Computational Theory of Executive Processes and Multiple-Task Performance: Part 2. Accounts of Psychological Refractory-Period Phenomena. *Psychological Review, 104(4)*, pp. 749-791.

Meyer, D.E., Kieras, D.E., Lauber, E., Schumacher, E.H., Glass, J., Zurbriggen, E., Gmeindl, L. & Apfelblatt, D. (1995). Adaptive Executive Control: Flexible Multiple-Task Performance Without Pervasive Immutable Response-Selection Bottlenecks. *Acta Psychologica 90*, pp. 163-190.

Niessen, C., Eyferth, K. & Bierwagen, T. (1997). Modelling Cognitive Processes of Experienced Air Traffic Controller. In: S. Bagnara, E. Hollnagel, M. Mariani & L. Norros (Eds.), *Sixth European Conference on Cognitive Science Approaches to Process Control, Time and Space in Process Control*, 23.9.-26.9.1997 Baveno, Italy.

Opwis, K. & Spada, H. (1994). Modellierung mit Hilfe wissensbasierter Systeme. In: *Enzyklopädie der Psychologie* (pp 199-248). Göttingen: Hogrefe.

Rasmussen, J. (1986). *Information Processing and Human-Machine Interaction.* New York: North-Holland.

Whitfield, D. & Jackson, A. (1982). The Air Traffic Controller's Picture As an Example of Mental Model. In: G. Johannsen & J. E. Rijnsdorp (Eds.), *Proceedings of the IFAC Conference on Analysis, Design and, Evaluation of Man-Machine Systems* (pp 45-52). London: Pergamon Press.

Spatial Learning and Localization in Animals: A Computational Model and Behavioral Experiments

Karthik Balakrishnan, Rushi Bhatt, and Vasant Honavar
Artificial Intelligence Research Laboratory
Department of Computer Science
Iowa State University
Ames, IA - 50011, USA.
+1 515 294 3588
{balakris|rushi|honavar}@cs.iastate.edu

ABSTRACT

This paper describes a computational model of spatial learning and localization. The model is based on the suggestion (based on a large body of experimental data) that rodents learn metric spatial representations of their environments by associating sensory inputs with dead-reckoning based position estimates in the hippocampal place cells. Both these sources of information have some uncertainty associated with them because of errors in sensing, range estimation, and path integration. The proposed model incorporates explicit mechanisms for information fusion from uncertain sources. We demonstrate that the proposed model adequately reproduces several key results of behavioral experiments with animals.

Keywords: cognitive modeling, cognitive maps, Hippocampus, probabilistic localization.

INTRODUCTION

Animals display a wide range of complex spatial learning and navigation abilities (Schone, 1984; Gallistel, 1990), far more impressive than the capabilities of contemporary robots. Considerable research effort has been devoted to understanding different aspects of these spatial behaviors through cognitive, behavioral, neurophysiological, and neuropharmacological studies. This has resulted in a large corpus of experimental data, a number of theories and models of animal spatial learning, and several implementations of such models in robots and other artificial automata (Mataric, 1992; Kuipers and Byun, 1991; Kortenkamp, 1993; Bachelder and Waxman, 1994; Recce and Harris, 1996). However, animal spatial learning is still far from being completely understood or successfully imitated.

Based on a large body of experimental data it has been suggested that rodents learn *cognitive maps* of their spatial environments (Tolman, 1948). These cognitive maps have been postulated to contain *metric* information, i.e., the places in the environment are represented in a metric coordinate system, allowing the animal to take novel short-cuts and measured detours. In addition, there is also a vast body of experimental data from *lesion studies* of hippocampal regions and *cellular recordings* of hippocampal cells that directly implicate the *hippocampal formation* in rodent spatial learning (O'Keefe and Nadel, 1978). Based on this data, O'Keefe and Nadel proposed the *locale system hypothesis*, suggesting that the hippocampal place cells learn metric cognitive maps by associating *sensory inputs* with *dead-reckoning*[1] position estimates generated by the animal.

In the two decades since the locale hypothesis was first proposed, a number of computational models of hippocampal spatial learning have been developed (Trullier et al., 1997). Surprisingly, only a few of the models support *metric* spatial representations. Furthermore, the few models that are based on the locale hypothesis make the unrealistic assumption that the two information streams, namely, sensory inputs and dead-reckoning, are largely error-free. However, sensory and dead-reckoning systems of animals are prone to several sources of errors (e.g., errors in place recognition, distance estimation, dead-reckoning drifts, etc.), and any computational model of hippocampal spatial learning and localization must therefore be capable of satisfactorily dealing with these associated uncertainties.

In this paper we develop a computational model of hippocampal spatial learning that allows the animal to learn a metric place map (or a *cognitive map*) and that explicitly addresses information fusion from uncertain sources. Following a brief discussion of experimental data supporting the model, we present the key features of the model and simulation results that demonstrate that the proposed model satisfactorily reproduces the results of behavioral experiments on gerbils reported by Collett et al., (1986). We also discuss the relationship between this neuro-cognitive model and some approaches to spatial learning that have been employed in contemporary robotics.

HIPPOCAMPAL SPATIAL LEARNING

The *hippocampal formation* is one of the highest levels of association in the brain and receives highly processed sensory information from the major associational areas of the cerebral cortex (Churchland and Sejnowski, 1992). It is composed of the *dentate gyrus* (Dg), and areas *CA3* and *CA1* of Ammon's horn as shown in Figure 1. It receives input primarily from the *entorhinal cortex* (EC), which is a part of a larger convergence area called the *parahippocampal cortical area*, and outputs to the *Subiculum* (Sb) and back to the EC (Churchland and Sejnowski, 1992). (For other anatomical and physiological details the reader is referred to (Churchland and Sejnowski, 1992).)

The *hippocampal formation* has been strongly implicated in animal spatial learning and localization based on evidence from *hippocampal lesion studies* and *cellular*

[1] Dead-reckoning or path-integration refers to the process of updating an estimate of one's position based on self-knowledge of time, speed, and direction of self-motion.

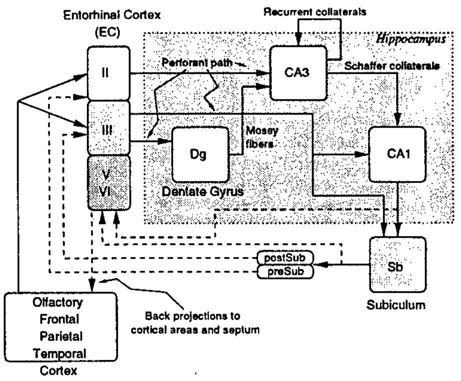


Figure 1: Anatomy of the hippocampal formation.

recordings. While hippocampal lesions have been found to produce *severe deficits* in learning *spatial tasks* such as the *object-place task* (Churchland and Sejnowski, 1992), and the ability of rodents to traverse complex mazes (cf. appendix of O'Keefe and Nadel, (1978)), cellular recordings have led to the discovery of *place cells* and *head-direction cells* which demonstrate *highly correlated* firings during the execution of such tasks. Pyramidal cells in regions CA3 and CA1 of the rat hippocampus have been found to fire selectively when the rat visits particular regions of its environment. These cells thus appear to code for specific places and have been labeled *place cells* (O'Keefe and Dostrovsky, 1971). Cells with such location-specific firing have been found in almost every major region of the hippocampal system, including the EC, the Dg, regions CA3 and CA1, the Sb, and the postsubiculum.

In addition to place cells, *head-direction cells* have also been discovered in the hippocampal region (Taube et al., 1990). These cells respond to particular orientations of the animal's head irrespective of its location in the environment and fire only when the animal faces some particular direction (over an approximately 90 degree range) in the horizontal plane. These cells thus appear to function as some sort of an *in built compass*.

A number of experiments have served to identify crucial properties of place cells and head-direction cells (see McNaughton et al., (1996) for a detailed exposition of the properties). In brief, these cells have been found to respond to sensory as well as path-integration inputs. Further, places appear to be represented by an *ensemble* of cell firings, with the cells being active in *multiple* environments and often at *multiple places* in the *same* environment. The firing of these cells is conserved in darkness, provided the animal is first allowed to orient itself under lighted conditions. Further, any restraint on active motion ceases the cell firings.

HIPPOCAMPAL COGNITIVE MAP

Based on extensive experimental evidence it has been suggested that rodents learn *cognitive maps* of their environments (Tolman, 1948). These cognitive maps are metric in nature, i.e., the spatial representation encodes distances and directions between the environmental cues. Against this background, (O'Keefe and Nadel, 1978) forwarded the *locale system hypothesis* (based on an immense corpus of neurophysiological and behavioral data) suggesting that the cognitive map resides in the hippocampus and that the place cells use sensory and dead-reckoning inputs to encode the metric map. A computational implementation of this locale system hypothesis of hippocampal spatial learning has been developed which allows the animal to learn its environment in terms of *distinct places*, with the *center* of each place also being labeled with a *metric position estimate* derived from dead-reckoning. A detailed treatment of this model can be found in (Balakrishnan et al., 1997); here we will only present a brief summary.

As the animal explores its environment the model creates new EC units that respond to landmarks located at particular positions *relative* to the animal. Concurrent activity of EC units defines a *place* and CA3 place cells are created to represent them. These sensory input-driven CA3 place cells are then associated with position estimates derived from the dead-reckoning system to produce place firings in the CA1 layer. Thus, the firing of CA1 cells is dependent on two information streams: sensory inputs from CA3 and the animal's dead-reckoning position estimates. The dead-reckoning input is used to learn the center of the place in terms of metric coordinates.

When the animal revisits familiar places, incoming sensory inputs activate a place code in the CA3 layer that corresponds to a familiar place. Since multiple places in the environment can produce the same sensory input (called *perceptual aliasing* in robotics), the CA1 layer uses dead-reckoning estimates to disambiguate between such places and produces a *unique* place code that corresponds to the current place. The hippocampal system then performs spatial localization by *matching* the *predicted* position of the animal (its current dead-reckoning estimate) with the *observed* position of the place field center (dead-reckoning estimate previously associated with the activated CA1 place code). Based on this match, the dead-reckoning estimate as well as the place field center are updated as shown in Figure 2.

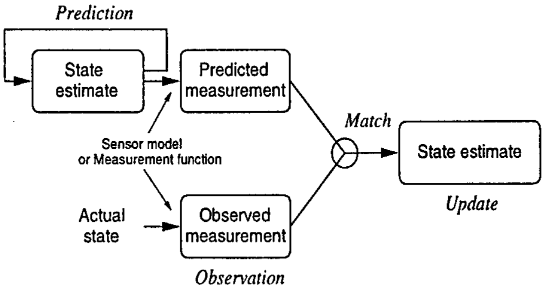


Figure 2: A schematic of hippocampal localization.

Thus, not only does the hippocampal model learn a metric cognitive map of the environment, but it also permits the metric estimates to be *updated* when the animal revisits familiar places. Further details of the model may be found in (Balakrishnan et al., 1997).

Hippocampal Kalman Filtering

In the locale system hypothesis of hippocampal spatial learning, information is integrated from two streams: the sensory inputs and the dead-reckoning system (O'Keefe and Nadel, 1978). It should be noted that information provided by both these streams is uncertain because of errors in object recognition, distance estimation, and path integration. For instance, the firing of place cells and head-direction cells drift in darkness, suggesting errors in path-integration. Thus, in order for the hippocampus to perform robust spatial localization using these uncertain information sources, it must have adequate mechanisms for handling uncertain information sources. Although several hippocampal models of spatial learning have been proposed, including some that are closely related to the model described above, *none* of the models are capable of *explicitly* handling such uncertainties.

As with animals, mobile robots too have to deal with uncertainties in sensing and action. This has led to many probabilistic localization approaches for mobile robots. One such localization tool is the *Kalman filter* (KF) (Gelb, 1974) (or some extension or generalization of it), which allows the robot to build and maintain a *stochastic spatial map*, propagate sensory and motion uncertainties, and localize in *optimal* ways (Ayache and Faugeras, 1987). A schematic for a KF is shown in Figure 3.

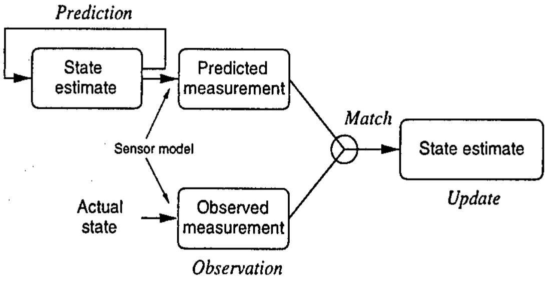


Figure 3: A schematic of Kalman filtering.

As can be observed from Figures 2 and 3, the computational model of hippocampal function and KF both embody the same *predict-observe-match-update* principle. Further, KF provides a framework for performing *stochastically optimal* updates even in the presence of prediction and observation errors. Based on the similarities between the two, Balakrishnan et al. (1997) developed a KF framework for uncertain information fusion in the hippocampal localization model described above. In this framework, KF helps the animal in maintaining and updating an estimate of its own position as well as the estimates of the place field centers. These estimates, referred to as the *state*, include:

$$\mathbf{x}_k \quad = [x_{0,k}, x_1, \ldots, x_n]^T$$

where $x_{0,k}$ denotes the position of the animal at time instant k, x_i denotes the center of place field i, and n is the number of distinct places that have been visited by the animal. Without loss of generality, these position estimates are assumed to be specified in 2D Cartesian coordinates, i.e., $x_i = (x_{i_x}, x_{i_y})$. The animal also computes and updates the covariance matrix associated with this state vector, denoted by $\mathbf{P}_k$, which is given by:

$$\mathbf{P}_k = \begin{pmatrix} \mathbf{C}_{00} & \mathbf{C}_{01} & .. & \mathbf{C}_{0n} \\ \mathbf{C}_{10} & \mathbf{C}_{11} & .. & \mathbf{C}_{1n} \\ . & . & . & . \\ . & . & . & . \\ \mathbf{C}_{n0} & \mathbf{C}_{n1} & .. & \mathbf{C}_{nn} \end{pmatrix}$$

where

$$\mathbf{C}_{ij} = \begin{pmatrix} C_{i_x j_x} & C_{i_y j_x} \\ C_{j_x i_x} & C_{j_y i_y} \end{pmatrix}$$

denotes the covariance between the 2D Cartesian representations of the state elements $x_i = (x_{i_x}, x_{i_y})$ and $x_j = (x_{j_x}, x_{j_y})$.

When a new place is visited, the state vector is augmented by the center of this new place and the state estimate and its covariance matrix are modified accordingly. If the animal motions are assumed to be *linear* and the *measurementfunction* in Figure 3 is also a *linear* function of the state, a framework for *hippocampal Kalman filtering* can be developed that updates the place field centers and the animal's position estimate in *stochastically optimal* ways. These details can be found in Balakrishnan et al., (1997).

Frame Merging

The procedure described above allows the *animat*[2] to learn a metric place map. However, it does not allow the animat to learn and integrate *independent local* metric maps corresponding to different regions of the environment, or to learn and integrate a *new* map into an existing one. We have developed an extension of the computational model described above that permits the animat to learn separate place maps in different *frames* and to *merge* frames together in a well-defined manner.

Suppose the animat has learned a place map, labeling the places with metric position estimates derived from its dead-reckoning system. Let us refer to this frame as f_{old}. Suppose the animat is now reintroduced at another place. The animat stores away f_{old} in its memory, and begins a new frame f_{new} at the point of reintroduction. It also resets its dead-reckoning estimates to zero, thereby making the point of reintroduction the origin of its new dead-reckoning frame. Now it proceeds as before, learning places and creating EC, CA3, and CA1 cells using the algorithms detailed in (Balakrishnan et al., 1997). At each step it also checks to see if sensory inputs excite CA1 cells residing in f_{old}. If this happens, the animat is at a place it has seen earlier in the older frame (f_{old}). It then *merges* the two frames, labeling the places in the two frames in a uniform coordinate system as follows.

Suppose CA1 unit *c* fires in f_{new} and *m* fires in f_{old}. The goal is to merge f_{old} into f_{new}. We do this by changing the position labels of all CA1 units in f_{old} to equivalent labels in f_{new}. Let $\hat{\mathrm{x}}_c^{f_{new}}$ and $\hat{\mathrm{x}}_m^{f_{old}}$ denote the estimated center of the animat's current place in the two frames f_{new} and f_{old}. Since $\hat{\mathrm{x}}_c^{f_{new}}$ and $\hat{\mathrm{x}}_m^{f_{old}}$ correspond to the center of the same

[2] A simulated animal

place field, albeit in different frames, $\Delta \mathbf{x} = \hat{\mathbf{x}}_m^{f_{old}} - \hat{\mathbf{x}}_c^{f_{new}}$ denotes the amount by which frame f_{old} has to be transformed to coincide with f_{new}. Assuming a metric coordinate representation, we can update the place field centers of f_{old} to f_{new} via the transformation:

$$\hat{\mathbf{x}}_i^{f_{new}} = \hat{\mathbf{x}}_i^{f_{old}} - \Delta \mathbf{x} \qquad \forall i \in f_{old} \tag{1}$$

The covariances between units in f_{old} and f_{new} can be updated using the following expressions (details of the derivations can be found in (Balakrishnan et al., 1998)):
Case I: i and j were both units in f_{old}

$$C_{ij}^{f_{new}} \quad = C_{ij}^{f_{old}} - C_{mj}^{f_{old}} - C_{im}^{f_{old}} + C_{mm}^{f_{old}} + C_{cc}^{f_{new}}$$

Case II: i was a unit in f_{new} and j was in f_{old}

$$C_{ij}^{f_{new}} \quad = C_{ic}^{f_{new}}$$

where C_{ij}^{f} refers to the covariance between units i and j in a particular frame f.

Once these updates have been carried out, frame f_{old} has been effectively merged into f_{new}. However, it must be borne in mind that this frame merge procedure is currently blind to *perceptual aliasing*. Consequently, the animat localizes to the first place that sensorily matches a place it has seen before. If multiple places in the environment produce similar sensory inputs, this procedure will lead to localization problems.

Goal Representation

Since the computational model of (Balakrishnan et al., 1997) allows the animat to learn places in a metric framework, goals encountered by the animat can also be remembered in terms of their metric positions. Thus, when an animat visits a goal location, it computes an estimate of the goal position based on its current dead-reckoning estimate. However, since dead-reckoning is error prone, the remembered (or computed) position of the goal is also erroneous. We need a procedure that explicitly handles this uncertainty, much like the KF for updating place field centers. We have developed a mechanism that maintains and updates the goal location estimate and its variance using the expressions in equation 2

$$\begin{aligned} \hat{\mathbf{x}}_G &= \frac{\sigma_0^2}{\sigma_0^2 + \sigma_G^2}\hat{\mathbf{x}}_G + \frac{\sigma_G^2}{\sigma_0^2 + \sigma_G^2}\hat{\mathbf{x}}_0 \\ \sigma_G^2 &= \frac{\sigma_G^2 . \sigma_0^2}{\sigma_0^2 + \sigma_G^2} \end{aligned} \tag{2}$$

where $\hat{\mathbf{x}}_G$ is the estimated goal position and σ_G^2 its variance, $\hat{\mathbf{x}}_0$ is the current dead-reckoning estimate with associated variance σ_0^2. It can be shown that this update expression *minimizes* the variance of the goal position estimate (Balakrishnan et al., 1998). These update expressions are applied each time the animat reaches the goal. If the animat has never encountered the goal before, the goal variance σ_G^2 is set to ∞. Thus, when the animat encounters the goal for the first time, the above expressions result in the setting of the goal position estimate to the value of the dead-reckoning estimates.

The animats in our experiments navigate to goal locations through two means. If the goal is visible, the animats directly move towards the goal (*goal approaching*). However, if the goal is not visible but the animat has previously visited the goal location and thus remembers its position, it simply moves in a fashion that reduces the discrepancy between its current position estimate and the remembered position of the goal. We call this the *goal seeking* behavior. The goal seek behavior takes the animat along the shortest path to the goal. It is possible that the direct short-cut to the goal is blocked or has obstacles that the animat must then avoid. However, for the purposes of the experiments described in this paper the environments are assumed to be largely open and obstacle-free.

SIMULATION DETAILS

In this paper we attempt to simulate the behavioral experiments of Collett et al.(1986) using the computational model of hippocampal spatial learning described earlier. The experimental setup of Collett et al. consisted of a circular arena of diameter 3.5 meters placed inside a light-tight black painted room. Gerbils were trained to locate a sunflower seed placed in different geometric relationships to a set of visible landmarks. The floor of the arena was covered with black painted granite chips to prevent the gerbil from spotting the seed until it was very close to it (Collett et al., 1986).

In our simulations, we used a square arena of size 20×20 units. The walls of the arena were assumed to be impenetrable and devoid of any distinguishing sensory stimuli. This is in keeping with the original experiment in which the walls were in complete darkness and presumably not visible to the animal. The landmarks, on the other hand, were assumed to be visible to the animat from all points in the arena. The animats could also estimate landmark positions relative to themselves, but this estimate was assumed to be corrupted by a zero-mean Gaussian sensing error with standard deviation $\sigma_S = 0.01$ units per unit distance. Sensory inputs obtained in this fashion were used to generate the activations of the EC layer as well as the place firings of the CA3 and CA1 layers, using the algorithms described in (Balakrishnan et al., 1997). The animat motions were also error-prone, with motion error modeled by zero-mean Gaussians with $\sigma_M = 0.5$. The animats possessed means for fairly accurate dead-reckoning with errors being modeled as zero-mean Gaussians with $\sigma_D = 0.05$ units. Animats could approach a visible goal and were said to have consumed the goal if they entered a circular region of radius 0.33 units around it.

The experiments of Collett et al. were simulated by first setting up the arena with the landmark(s) in the appropriate positions. The animat was then introduced into the arena at a random position and allowed to perform 500 steps of *sensing, processing*, and *moving*. In this mode the animats learned places by inducting EC, CA3, and CA1 units in appropriate ways, and updating the position estimates using the Kalman filtering mechanism described in (Balakrishnan et al., 1997). If the animat happened to see the goal during these sessions, it was made to approach and consume it. This constituted one *training trial*. Once a trial was complete, the animat was removed from the environment and reintroduced at another random position for the next trial. Each animat was subjected to five such training trials. In each trial the animat learned places in a new *frame* and merged frames if they lead to the same place. The firing threshold of CA3 units (`CA3Threshold`), which signals place recognition based on sensory inputs, was set to 0.75 during training.

Once training was complete, the animat was subjected to ten *testing trials* in which the landmarks in the arena were manipulated in specific ways and, importantly, the goal was absent. During these tests the animat was released at predetermined positions in the arena with its dead-reckoning variance set to ∞. Further, spatial learning was turned off in these animats and they were only capable of localizing. The animats had a maximum of 150 steps within which to localize by visiting a familiar place. Un-localized animats were removed from the environment, with that testing trial being dubbed a failure, and the process continued with the next testing trial. During testing, `CA3Threshold` was lowered to 0.25 to enable the animats to localize even if the landmark arrangements had been changed in critical ways. A localized animat was allowed a maximum of 300 timesteps to navigate to the goal using the goal seek behavior described earlier. Since the goals were absent during testing, the animats searched in the region of the remembered goal location. If the animat reached a circular region of radius 0.5 units around the predicted goal location, it was allowed to spend 25 timesteps searching for the goal. After this, the variance of the position estimate of the animat was once again set to ∞ and the animat was permitted to *re-localize* to enable it to correct its localization if it had wrongly localized earlier. This had interesting behavioral consequences as will be explained later.

For the training as well as testing trials, the trajectories followed by the animats were recorded. Also, the 20 × 20 arena was decomposed into cells of size 0.33 × 0.33 and a count of the amount of time spent by the animats in each cell was kept. These statistics for training and testing were computed for *five different animats*. The cell with the largest value (amount of time spent by the five animats) was used to normalize the values in the other cells, and was plotted in the form of a search histogram. Thus, darker cells in the histogram indicate that the animats spent more time in that region of the arena compared to the regions corresponding to the lighter ones. It must be mentioned that the arena size, the histogram cell size, as well as the goal visibility range were roughly chosen to correspond to actual values used by Collett et al.

EXPERIMENTS AND RESULTS

In this section we present simulations of Collett et al.'s behavioral experiments, using the computational model of spatial learning and localization detailed in (Balakrishnan et al., 1997; Balakrishnan et al., 1998).

One Landmark Experiment

In this experiment, Collett et al. placed the seed at a constant distance and orientation from a single landmark and trained gerbils to reliably approach the goal position. They found that well-trained gerbils run directly to the seed when introduced into the environment. Further, in testing trials the gerbils were found to concentrate their search efforts at the expected location of the seed even though the seed was absent (Figure 1 in (Collett et al., 1986)). In our simulation of this experiment, the goal location was 4 units to the south of a single landmark, as shown by the search distribution concentrated in that region (Figure 4, Left). In these figures, filled squares represent landmarks. This compares rather well with the observations of (Collett et al., 1986).

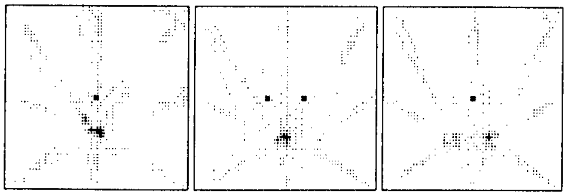


Figure 4: Left: One landmark experiment. Middle: Two landmarks experiment. Right: Two landmarks experiment with one landmark removed.

Two Landmark Experiments

In the next set of experiments, Collett *et al.* trained gerbils to locate a sunflower seed placed to the south of a line connecting two identical landmarks. In this case, the goal was equidistant from the two landmarks. In our simulations, the goal was placed 4 units to the south of the line connecting two landmarks placed 4 units apart. As seen in Figure 4 (Middle), the search effort of the animats is reliably concentrated in a region rather close to the position of the goal in the training trials. This figure compares well with Figure 7b in (Collett et al., 1986).

Collett *et al.* also trained gerbils on the two landmark task and tested them with one landmark removed. They found that the gerbils searched on both sides of the sole landmark apparently matching the landmark either to the left or the right landmark of the original configuration (Figure 7c in (Collett et al., 1986). Our animats demonstrated a similar behavior as seen in Figure 4 (Right).

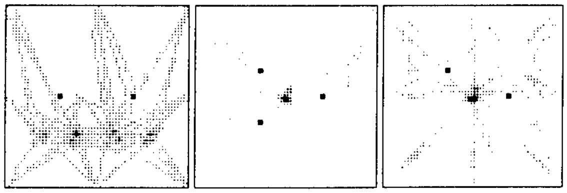


Figure 5: Left: Two landmarks experiment with landmark distance doubled. Middle: Three landmarks experiment. Right: Three landmarks with one removed.

Also, when the gerbils were trained with two landmarks and tested with the landmark distance doubled, Collett *et al.* found that the gerbils searched predominantly at the two interior locations each at the correct distance and orientation from one of the landmarks (Figure 7d). We observed similar search histograms in our experiments, as seen in Figure 5 (Left). We also found that *all* the animats that first searched at the outer locations later searched in one of the interior two locations, when asked to relocalize. Further, *most* animats that first searched at the interior locations, *did not* search at the outer locations upon relocalization.

Three Landmark Experiments

In this experiment, three identical landmarks were arranged to form the vertices of an equilateral triangle with the goal located at the centroid of the triangle. Animats trained in this environment produce search histograms concentrated reliably at the correct position of the goal, i.e., the centroid of the triangle as shown in Figure 5 (Middle). This compares favorably with Figure 6b in (Collett et al., 1986).

Collett *et al.* also trained the gerbils on the three landmark task and tested them in environments with one or two of the landmarks removed. With one landmark removed they found that the gerbils searched at a location at the correct distance and orientation from the two remaining landmarks (Figure 6c). As can be seen from Figure 5 (Right), our animats demonstrate largely similar search behaviors.

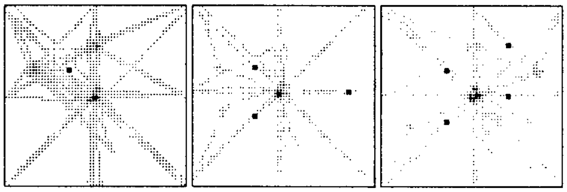


Figure 6: Left: Three landmarks with two removed. Middle: Three landmarks with one distance doubled. Right: Three landmarks with an extra landmark added.

With two of the three landmarks removed, Collett *et al.* found that the gerbils distributed their search time between three sites, one for each of the three possible matches of the sole landmark (Figure 6d). This can be compared directly with our simulation results in Figure 6 (Left). Similarly, when the gerbils were trained on the three landmark task but tested with one landmark distance doubled they were found to search at a goal location at the correct distance and bearing from the two unmoved landmarks (Figure 8 in (Collett et al., 1986)). Our animats display similar behaviors (Figure 6 (Middle)).

When gerbils were trained on the three landmark task, but tested in an environment with an additional landmark placed so as to create another equilateral triangle with a different orientation, Collett *et al.* found that the gerbils reliably searched at the goal location within the correctly oriented triangle. Our simulation of this experiment produced similar results as shown in Figure 6 (Right).

DISCUSSION

In this paper we have extended the spatial learning and localization model developed in (Balakrishnan et al., 1997) along several significant directions. We have developed mechanisms to learn local place maps in *disjoint* frames, and to merge these frames to produce global place maps. We have also incorporated a mechanism for learning and remembering goals in terms of their metric positions, with an associated mechanism for updating goal positions in a stochastically consistent manner. With these additions, animats can not only learn maps of environments in a piecemeal fashion but also learn and reliably navigate to goals in the environment.

This allowed us to simulate the behavioral experiments of (Collett et al., 1986). The primary goal was to test whether our computational model of hippocampal spatial learning and localization was capable of explaining their behavioral data with gerbils. We simulated a number of their experiments and the search histograms generated by our animats were found to be very similar to those produced by the gerbils in their experiments. This is especially interesting because our animats did not remember goals in terms of *independent vectors* to individual landmarks, as suggested by (Collett et al., 1986). Our results indicate that if goals are remembered in terms of metric position estimates, localization errors are enough to explain the search distributions of the gerbils observed in environments with landmark configurations changed.

To the best of our knowledge, the only computational simulation of the (Collett et al., 1986) experiments, apart from the work presented in this paper, is that of (Redish and Touretzky, 1996). In their simulations, the animat was placed at different random positions in the arena and was given its position relative to the goal (which was assumed to coincide with the origin). The animat then created place cells using a combination of this position estimate and sensory inputs from the visible landmarks. *Ego-centric* angles between landmarks were also encoded in the place cells, which allowed the animat to initialize its head-direction if it happened to be disoriented. In test trials they introduced the animat at a random position and allowed it to localize, i.e., the animats performed head-direction and position estimate resets. Once the animat had localized, it could *predict* the goal location which was simply the origin of the coordinate frame with respect to its current localized position. They repeated this process a number times and calculated a histogram of predicted goal positions (Redish and Touretzky, 1996).

Our computational model of hippocampal spatial learning is closely related to that of (Redish and Touretzky, 1996) (referred to hereafter as the RT model) since both models are based on the *cognitive map* concept of (Tolman, 1948) and its implicated substrate in the hippocampus (O'Keefe and Nadel, 1978). Further, both these models make use of the *locale system* hypothesis of (O'Keefe and Nadel, 1978) with places being learned using a combination of sensory inputs and dead-reckoning information. Finally, both simulations represent goals in terms of metric position estimates derived from dead-reckoning.

Despite these similarities, there are some significant differences between the two models and the behavioral results generated by them. Our model assumes that errors exist in the sensory and dead-reckoning input streams and our computational framework explicitly addresses the issue of information fusion from erroneous (or uncertain) sources. By formulating the the place learning and localization problem within the framework of Kalman filtering, we have been able to derive *update expressions* that can be proven to be *stochastically optimal.* The RT model incorporates a mechanism for initializing the head direction. However, doing so makes the place cells *directional*, which appears to be at odds with experimental results that suggest the *non-directionality* of the CA3 and CA1 pyramidal cell firings. Our model assumes that the place cells are non-directional and this requires that the animats have reliable head-direction information, i.e., we assume that the animals have not been disoriented. Further, animals learn and remember multiple goal locations, and plan and execute multi-destination routes. Extending our model to handle learning and representation of multiple goal locations is rather straightforward. However, it is not clear how one could represent multiple goals in the RT model considering that goals in their model correspond to the origin of the dead-reckoning system. Finally, animats in our simulations were capable of actually moving in their environment, whereas the animats used in the RT simulations do not move. Consequently, the histograms reported in (Redish and Touretzky, 1996) correspond to *predictions* of the goal position rather than the time spent by the animat in different regions of the environment. Thus, a dark histogram cell that is far from the goal in the RT model implies that the animat has a completely wrong estimate of the goal position and hence a completely wrong localization, while a similar cell in the histograms of Collett et al. simply means that the animal spent some time in that region localizing (or moving slowly on its way to the goal), and does not necessarily imply that the animal's localization or its prediction of the goal position is wrong. Since the animats in our simulations were capable of navigating, the search histograms generated in our experiments correspond more closely to those reported by Collett et al. (1986).

Other Robot Localization Approaches

Owing to the Kalman filtering framework, our computational model of hippocampal spatial learning is directly related to KF approaches for robot localization (Crowley, 1995; Leonard and Durrant-Whyte, 1992). However, these KF based approaches require a *sensor model* of the environment (as shown in Figure 3) and often run into *matching problems* in environments with multiple identical landmarks and limited sensor ranges. The hippocampal model, on the other hand, provides a *place-based* extension of KF and easily addresses these problems (Balakrishnan et al., 1997). A number of robot localization approaches based on *cognitive mapping* theories (or *multi-level space representations*) have also been developed (Levitt and Lawton, 1990; Kuipers and Byun, 1991; Kortenkamp, 1993). Although closely related to the hippocampal spatial learning model, they are not formulated to computationally characterize a specific brain region and differ in this regard. Finally, a number of *neurobiological models* of robot navigation have been developed (Mataric, 1992; Bachelder and Waxman, 1994; Recce and Harris, 1996). However, these models deal with *topological* space representations (not metric ones), and are thus at discord with the *cognitive map* theory of (Tolman, 1948) and the *locale hypothesis* of (O'Keefe and Nadel, 1978). These differences are treated at length in (Balakrishnan et al., 1997).

Future Work

As we mentioned earlier, our computational model assumes that the animat has an accurate head-direction estimate. This may not be the case if the animal has been disoriented. We are currently exploring the possibility of such a head-direction reset mechanism being implemented by place cells in the *subiculum* with the correction being performed by the *head-direction* cells in the *post-subicular region.* We have also developed a method to incorporate multiple goal locations in the model (Balakrishnan et al., 1998).

Given the fact that Kalman filter based models of place learning and localization satisfactorily reproduce an interesting collection of results from behavioral experiments in animals, it is natural to ask: *Can the hippocampus perform the Kalman filter computations? If so, how?* Some suggestions have been forwarded for the neural basis of these computations in the hippocampus, including the role of CA3 *recurrent collaterals* in the propagation and update of estimates and covariances of the places, *sharp waves* in the consolidation of position and covariance estimates, and the CA1 region in the computation of matrix inversions required for KF (Balakrishnan et al., 1997). These issues remain to be explored and explained, both through computational modeling efforts of neuro-physiological and behavioral phenomena, and through biological studies in living, behaving animals.

ACKNOWLEDGMENTS

Olivier Bousquet's contributions to the theory of Hippocampal Kalman filtering is gratefully acknowledged. Karthik Balakrishnan and Rushi Bhatt are supported by an IBM Cooperative Fellowship and a Neuroscience Graduate fellowship from Iowa State University respectively. This research was partially supported by grants from the National Science Foundation (IRI-9409580) and the John Deere Foundation to Vasant Honavar.

References

Ayache, N. and Faugeras, O. (1987). Maintaining representation of the environment of a mobile robot. In *Proceedings of the International Symposium on Robotics Research*, Santa Cruz, California, USA.

Bachelder, I. and Waxman, A. (1994). Mobile robot visual mapping and localization: A view-based neurocomputational architecture that emulates hippocampal place learning. *Neural Networks*, 7:1083–1099.

Balakrishnan, K., Bhatt, R., and Honavar, V. (1998). Spatial learning in the rodent hippocampus: A computational model and some behavioral results. (in preparation).

Balakrishnan, K., Bousquet, O., and Honavar, V. (1997). Spatial learning and localization in animals: A computational model and its implications for mobile robots. Technical Report CS TR 97-20, Department of Computer

Science, Iowa State University, Ames, IA 50011. (To appear in Adaptive Behavior).

Churchland, P. and Sejnowski, T. (1992). *The Computational Brain.* MIT Press/A Bradford Book, Cambridge, MA.

Collett, T., Cartwright, B., and Smith, B. (1986). Landmark learning and visuo-spatial memories in gerbils. *Journal of Neurophysiology A*, 158:835–851.

Crowley, J. (1995). Mathematical foundations of navigation and perception for an autonomous mobile robot. In *Proceedings of the International Workshop on Reasoning with Uncertainty in Robotics*, pages 9–51. Springer-Verlag.

Gallistel, C. (1990). *The Organization of Learning.* Bradford-MIT Press, Cambridge, MA.

Gelb, A. (1974). *Applied Optimal Estimation.* MIT Press.

Kortenkamp, D. (1993). *Cognitive Maps for Mobile Robots: A Representation for Mapping and Navigation.* PhD thesis, University of Michigan, Electrical Engineering and Computer Science Department.

Kuipers, B. and Byun, Y.-T. (1991). A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Robotics and Autonomous Systems*, 8.

Leonard, J. and Durrant-Whyte, H. (1992). Dynamic map building for and autonomous mobile robot. *International Journal of Robotics Research*, 11(4).

Levitt, T. and Lawton, D. (1990). Qualitative navigation for mobile robots. *Artificial Intelligence*, 44(3):305–360.

Mataric, M. (1992). Integration of representation into goal-driven behavior-based robots. *IEEE Transactions on Robotics and Automation*, 8(3).

McNaughton, B., Barnes, C., Gerrard, J., Gothard, K., Jung, M., Knierim, J., Kudrimoti, H., Qin, Y., Skaggs, W., Suster, M., and Weaver, K. (1996). Deciphering the hippocampal polyglot: the hippocampus as a path-integration system. *The Journal of Experimental Biology*, 199(1):173–185.

O'Keefe, J. and Dostrovsky, J. (1971). The hippocampus as a spatial map: Preliminary evidence from unit activity in the freely moving rat. *Brain Research*, 34:171–175.

O'Keefe, J. and Nadel, L. (1978). *The Hippocampus as a Cognitive Map.* Oxford:Clarendon Press.

Recce, M. and Harris, K. (1996). Memory for places: A navigational model in support of marr's theory of hippocampal function. *Hippocampus*, 6:735–748.

Redish, D. and Touretzky, D. (1996). Navigating with landmarks: Computing goal locations from place codes. In Ikeuchi, K. and Veloso, M., editors, *Symbolic Visual Learning*. Oxford University Press.

Schone, H. (1984). *Spatial Orientation: The Spatial Control of Behavior in Animals and Man.* Princeton University Press, Princeton, NJ.

Taube, J., Muller, R., and Ranck, J. (1990). Head direction cells recorded from the postsubiculum in freely moving rats: I. description and quantitative analysis. *Journal of Neuroscience*, 10:420–435.

Tolman, E. (1948). Cognitive maps in rats and men. *Psychological Review*, 55:189–208.

Trullier, O., Wiener, S., Berthoz, A., and Meyer, J.-A. (1997). Biologically-based artificial navigation systems: Review and prospects. *Progress in Neurobiology*, 51:483–544.

A Connectionist Model of Perceptual Field Dynamics in Homogeneous Stimulus Areas

Eliano Pessa **Maria Pietronilla Penna**

Dipartimento di Psicologia
Università di Roma "La Sapienza"
Via dei Marsi, 78 I-00185 Roma, Italy
+39 6 49917631
pessa@axcasp.caspur.it

ABSTRACT

We present a connectionist architecture to model perceptual-motor processing of subjects engaged in the task of drawing a reproduction of a previously observed point on a white paper sheet. Such a task was designed to investigate the structure of perceptual field. Computer simulations showed a satisfactory agreement between model's forecastings and the experimental data obtained from an experiment performed on human subjects.

Keywords

Perceptual field, neural networks, spatial memory

INTRODUCTION

Every model of human visual perception must take into account the evidence, given by Gestalt psychologists (see, e.g., Koffka, 1935), for global factors of wholistic nature. In most cases, however, the study of such factors was done only in a qualitative way. For this reason Gestalt psychologists were unable to build a formalized theoretical model of visual perceptual processing, designed to do quantitative forecastings of experimental data. Notwithstanding they introduced a fundamental concept, the one of *perceptual field*, viewed as similar to a vector field of forces acting within perceptual space. The lines of force of such a perceptual field should coincide, on one hand, with the paths followed in apparent movement phenomena, whereas, on the other hand, they should be orthogonal to perceived figural contours. A semi-quantitative investigation of perceptual field was undertaken already by Brown & Voth (1937), and by Orbison (1939). Such a task, however, requires to face strong theoretical and experimental difficulties in the case of nonhomogeneous stimulus areas, due to the great number of possible different situations, and of factors to be controlled.

In more recent times some authors (Stadler & Kruse, 1990; Stadler *et al.*, 1991) proposed an experimental procedure to investigate in a quantitative way the perceptual field structure in the case of homogeneous stimulus areas. Such a procedure was, in some way, inspired by Bartlett's early observation of the *wandering point* phenomenon (Bartlett, 1951). The experimental paradigm used to detect this latter can be described as follows. To a first subject is shown a white paper sheet on which, in a particular position, a black point was drawn. After the sheet has been removed, the subject is asked to draw, on a second white paper sheet, a point exactly in the same position in which was placed the point previously observed on the first sheet. After the first subject has drawn the point, the second paper sheet is shown to a second subject which, subsequently, is asked to do, on a third paper sheet, the same task as the first subject. Then the third paper sheet is shown to a thirs subject, and so on. In this way it is possible to obtain an ordered sequence of reproduced points, starting from the first presented one. Such a sequence, once transferred on a single sheet, evidences a wandering path, starting from the first point, which can be considered as a visualization of the line of force of perceptual field passing through this point.

Bartlett's idea appears as very appealing, mainly because the drawn point behaves like a probe, useful to investigate a perceptual field - the one created by sheet boundaries - in an homogeneous stimulation condition, without influencing in an essential way the field itself. However, such a procedure is practically unsuitable to study perceptual field structure in all locations belonging to paper sheet, as it would require a too great number of experimental subjects. A more easily implementable method is the one which makes use of a previous suitable sampling of locations, and, for each sampled location, ask the same subject to reproduce the point drawn in this location. In this way the data coming from a single subject let us obtain the *displacements* (of the reproduced point with respect to the observed point) associated to all sampled locations. These displacements, in turn, are proportional to the vector forces acting in each one of sampled points. We can thus obtain a quantitative representation of perceptual field structure and of its lines of force.

Such a representation, once obtained,, should be considered as a remarkable result, because it lets us characterize in a quantitative way the perceptual field postulated by Gestalt psychologists. However it raises an important problem, concerning the origin of observed perceptual field structure. Does this latter derive from some general Maximum (or Minimum) Principle, such as the one of goodness of form? Or it is a byproduct of sensorimotor processing, required by experimental task described above, and of principles ruling the operation of neural architectures involved? In order to support the evidence for the latter alternative we built a connectionist model designed to represent perceptual-motor processing by the experimental subject engaged in such a task, and to forecast the displacements observed in an experiment we performed, according to the paradigm presented above, on 10 subjects. Such a model was implemented through an architecture constituted by several different neural networks reciprocally interconnected, each one designed to do a particular task. Such a choice was dictated by the complexity of the experimental situation to be modelled. Namely this latter involves, first of all, an *acquisition system*, to grant for input of stimulation patterns, both of the sheet with the drawn point, and of the empty sheet where the point has to be reproduced, together with the instantaneous position of the point of pencil used to draw the reproduction. Moreover, we need a *spatial memory* , to store the information relative to the observed point, and a *motor system*, able to command hand motion in order to move pencil point up to the location where the point should be reproduced. Such a model was implemented through a computer program, and the outcomes of simulations we did were compared with the mean displacements observed in experiment with human subjects. We found a satisfactory agreement between computer simulation results and experimental data. Such an effect was essentially a consequence of general principles underlying the operation of single neural networks belonging to the architecture we described, rather than a consequence of *ad hoc* mechanisms already embodied within our model. Notwithstanding we feel that, in order to obtain a better agreement, some further experimental and theoretical problems remain to be solved.

Before undertaking a detailed illustration of proposed model, we will describe, in the second section, the experiment done on human subjects. The third section will contain a description of the component of our model we consider as the most critical one: the spatial memory. The other networks belonging to model architecture will be presented in a fourth section. The fifth section, then, will be devoted to a description of simulations done, and to a comparison between the results so obtained and experimental data coming from human subjects. The conclusion will be the object of sixth section.

THE EXPERIMENT

The experiment was designed with a procedure similar to the one described, e.g., in Stadler *et al:* (1991), but with a systematic control of experimental variables.

Subjects

The experiment was performed on 10 subjects, all students of Psychology, 5 males and 5 females, all with normal vision, or correct to normal.

Stimuli

The stimuli were constituted by 609 A4-sized paper sheets, each one with a single point in a particular location. Each point had a circular form, whose radius was 1mm. The set of all locations filled a lattice with 29 rows and 21 columns, in which the distance between two neighbouring points, both along the horizontal and the vertical direction, was 1 cm.

Procedure

To each subject were presented, once at time and each one for a duration of 1 s, all 609 stimulus sheets. The subject was sitting in a dark room, before a suitably built device, constituted by a box, with an upper opening to look inside and a lateral opening to insert subject's hand holding a pencil. Only the inner box was enlightened, so that the subject was forced to focus his/her attention only on stimulus sheet. After 1 s the sheet was removed through a suitable opening, existing in the box, by an experimenter , located in the dark , which substituted the stimulus sheet with an A4-sized blank sheet. The subject was asked to draw on this sheet a point exactly in the same location occuped by the point contained within the stimulus sheet presented before. The experimenter controlled that the initial position of subject's hand was always the same across all trials. Once the subject drew the reproduction of the observed stimulus point, the sheet was removed a new stimulation sheet was presented. The presentation order was randomized, and different from subject to subject. Each experimental session was preceded by a training period, to ensure the understanding of the task by the subject.

Results

For each stimulus point and for each subject we measured the difference between the position of the reproduced point and the one of the stimulus point. Such a difference led us individuate the vector field acting in the location of stimulus point, and whence the tangent vector to the line of force of perceptual field passing through this point. Afterwards, we computed for each point a mean tangent vector, by averaging the results relative to the different subjects. The spatial distribution of mean tangent vectors thus obtained evidenced a regular trend (see Fig. 1). More precisely, the majority of straight lines individuated by each tangent vector were crossing in a small number of points, which Stadler *et al.* (1991) identified with the *attractors* of perceptual field. We found a strong evidence for the presence of two attractors located near the two corners on the upper part of the sheet (here the attribute "upper" refers to the observational point of view of experimental subject), in agreement with the findings by Stadler *et al.* On the contrary, we found only a weak evidence for the presence of other two attractors located near the two corners on the lower part of the sheet , differently from what found by the Authors quoted above.

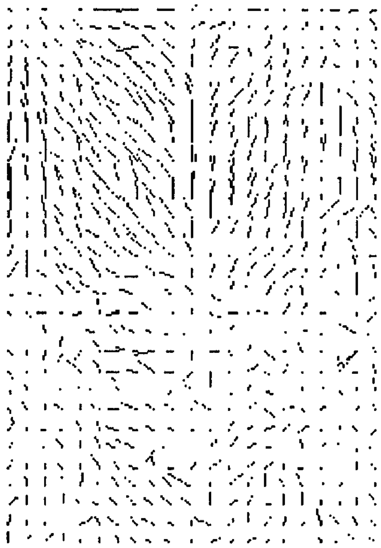


Fig.1

Observed distribution of tangent vectors to lines of force of perceptual field (averaged on all subjects).

MODELLING SPATIAL MEMORY

The general architecture of the model we proposed, to describe perceptual-motor processing by a subject within the experiment described above, consists of the following interconnected neural networks: 1) a *retina* designed to receive input patterns, 2) a *spatial memory*, designed to process retinal output values, and to store the location of the point to be reproduced, 3) two *filtering networks*, designed to detect, respectively, the position of the point to be reproduced (determined as output of spatial memory), and the one of pencil point during reproduction, 4) a *motor network*, designed to give the right motor commands to the hand holding the pencil, as a function of the location of the point to be reproduced, and of the instantaneous position of pencil point.

The choice of implementing the above described subsystems through neural networks was dictated by the following reasons:

a) neural networks algorithms appear as more suitable to model, by using only a small number of rules of interaction between network units, behaviours such as the ones implied by perceptual or motor processing, which, stated in terms of traditional symbolic rule systems (such as the ones expressed through usual Predicate Calculus), would be too difficult to describe; such a circumstance is proved by fast diffusion, in recent times, of neural-network-based systems which do in a very efficient way artificial vision tasks, such as pattern recognition, visual scene analysis, object identification, and motor control tasks;

b) neural network structures appear as closer than usual symbolic rule systems to biological structures involved in visual and motor tasks, so that an interrelation between neurophysiological study and cognitive modelling becomes easier;

c) a parallel hardware implementation of neural network models can be faster than any serial processing of symbolic rules; such an argument would become crucial if our model would be used to command in real time an autonomous robot;

d) neural network algorithms appear as more robust, with respect to traditional symbolic rule systems, with respect to errors, variations of input patterns, variations of model parameter values.

We underline that the previous arguments, within this paper, have nothing to do with the traditional contraposition between symbolic and subsymbolic approach. Our neural network algorithms are symbolic, in the same way as usual symbolic rule systems. We feel only they are more convenient.

Within our model architecture the retina is modelled as a planar lattice of units, each one of which can be, at a given instant of time (henceforth we will suppose the time be discretized: $t = 1, 2, 3, \ldots$), in one of two states: activated or non-activated (corresponding to the activation levels 1 and 0, respectively). As regards neural network representing spatial memory many different modelling possibilities exist. They can be grouped within two fundamental categories: models which make use of *correlation matrices*, and are based on long-range connections, and models implemented through *cellular neural networks*, based on short-range connections. The prototype of models belonging to the first category is the celebrated Hopfield's associative memory model (Hopfield, 1982). There exist, however, memory models belonging to this category, but not directly implemented under the form of neural networks (see, e.g., Pike, 1984; Humphreys, Bain & Pike, 1989). A more recent neural network model of spatial memory of this type is the one proposed by Fukushima *et al.* (1997). A feature common to all these models is that spatial patterns are stored as contributions to a matrix of connection weights, each element of which captures the correlation between two elements of a pattern lying in different locations. This implies that the neural network implementing spatial memory must be constituted by a number of units equal to the one of pattern elements, with connection lines linking every pair of units, independently from the spatial distance between the elements corresponding to the units. The presence of such long-range connections not only is biologically implausible, but can give rise to strong interference effects between stored patterns, if we need to memorize more than one pattern. Such effects can worsen in a dramatic way network performance in recall phase. Moreover, this kind of neural networks appear as particularly suitable to memorize complex patterns, rather than very simple ones, as it is the case in our experiment, where the pattern is constituted by a single point.

The second category of neural network models of spatial memory, the one based on Cellular Neural Networks (CNN), derives from the fundamental paper by Chua & Yang (1988). Shortly, a CNN is constituted by a spatial lattice of units, each one endowed with a particular activation fuction (of neural-like nature), and with a neighbourhood function, stating what units can send their output signals to the input lines of the unit itself. Each line connecting a given unit to its neighbouring units is characterized by a suitable connection weight. In practical applications CNN showed very good performances in artificial vision tasks relative to processing of simple spatial patterns. For this reason we choose this category of models to implement our spatial memory.

Within our model spatial memory was represented as a planar lattice whose dimensions and number of units were identical to the ones of the retina. Each spatial memory unit received input signals both from the retinal unit lying immediately under it, and from its neighbouring units within spatial memory. To this regard, we choose as neighbourhood of a given unit the classical 8-neighbourhood. This means that the neighbouring units of the unit with coordinates (i, j) were the ones with coordinates $(i-1,j-1)$, $(i-1,j)$, $(i-1,j+1)$, $(i, j-1)$, $(i,j+1)$, $(i+1,j-1)$, $(i+1,j)$, $(i+1,j+1)$. If we denote by $x_{ij}(t)$ the activation level of the unit with coordinates (i, j) at the time t, we can write the activation law we choose under the form:

(1) $x_{ij}(t+1) = a\, Q_{ij}\ tgh\, [P_{ij}(t)] - d\, x_{ij}(t),$

where:

(2) $P_{ij}(t) = \Sigma_{r,s \in D}\, w_{ijrs}\, x_{rs}(t) + g\, x_{ij}(t) + I_{ij}(t) - s,$

and D denotes the neighbourhood of the unit (i,j), $I_{ij}(t)$ is the input signal coming from the retina, whereas s is a suitable threshold parameter. The quantities a, d, g denote other parameters to be fixed by the experimenter. Moreover Q_{ij} denotes a factor, depending on $x_{ij}(t)$, we varied, in order to investigate the effect of different choices of activation function on spatial memory performance. The forms of Q_{ij} we used within our computer simulations were the following:

(3.a) $Q_{ij} = 1$

(3.b) $Q_{ij} = 1 - x_{ij}(t)$

(3.c) $Q_{ij} = 1 - /x_{ij}(t)/^{1/3}$

(3.d) $Q_{ij} = 0.5 + x_{ij}(t) - 1.5\, [x_{ij}(t)]^3$

The connection weights w_{ijrs} associated to the lateral connections were varying with time according to a Hebb-like law of the form:

(4) $w_{ijrs}(t+1) = w_{ijrs}(t) + b\, M_{ijrs}\, x_{ij}(t)\, x_{rs}(t-1) +$
$- d\, w_{ijrs}(t),$

where b and d are other parameters, whereas M_{ijrs} is another factor, depending on $x_{ij}(t)$ and $x_{ij}(t+1)$, which we modified in order to investigate the effect of different forms of the Hebbian law on spatial memory performance. The explicit forms of M_{ijrs} we used within our computer simulations were the following:

(5.a) $M_{ijrs} = 1$

(5.b) $M_{ijrs} = 1 - x_{ij}(t+1)\, x_{rs}(t)$

(5.c) $M_{ijrs} = 1 - |\, x_{ij}(t+1)\, x_{rs}(t)|\ 1/2$

In all simulations we performed the operation of spatial memory was observed for a number of time steps, previously fixed by the experimenter. At the end of this period, the activation levels of the units were filtered in the following way. First of all, we searched for the units whose activation level was the maximum one. Once found these units, their activation level was set to 1, whereas the activation level of all other units was set to 0. The units whose activation level was 1 were considered as representing what was stored within spatial memory. In other words, they specified the locations where should be placed the point to be reproduced. Of course, in all computer simulations, only one unit of spatial memory was characterized by an activation level equal to 1. We underline that, apart from specific choices of the factors Q_{ij} and M_{ijrs}, the laws (1) and (4) are nothing but an expression of very general principles ruling neural activation and synaptic facilitation. Thus, the effects of spatial memory operation are to be viewed, essentially, as a consequence of the adoption of such principles.

FILTERING AND MOTOR NETWORKS

When applying our general architecture to modelling human subjects performance in point reproduction task, we needed two filtering networks: one to detect the position of the point to be reproduced, as deriving from spatial memory processing, and another to detect the actual position of the point of the pencil used to draw the reproduction of the point itself. The former network received as input the pattern of activation levels of spatial memory, whereas the latter received as inputs the activation levels of retinal units in presence of the pencil. To do our simulations, we were forced to introduce a particular schematic representation of the pencil together with the hand holding it (as it is perceived by human subjects in the real laboratory experiment). More precisely, we choose to represent the hand through a rectangular array of 3x2 units, to which was attached, in the middle of the longest side, a line of 3 units representing the pencil. We underline that both choices of filtering networks, and of pencil representation, were dictated by the need for proving that a neural-like, and somehow realistic, representation of the information flow from spatial memory to motor network is possible. We acknowledge that other different representations would be possible without changing the operation principles of the neural architecture we proposed. However, we feel that the representation we adopted should be particularly suitable if we would implement our architecture through a particular hardware to be installed within an artificial device, such as a robot able to draw a reproduction of a visually observed pattern.

The filtering network receiving inputs from spatial memory was designed in such a way as to let survive only patterns consisting of a single activated unit. It was implemented through a 2-dimensional array of units (essentially a time-discrete CNN), of slightly greater dimensions with respect to the ones of the retina, in such a way as to include the representation of the hand holding the pencil. Each unit had a 8-neighbourhood and its activation potential was given by:

(6) $P_{ij}(t) = x_{ij}(t) - \Sigma_{r,s \in D}\, w_{ijrs}\, x_{rs}(t) + I_{ij}(t)$

where D denotes the neighbourhood, all other symbols have the meaning defined in the previuous paragraph, and the connection weights w_{ijrs} were all positive. The activation law had the form:

(7) $x_{ij}(t+1) = 1$ if $P_{ij}(t) > 0.5$, otherwise $x_{ij}(t+1) = 0.$

In our simulations the operation time of this network was limited to only one time step.

As regards the second filtering network, the one receiving inputs from the retina and devoted to detect the position of the point of the pencil, we designed it in such a way as to let survive only the unit corresponding to the position of this latter. To this end, we adopted a 2-dimensional array of units, whose dimensions were identical to the ones of the first filtering network. Moreover, by taking again a 8-neighbourhood, we defined the activation potential as:

(8) $P_{ij}(t) = \Sigma_{r,s \in D}\, w_{ijrs}\, x_{rs}(t) - w_{OFF}\, x_{ij}(t) + I_{ij}(t),$

where w_{OFF} and w_{ijrs} were all positive. In our simulations we choose all w_{ijrs} values as identical to a common value w_E. The activation law had the form:

(9) $x_{ij}(t+1) = 1$ if $0 < P_{ij}(t) < (2w_E - w_{OFF})/2$,
otherwise $x_{ij}(t+1) = 0$.

Also in this case the network operation lasted only for one time step.

As regards the motor network, it was designed to transform the knowledge of the actual position of the point to be reproduced, and of the point of the pencil, in a motor command able to induce a displacement of the hand, and whence of the point of the pencil. To this end the coordinates of the point to be reproduced (as deriving from the first filtering network), and of the point of the pencil (as deriving from the second filtering network), were first transformed into a binary form, by using 5 binary digits for each coordinate. Thus, all knowledge relative to the actual positions of the points quoted above was coded through a 20-components binary vector. This latter was used as input for a 3-layer Perceptron, whose output layer contained two units, one devoted to code the motor activation along the horizontal direction, and another to code this activation along the vertical direction. As the allowed motions along these directions could be both positive and negative, we choose, as activation function of the Perceptron units, the hyperbolic tangent one (with a suitable amplification factor).

The Perceptron was trained on a sample of input patterns, containing different relative positions of the point to be reproduced and of the point of the pencil. The desired output to each input pattern was obtained by putting the wanted motor activation along a given direction as directly proportional to the difference between the coordinates of the points quoted above along the same direction. Such a choice was made in conformity with neurophysiological findings (cfr. Schwartz & Georgopoulos, 1987), which evidenced a direct proportionality between the electrical activity of motor cortex neurons and perceived target distance. Of course, the proportionality factor had to be considered as a parameter to be chosen by the experimenter. The training was done through usual error-backpropagation rule. To avoid computational problems, the wanted outputs were divided by a suitable scale factor.

Once trained, the Perceptron was used as a simple input-output device, giving motor activation as a response to the 20-component binary input vector. To compute the effective displacement of the point of the pencil, we set the velocity component of this latter along a given direction as directly proportional to the motor activation along the same direction. Such a choice was made in conformity with recent neurophysiological findings on the correlation between motor cortex activation and limb movement velocity (cfr. Schwartz, 1992; 1993). Once computed the velocity components, the new coordinates x_{new}, y_{new} of the point of the pencil were computed from the old ones x_{old}, y_{old} through the relationships:

(10) $$x_{new} = x_{old} + (kv_x + v_{bx})*\Delta t \;, \qquad y_{new} = y_{old} + (kv_y + v_{by})*\Delta t \;,$$

where k is a proportionality factor, v_x and v_y are the components of the velocity computed as afunction of the corresponding motor activations, v_{bx} and v_{by} are the components of a "base" velocity, whereas Δt is the time step amplitude. The introduction of a base velocity was made to represent the cerebellar modulation of limb movement, whereas the velocity obtained from motor activations represented the power impressed to the movement itself, in conformity with the hypothesis put forward by Flash & Hogan (1985).

COMPUTER SIMULATIONS AND COMPARISON WITH DATA OBTAINED FROM HUMAN SUBJECTS

We used our model architecture to simulate the behaviour of human subjects in the experiment previously described. A number of preliminary trials suggested the following parameter values: a = 0.5, d = 0.1, g = 0.3, s = 0, b = 0.4, d = 0.05, k = 1, Δt = 1, $v_{bx} = 0.1$, $v_{by} = 0.1$, $w_E = 0.15$, $w_{OFF} = 0.1$. Moreover all non-zero connection weights w_{ijrs} appearing in formula (6) were set to 0.8. The motor network was consisting of a 3-layer perceptron whose hidden layer had 2 units. The proportionality factor between motor activation and velocity was chosen as 0.6. The spatial memory processing lasted for 10 time steps after the disappearance of each stimulation pattern, and the hand movement had a limit duration of 10 time steps. We tested our model on the reproduction of the 609 points presented to human subjects. From the positions reached by the point of the pencil, as computed through our model, at the end of the movement period, we derived the tangent vectors through the same procedure used in the case of human subjects.

We did many different simulations with the same parameter values, corresponding to different combinations of choices relative to Q_{ij} and M_{ijrs}. In all cases the results evidenced very clearly the presence of four attractors located near the corners of the sheet, two in the upper part and two in the lower part. As a quantitative measure of model performance we choose for each stimulus point, the euclidean distance between the position reached, within the model, by the point of pencil and the corresponding average position of the point reproduced by human subjects. We then computed the mean value δ of such a distance, averaged on all stimulus points. As other two measures of model performance we choose:

1) the Bravais-Pearson correlation coefficient cy between the vertical components of the tangent vectors, obtained in our simulations, and the ones of mean tangent vectors, obtained from human subjects' data;

2) the Bravais-Pearson correlation coefficient cx between the horizontal components of the tangent vectors, obtained in our simulations, and the ones of mean tangent vectors, obtained from human subjects' data.

The values of δ, cy, and cx obtained in correspondence to the different choices of Q_{ij} and M_{ijrs} are listed in the following (to shorten the exposition, every choice is indicated through the numbers of the corresponding formulae).

A) choice (3.a), (5.a):

$$\delta = 21 \;\; , \;\; cy = 0.34 \;\; , \;\; cx = 0.24$$

B) choice (3.b), (5.a):

$$\delta = 17 \;\; , \;\; cy = 0.43 \;\; , \;\; cx = 0.20$$

C) choice (3.a), (5.b):

$$\delta = 48 \;\; , \;\; cy = 0.16 \;\; , \;\; cx = 0.17$$

D) choice (3.c), (5.a):

$$\delta = 12 \;\; , \;\; cy = 0.58 \;\; , \;\; cx = -\,0.20$$

E) choice (3.c), (5.b):

$$\delta = 11 \;\; , \;\; cy = 0.60 \;\; , \;\; cx = -\,0.21$$

F) choice (3.d), (5.b):

$$\delta = 25 \;\; , \;\; cy = 0.64 \;\; , \;\; cx = 0.18$$

G) choice (3.d), (5.a):

$$\delta = 12 \;\; , \;\; cy = 0.62 \;\; , \;\; cx = 0.17$$

H) choice (3.d), (5.c):

$$\delta = 24 \;\; , \;\; cy = 0.64 \;\; , \;\; cx = 0.19$$

In order to have an idea of the meaning of these numbers, we remember that a value $\delta = 10$ means that the average distance from the points reproduced by our model and the ones reproduced by human beings is only of one lattice cell. We could thus hold that the results obtained from the choices D), E), G) evidence a very good agreement between our model behaviour and the one of human subjects. We should, however, take into account also the values of cy and cx, which show a very strange trend. On one hand, namely, the correlations regarding vertical components evidence a very good agreement between our model and human data, chiefly in correspondence to the choices D), E), F), G), H). The choice G), then, seems to have realized the best compromise between a high value of cy and a small value of δ. On the other hand, the correlations regarding horizontal components appear as too small, in some cases even negative. The highest value was obtained in correspondence to the choice A), which, however, doesn't appear as particularly good, when we look at the values of δ and of cy. From simulation results it appears as evident that neither the choice of Q_{ij} nor the one of M_{ijrs} , isolately considered, can improve the performance of our model. This latter depends on both choices. The best one appears to be G), but the improvement in performance on cx, without a great worsening on cy and δ, suggests that a good research strategy would be the one of investigating what happens by replacing in (5.c) the exponent 1/2 with smaller exponents. In any case, such a circumstance evidences how model's performance depends essentially on general form of laws such as (1) and (4), rather than on particular choices of factors such as Q_{ij} and M_{ijrs} . Another possible explanation of the results we obtained can be found in the existence of some bias which influenced the performance of subjects during the experiment. Namely the variance of their behaviours is very high. Moreover, a comparison we did between our simulations and the results obtained from single subjects showed, in some cases, an agreement better than the one evidenced by a comparison with average subject behaviours, whereas, in other cases, such an agreement was worse. However, only a careful repetition of the experiment with human subjects can tell us whether this is or not the reason for the observed trend of *cx*.

CONCLUSION

The computer simulations so far done evidenced that our model was able to reproduce in a satisfactory way some qualitative (the presence of attractors) and quantitative features of subjects' performance in the point reproduction task. It is, thus, possible, to conclude that our model was able to reproduce some Gestalt-like properties of visual perception, owing essentially to a suitable choice of the dynamical laws underlying spatial memory operation. The usefulness of our proposal stems also from the fact that the neural network architecture we introduced is of modular nature, so that it becomes very easy to investigate the effects on model performance of different choices of the laws ruling the operation of each module. Besides, our model can be easily adapted to represent the cognitive processing of a subject engaged in other sensorimotor tasks, different from the one of point reproduction. A continual , and mutual, interaction between experimental and modelling activity is, however, needed in order that complex model architectures, such as the one we proposed, be useful to improve our knowledge about cognitive system.

REFERENCES

Bartlett, F.C. (1951). *The Mind at Work and Play.* Allen and Unwin, London.

Brown, J.W., & Voth, A.C. (1937). The path of seen movement as a function of the vector-field*American Journal of Psychology, 49,* 543-563.

Chua, L.O., & Yang, L. (1988). Cellular Neural Networks: Theory. *IEEE Transactions on Circuits and Systems, 35,* 1257-1272.

Flash, T., & Hogan, N. (1985). The coordination of arm movements: an experimentally confirmed mathematical model.*Journal of Neurophysiology*, 5, 1688-1703.

Fukushima, K., Yamaguchi, Y., & Okada, M. (1997). Neural network model of spatial memory: Associative recall of maps. *Neural Networks, 10,* 971-979.

Hopfield, J.J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of U.S.A., 79,* 2554-2558.

Humphreys, M.S., Bain, J.D., & Pike, R. (1989). Different ways to cue a coherent memory system: A theory for episodic, semantic, and procedural tasks. *Psychological Review,96,* 208-233.

Koffka, K. (1935). *Principles of Gestalt Psychology.* Harcourt & Brace, New York.

Orbison, W.D. (1939). Shape as a function of the vector-field. *American Journal of Psychology, 52,* 31-45.

Pike, R. (1984). Comparation of convolution and matrix distributed memory systems for associative recall and recognition. *Psychological Review, 91,* 281-294.

Schwartz, A. (1992). Motor cortical activity during drawing movements: single-unit activity during sinusoid tracing. *Journal of Neurophysiology, 68* 528-541.

Schwartz , A. (1993). Motor cortical activity during drawing movements: population representation during sinusoid tracing. *Journal of Neurophysiology, 70,* 28-36.

Schwartz, A., & Georgopoulos, A. (1987). Relations between the amplitude of 2-dimensional arm movements and single cell discharge in primate motor cortex. *Society of Neuroscience Abstracts, 13,* 244.

Stadler, M., & Kruse, P. (1990).The self-organization perspective in cognition research: historical remarks and new experimental approaches. In: H.Haken and M.Stadler (Eds.). *Synergetics of Cognition,* 32-52. Springer, Berlin, Heidelberg, New York.

Stadler, M., Richter, P.H., Pfaff, S., & Kruse, P.(1991). Attractors and perceptual field dynamics of homogeneous stimulus areas. *Psychological Research*, 53, 102-112.

Generating and Classifying Recall Images by Neurosymbolic Computation

Ernesto Burattini, Massimo De Gregorio, Guglielmo Tamburrini

Istituto di Cibernetica C.N.R.
I-80072 Arco Felice (NA), Italy
+39 81 853 4230
{ernb, massimo, gugt}@sole.cib.na.cnr.it

ABSTRACT

The neurosymbolic hybrid system ARCS, which extends a classifier for certain kinds of visually presented objects, generates recall images it is then capable of classifying. The modules performing classification are exploited for imagery, too. In particular, each weightless neural discriminator has been modified so as to generate a non-crisp example of the class of simple visual features it was trained to detect; a symbolic process prescribes how to assemble more complex patterns from such non-crisp examples; both generated features and their compositions are correctly classified, even though the system was originally conceived for actual visual inputs only. These cognitively significant aspects of the hybrid system are examined in the framework of a more general discussion of neurosymbolic integration for cognitive modelling.

Keywords

Hybrid systems, neurosymbolic integration, recall images, weightless neural systems, multidiscriminators, production rules

INTRODUCTION

"If you were going to program a computer to mimic human imagery," Kosslyn (1995, p. 269) remarks "perhaps the most fundamental problem the program would have to solve is the *generation* of images." The neurosymbolic system ARCS (**A**rches **R**ecall and **C**lassification **S**ystem) illustrates a way of solving this problem, relative to images of simple features and their compositions it is already capable of classifying.

The task domain contributes to highlighting the current interest of neurosymbolic integration for AI and cognitive modelling. The generation and classification of the selected, more complex patterns, which seem to elude a purely neural network approach, are naturally handled by means of the hybrid neurosymbolic system. The selected features are various line segments and angles; the more complex patterns represent various *portal shapes*. ARCS grew out of a hybrid classifier for portal shapes (De Gregorio, 1996), embedded into an architectural expert system for landmark building classification and preservation (Burattini, 1994).

There are aspects of the process by which ARCS generates recall images that are significant for cognitive modelling. Human image generation seems to involve a process for producing image parts, and another process for positioning individually activated parts in the image, so as to form more complex visual objects (Farah *et al.*, 1985, Kosslyn, 1994, Kosslyn, 1995, pp. 270-273). The same division of labour applies to the image generation mechanism of ARCS: one can distinguish between simple visual features and more complex patterns, and between two corresponding stages of image generation.

The first stage of image generation is carried out by the neural module of ARCS. This module is a weightless neural system formed by RAM-discriminators (Aleksander & Morton, 1990). The standard weightless discriminator model was slightly modified in order to make a wider repertoire of behaviours available (De Gregorio, 1997). In particular, such a modified discriminator can generate a grey-level, non-crisp example of the class of simple visual features it was trained to detect. The generated example differs from, but bears a precise relationship (spelled out in detail in the following sections) to *every* binary pattern of the corresponding training set. Roughly speaking, the grey intensity level of each non-white pixel in the example is proportional to the number of times that the corresponding memory locations of the discriminator were addressed by input training patterns. Since this relationship shows that each image in the training set contributes to forming the generated class example, a question that naturally arises is whether such recall images might be regarded as *typical* examples of the classes of visual patterns detectable by ARCS. From a computational perspective, it is worth pointing out that the first stage of image generation is carried out by neural nodes that are endowed with functionalities akin to (but not identical with) those of bidirectional associative memories (Kosko, 1988).

In the second stage of image generation, more complex objects are formed by properly assembling the elementary features together. This latter process is governed by the symbolic module of ARCS, a system of production rules determining the features to be assembled together and their categorical spatial relationships in the complex recall image.

In addition to generating recall images by a two-stage process, ARCS can inspect and classify them. It achieves this goal using a pre-existing hybrid classifier for actual visual inputs. Thus, classification of both mental images and actual visual inputs is taken care of by the same process. This is consistent with the widespread conviction (Damasio and Damasio, 1994, Finke, 1985, Kosslyn, 1994) that visual imagery exploits the mechanisms of visual perception; more generally, that mental imagery, in any of its modalities (visual, auditory, tactile, etc.), exploits the mechanisms of same-modality perception. Yet another aspect of ARCS which is worth mentioning in this connection is the coarse internal organisation of the (recall) image classification process, as it closely reflects Kosslyn's protomodel of visual perception (Kosslyn, 1994, p. 69). In ARCS, shape and location data are handled by different processes and trigger classificatory hypothesis formation and the hypothesis-driven testing of proposed classifications by means of additional perceptual clues.

To sum up, the following claims concerning image generation and classification in ARCS seem, in our view, to capture the more relevant aspects of this computational system for cognitive modelling.

(i) The recall images generated by ARCS are non-crisp *examples* of the classes of visually presented features and objects the system is capable of classifying. (This claim is supported by (ii).)

(ii) The classification of recall images is correctly achieved by the same mechanism performing classification of actual visual inputs.

(iii) Image generation results from the composition of two distinct computational processes: simple visual features are first *generated* and then *assembled* into more complex patterns.

(iv) The coarse internal organisation of the image generation and classification mechanisms reflects distinguishing traits of current models of high-level vision.

These claims are more precisely specified in the next four sections, which describe the symbolic and neural components of the hybrid image classifier and generator. In the final section, the cognitively significant aspects of this system are rehearsed in the light of a more general discussion on neurosymbolic hybrid approaches to cognitive modelling.

RAM-DISCRIMINATORS AND CLASSIFICATION

In this section, we briefly describe the structure of RAM-discriminators, their training procedure, and the feature classification task performed by the multidiscriminator system of ARCS.

RAM-discriminators

A RAM-discriminator consists of a set of N one-bit word RAMs with X inputs and a summing device (Σ). Any such RAM-discriminator can receive a binary pattern of X·N bits as input. The RAM input lines are connected to the input pattern by means of a so-called "random mapping". The summing device enables this network of RAMs to exhibit — just like other artificial neural nets that more directly model features of biological neural networks — generalisation and noise tolerance. (See fig. 1 for a schematic representation of a particular RAM-discriminator.)

In order to train the discriminator one has to set to 0 the RAM memory locations and to choose a training set formed by binary patterns of X·N bits (see fig. 2 in which a possible training set for the feature *vertical line* is proposed). For any training pattern a 1 is stored in that memory location of each RAM which is addressed by this input pattern. Once the training is completed, the RAM memory contents will be set to a certain number of 0's and 1's.

The information stored by the RAM during the training phase is used to deal with previously unseen patterns. When one of these is given as input, the RAM memory contents addressed by the input pattern are read and summed by Σ. The number *r* thus obtained, which is called the discriminator *response*, is equal to the number of RAMs that output a 1. *r* reaches the maximum value N if the input pattern belongs to the training set (in the present example, if the input pattern is one of the patterns in fig. 2). *r* is equal to 0 if no three-bit component of the input pattern appears in the training set (no RAM outputs a 1). The other, intermediate values of *r* express some kind of "similarity measure" of the input pattern with respect to the patterns in the training set.

We selected RAM-discriminators as digital neural components for our system on the basis of the following considerations: their training algorithm can be easily modified as needed for image generation tasks; RAM-discriminators are tailored for efficient implementation on conventional computers; the use of artificial neurons more closely reflecting biological neurons would not make a difference at the coarse level of cognitive modelling sketched in the introduction.

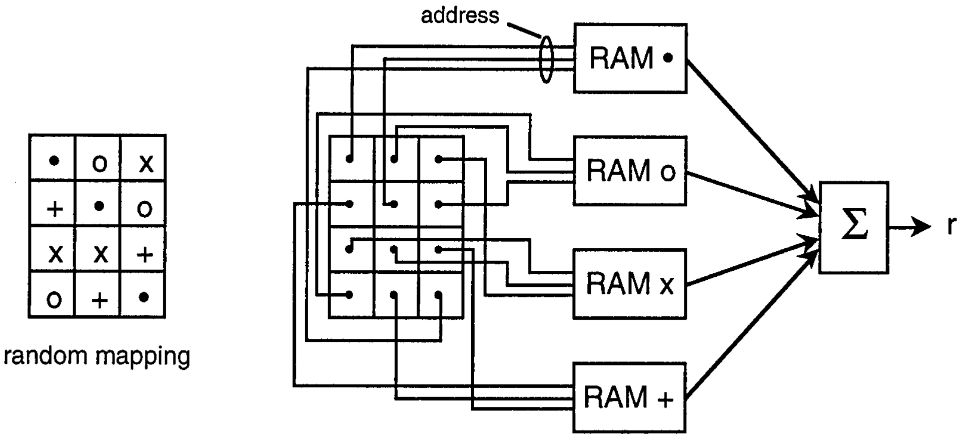


Figure 1 - RAM-discriminator

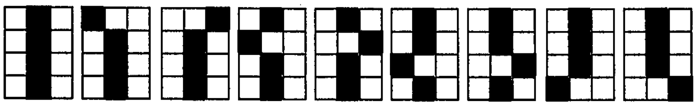


Figure 2: a possible training set for the feature *vertical line*

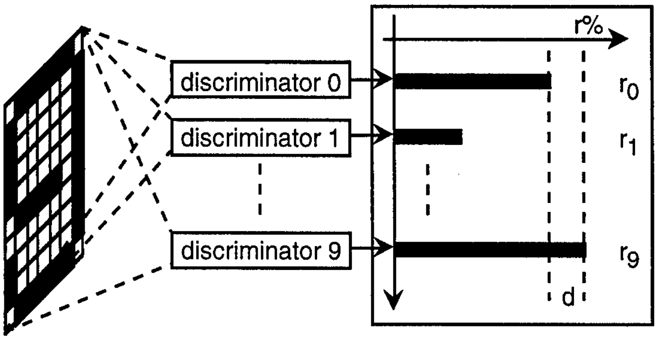


Figure 3 - a multi-discriminator system

Multidiscriminators

Multidiscriminator systems are formed by various RAM-discriminators (Aleksander *et al.*, 1984). Each discriminator is trained on a particular class of patterns, and classification by the overall multidiscriminator system is achieved in the following way.

When a pattern is given in input, (see fig. 3 for a schematic representation of a multidiscriminator system formed by 10 RAM-discriminators), each discriminator gives a response on that input. The various responses are evaluated by an algorithm which compares them and computes the relative confidence *c* of the highest response (that is, the difference *d* between the highest and the second highest response, divided by the highest response).

In ARCS, six discriminators were trained with drawings representing variations (in angle width, size, or position) on the simple geometric features of fig. 4. The discriminators are organised into a multidiscriminator system which ranks their responses.

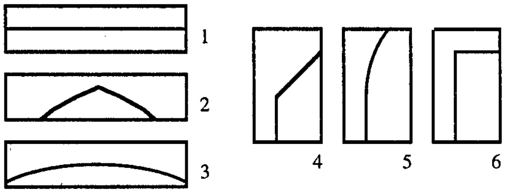


Figure 4 - geometric features

(MODIFIED) RAM-DISCRIMINATORS AND FEATURE GENERATION

RAM-discriminators were modified in what their memory locations may hold and, correspondingly, in their training algorithm. These changes, which produce something very similar to the PLN nodes introduced in (Aleksander, 1988), allow one to store *q*-bit words in memory locations (where *q* is usually not greater than 8); in turn, this information can be exploited for producing recall images (and improving in other ways the behaviour of RAM-discriminators).

Another training algorithm

The training algorithm of RAM-discriminators was changed in one respect only: instead of storing 1's, one just increases by 1 the memory location contents that are addressed by the input patterns. At the end of the training phase, the values of the memory contents will vary between 0 and M (where M is the number of training patterns). Fig. 5 shows the result of training the same RAM-discriminator of fig. 1 on the patterns of fig. 2, by means of the new algorithm.

The various memory content values can now be associated to subpattern frequency in the training set. For instance, the memory content of the address 010 associated to the +-th RAM is 5. This value indicates that the subpattern 010 is present 5 times in the training set of fig. 2. Moreover, one has to notice that the newly obtained memory contents do not give rise to different behaviours with respect to regularly trained RAM-discriminators, if one replaces the Σ device with another summing device, outputting the number of addressed memory locations whose content differs from 0.

One may take advantage of the new values stored in the RAMs in order to produce recall images (De Gregorio, 1997). This behaviour is significantly related (but not identical) to the exact input/output reversibility exhibited by the Bidirectional Associative Memories (BAM) introduced in (Kosko, 1988). The form of bidirectional behaviour we want to obtain from a RAM-discriminator D, trained with the new algorithm to pick out the elements of class X, must satisfy the following conditions:

(a) in one direction, D has to perform the usual classification process of RAM-discriminators;

	•	x	o	+
000	2	1	2	2
001	0	0	0	0
010	5	6	5	5
011	1	1	1	1
100	0	1	0	0
101	0	0	0	0
110	1	0	1	1
111	0	0	0	0

Figure 5 - RAM-discriminator of fig. 1 trained with the new algorithm

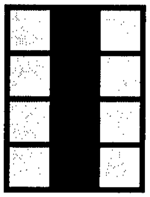
(b) in the opposite direction, D has to provide, when given the name of class X in input, an example of X.
It is not required that the example be identical to a member of the training set for D. Furthermore, we regard (b) as satisfied for any example of geometric feature in fig. 4 if the example is correctly classified by the multidiscriminator of ARCS.

The solution outlined here involves the construction of grey-level (rather than black and white) images exploiting the information held in the modified RAM memory locations. (A mathematical framework for approaching the reversibility problem for weightless systems is briefly sketched in (Redgers & Aleksander, 1992).)

Generating grey-level images

The procedure for constructing grey-level images is the following. Let b_1, b_2, and b_3 be the first, second, and third bit forming the address of a memory location (for instance, $b_1 = 0$, $b_2 = 1$ and $b_3 = 1$ represent the address of the 011 memory location). To each of these bits a particular pixel of the image is associated (see the mapping in fig. 1). For any RAM, let B_i, for $i = 1, 2, 3$, be the sum of all memory location contents for which b_i is 1 and the value stored is not equal to 0. For instance, for the •-discriminator in fig. 5 we obtain: $B_1 = 1$, $B_2 = 7$ and $B_3 = 1$. Applying this condition to every RAM in fig. 5 we obtain: $\forall j : j \in \{\bullet, \mathrm{x}, \mathrm{o}, +\}$, $B_{1j} = 1$, $B_{2j} = 7$, $B_{3j} = 1$. This regularity over the four RAMs depends on the fact that each pixel in the left-hand and right-hand columns of the matrix assumes value 1 (black) only once in the training set, whereas each pixel in the central column assumes value 1 (black) seven times in the training set.

Now, one can set the grey intensity level of each pixel associated to the bit b_{ij} in such a way that it is proportional to the corresponding value B_{ij}: the higher is B_{ij} the darker will be its grey intensity level. The result of this procedure applied to the modified RAM-discriminator trained for the feature *vertical line* is shown in fig. 6.



1	7	1
1	7	1
1	7	1
1	7	1

Figure 6 - the generated example of vertical line

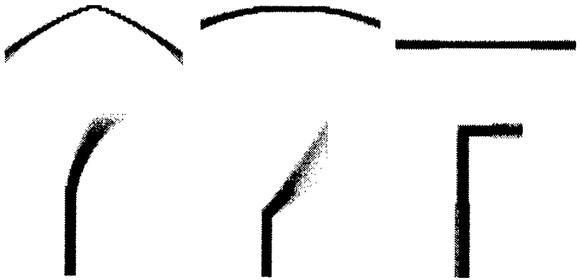


Figure 7 - examples generated by the modified multi-discriminator

Let us now turn to consider the wider class of simple visual features exemplified in fig. 5. In recognising these features, the multidiscriminator system of ARCS (trained by the modified algorithm described in this section) works in a canonical way, i.e. just as any regular multidiscriminator system. Moreover, the system may also provide, upon request, an example of each geometric feature it can classify. The results are shown in fig. 7.

CLASSIFYING COMPLEX PATTERNS

ARCS was originally conceived for classifying actual photographs of portal shapes into one of the classes *a* to *f* exemplified in fig. 8. The hybrid approach was pursued after direct classification through a multi-discriminator system failed.

The purely neural approach

The training set employed in the first, purely neural approach contained several drawings varying from the examples in fig. 8 only in the way of their position and size. The results obtained in a test made with 85 actual photographs of portals showed that only portal shapes belonging to classes *a* and *b* were correctly classified in a systematic way. The main reason for this failure emerges clearly from fig. 9, where the differences *a-b*, *e-f*, and *c-d* are shown.

While the relative complement of *b* in *a* is a rather large set of points, the other relative complements are much smaller and more localised. Thus, the information enabling one to discriminate between some such classes concerns the geometrical properties of small collections of points. It seems that spatial reasoning about geometrical features is crucially involved in this classification task. In particular, the more useful geometric cues are the top, the horizontal, and the vertical parts of portals, as exemplified in fig. 10 for polygonal portals. (Notice, however, that the horizontal parts are not essential for the non-linear portals *a*, *e*, and *f* in fig. 8.)

In order to mimic this geometrical reasoning capability, a hybrid system composed of a neural module and a symbolic module was adopted (De Gregorio, 1996): a

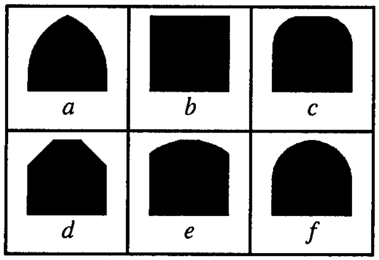


Figure 8 - classes of portal shapes

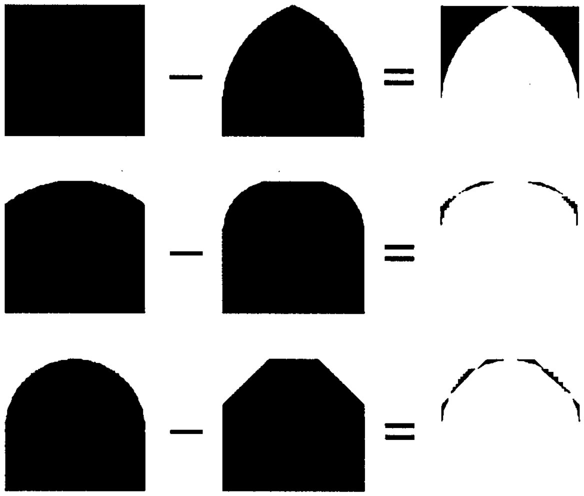


Figure 9

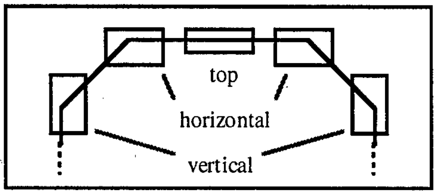


Figure 10

neural network for recognising geometric features was combined with a set of production rules specialised in classifying special arrangements of these features.

The symbolic module

In the symbolic module one can distinguish between three different sets of production rules.

The first set of rules enables the system to evaluate the geometric "coherence" of the discriminator responses. For instance, suppose that on the current input "straight angle" and "obtuse angle" are the best and the second best response for both left and right horizontal parts, respectively. Then, the system uses these rules to verify whether the left and right recognised features are almost at the same height, almost aligned with the top, symmetric with respect to the top. If the "straight angle" responses do not satisfy these conditions while the "obtuse angle" responses do, the system selects the "obtuse angle" ones as possible responses, because they are geometrically "coherent".

The second set rules implements an *abduction-prediction-test* inference cycle (Burattini and De Gregorio, 1994) which can be roughly described in the following terms. From the ranked list of responses for the top feature, which is provided by the multidiscriminator system, the best response is picked out to start the cycle. The system abduces the portal shapes (hypotheses) that are consistent with the higher-ranked top feature. Given these hypotheses on overall portal shape, the system predicts which horizontal features may be detected, and activates the appropriate discriminators. Then, if one of these horizontal features is actually detected, the associated hypothesis is selected for further scrutiny, and the system activates the relevant discriminator to test again that hypothesis with respect to the vertical features; otherwise, the cycle is repeated on the next hypothesis (with the obvious termination conditions).

The third set of rules is formed by six rules, one for each portal shape, and enables the system to infer the final portal classification (if any). For example, the rule concerning polygonal portals can be informally stated as follows:

(R_p) If top feature is in class no. 1 of fig. 4 and the horizontal and vertical features are in (or are obtainable by 90° rotation or specular reflection from instances of) class no. 4, then portal shape is polygonal (as in fig. 8, d).

The hybrid classifier has correctly classified the 85 actual photographs of portals that showed the inadequacy of the previously attempted, one-step neural classification approach (see fig. 11 for an input image - left - which is filtered - centre - and eventually classified - right; the higher the responses the darker the lines).

GENERATING AND CLASSIFYING RECALL IMAGES OF COMPLEX PATTERNS

We have pointed out that the third set of production rules of the symbolic module enables the system to infer portal classification from portal components. By exchanging condition and action parts of these rules, one obtains new rules specifying which parts are to be assembled together to form the recall image of a given portal shape. For example, from rule (R_p), one obtains a rule which can be informally stated as follows:

(R_p<–) If portal shape is polygonal (as in fig. 8, d) then top feature is in class no. 1 of fig. 4 and the horizontal and vertical features are in (or are obtainable, by 90° rotation or specular reflection, from instances of) class no. 4.

Similarly, by exchanging condition and action parts of the "geometrical coherence" rules, we obtain rules determining categorical spatial relationships between parts of complex recall images (ruling that vertical and horizontal components must be aligned and symmetrical with respect to the top, that the horizontal components must be aligned with the top).

An assembly problem which is not solved in the current implementation is how to determine the proportions of the components and their metric spatial relationships. A dynamic solution would require the recording of the corresponding data during the classification of actual visual inputs — in order to set, and then progressively refine the values of such spatial relationships as new examples are being classified. We have not yet implemented a process of this sort, which takes dynamically into account all past experience of the system. Currently, the system assigns fixed default values to such relationships.

The grey-level recall images obtained by means of the "assembly" rules are correctly recognised by the classifier of ARCS (see fig. 12). As with the actual photographs of portals given in input to the system, these recall images are first processed by a grey-level filter, which produces a binary image sharpening the non-crisp input image. The latter is given in input to the hybrid classifier described in

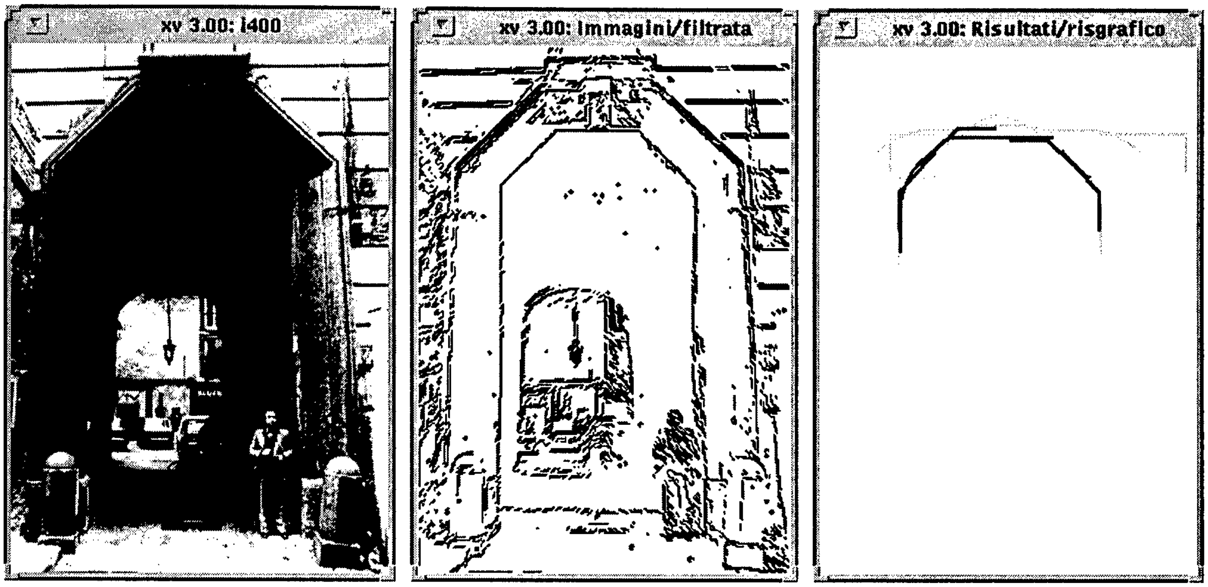


Figure 11 - input image of a polygonal portal (left), filtered (centre) and classified by the multidiscriminator (right).

the previous section. (Let us notice, in passing, that the existence of a filter editing visual information, by filling in missing details or sharpening fuzzy data is postulated in various cognitive models. See, for instance, Kosslyn (1994, p.389) and Rocha (1997, p. 157).)

Fig. 12 shows the grey-level recall image of a polygonal portal (left) which is filtered into a black and white image (centre). The responses of the discriminators for the various features (right) are represented in different intensities of grey: darker lines correspond to higher response values.

Is a recall image obtained in this way a *typical* example of the corresponding class? To the extent that frequency and typicality can be assimilated, the (filtered) recall images might be regarded as typical pictorial representatives of their classes. The non-crisp recall image preserves in its darker, more noticeable parts a trace of the more frequently encountered patterns during the training phase. In the filtered recall image (see fig. 12, centre), the sharpened trace induced by such patterns stands out even more clearly.

Under the hypothesis that frequency and typicality are identifiable in the domain under consideration, one can assert that the various classes of portals in fig. 8 are pictorially represented in the system by means of recall images, whereas the more abstract, general class *portal* can only be represented by a disjunction of statements such as (R_P). (See Ullman (1996, p. 184) for a recent discussion of related issues.)

In concluding this section let us notice that (complex) recall images may be adjusted on the basis of further experience. If, during a new training session, the memory location contents of the various RAMs are modified, then the grey-intensity level of the associated pixels in the generated example will change accordingly.

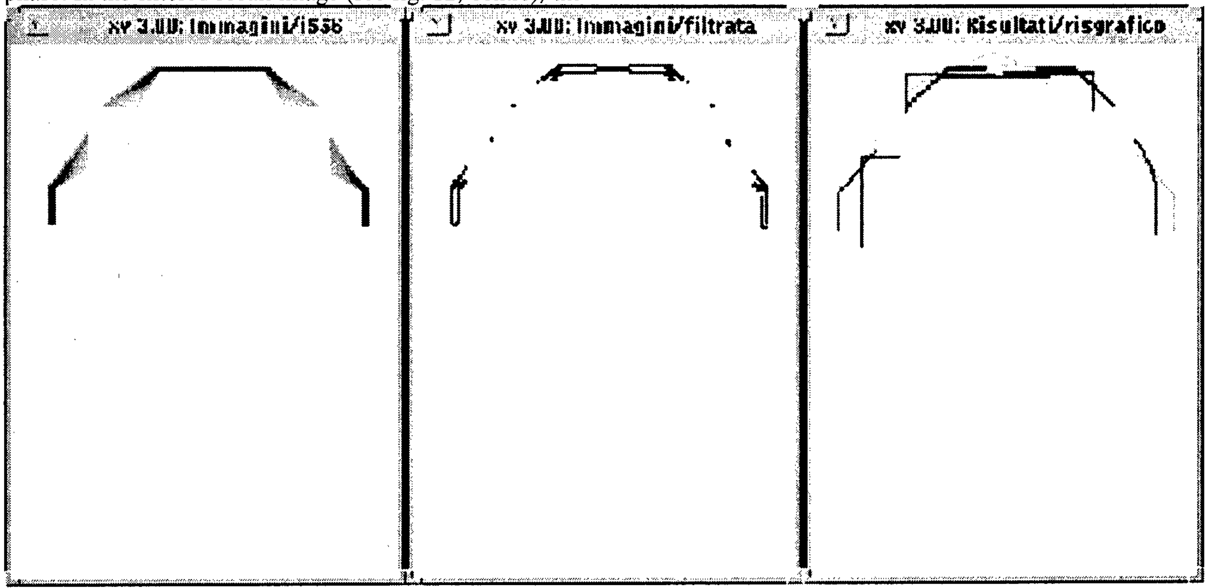


Figure 12 - recall image of a polygonal portal (left), filtered (centre) and processed by the multidiscriminator (right).

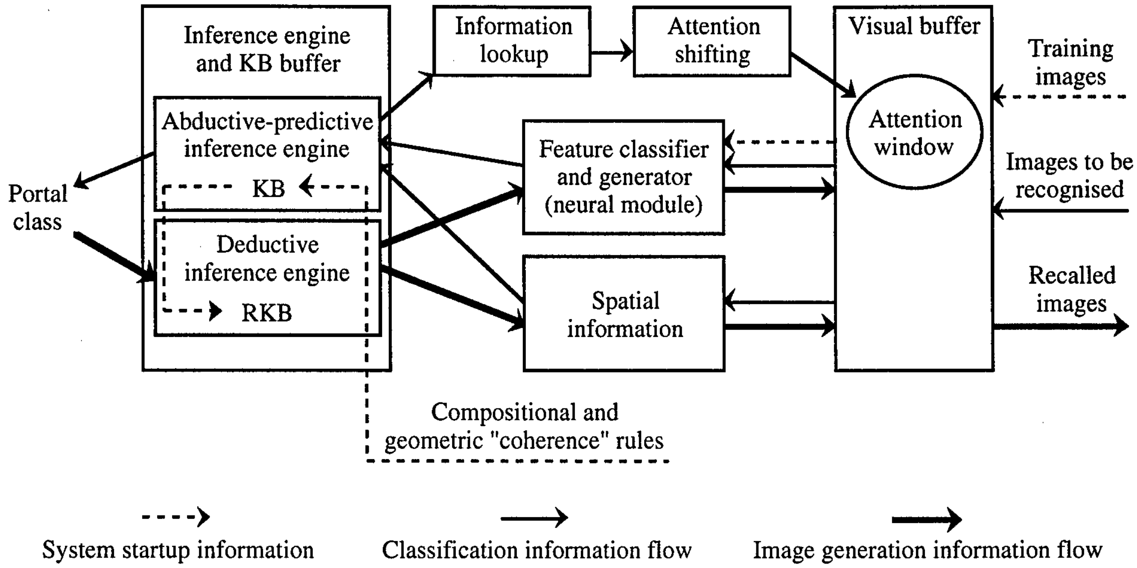

Inference engine and KB buffer
Abductive-predictive inference engine
KB
Deductive inference engine
RKB
Portal class
Information lookup
Attention shifting
Visual buffer
Attention window
Training images
Images to be recognised
Recalled images
Feature classifier and generator (neural module)
Spatial information
Compositional and geometric "coherence" rules
System startup information
Classification information flow
Image generation information flow

Figure 13 - system layout (RKB stands for Reverse Knowledge Base).

NEUROSYMBOLIC HYBRID SYSTEMS FOR COGNITIVE MODELLING

The idea that visual perception and imagery share the same underlying mechanisms is supported by various behavioural, neuropsychological and more recently accrued brain imaging data. ARCS reflects this hypothesis, insofar as visual object classification and generation of recall images in the task domain are concerned. Notably, classification of both actual and recall images is performed by exactly the same mechanism, whereas just a reversing of some neural and symbolic mechanisms involved in classification is needed to generate recall images. This reversing might be viewed as contributing to an abstract, purely functional modelling of reentrant neuronal connections (Edelman, 1989, p. 195, Damasio & Damasio, 1994), which combines the two quite different computational techniques of production rules and weightless neural systems, that are usually associated to the symbolic and subsymbolic paradigm, respectively.

Neural and symbolic techniques converge into hybrid neurosymbolic approaches to cognitive modelling (Hilario, 1995). These approaches seem to provide appropriate tools for modelling the interaction between top-down reasoning processes (typically simulated by symbolic computation) and bottom-up perceptual processes (often simulated by neural computation). The hybrid system ARCS is a case in point, since classification is accomplished by the interaction of both types of processes. We were unable to carry out this task by means of the purely neural, one-step classification process that was previously adopted.

Also the generation of recall images in ARCS is not a one-step process. The neural module generates examples of the various classes of features. Categorical spatial information about the position of features is represented in the symbolic module, and enables the system to assemble individually generated features into a complex recall image. Thus, the partition of the system into a neural and a symbolic component corresponds to a decomposition into functionally distinct subsystems, which is postulated in Kosslyn's model of imagery (Kosslyn, 1994 and 1995). A more detailed system layout is shown in fig. 13.

In general, the cognitive models taking the form of hybrid systems may raise a special epistemological problem since, in principle, model and cognitive reality might be compared at the symbolic, subsymbolic, and even neuronal levels. The problem does not arise if the hybrid system is proposed as a coarse model of input/output behaviours, relative to the major components of the cognitive system only. At this level of comparison, it is immaterial whether the various components are implemented as a neural net or as a symbolic system: only the functions computed by each component are relevant. Claims (i)-(iv) — made in the introduction and relating ARCS to the computational modelling of aspects of high-level vision and visual imagery — are to be understood just at this functional level of comparison. Thus, the choice of a hybrid architecture for ARCS is only pragmatically motivated: it allows one to simulate cognitive functions that seem to elude symbolic (respectively, neural) computation techniques in isolation.

These various observations suggest that the hybrid approach pursued here is not in principle incompatible with later developments possibly allowing one to substitute a neural module for a symbolic module, *salva* functional equivalence, and eventually transforming a hybrid system into a *unified* neurosymbolic system. In such unified systems, neurally implemented modules perform symbolic reasoning, too (Hilario, 1995). In principle, ARCS may be transformed into a unified system, by neurally implementing the production rule system performing geometric reasoning (adopting, e.g., the methodology proposed in (Aiello *et al.*, 1997, Aiello *et al.*, 1995)). It is not obvious, however, that every hybrid system can be turned into a unified system. There are forms of reasoning that currently go beyond unified approaches; these include most forms of classical reasoning in first-order logic, which lack (efficient) neural implementations. For discussion, see (Aiello *et al.*, 1997, Ajjanagadde & Shastri, 1993, Sun, 1994).

Finally, some remarks about future work on recall images in problem solving. One may endow the system ARCS with an explanation module combining words and pictures. If a user wants to know why a given house portal was classified as, say, polygonal, the system may justify this conclusion roughly as follows:

(a) it generates and displays examples of the visual features detected in the input image;

(b) it verbally declares and visually exemplifies the spatial relationships that were recognised to hold between the displayed features;

(c) it displays an example of the overall portal shape, obtained by properly arranging the generated features, next to the input image.

Similar uses of multiple representations have been recently discussed in (Tabachneck-Schijf *et al.*, 1997). Both in ARCS and in other contexts that we are currently exploring (concerning simple 2-D geometric figures), recall images may be used to complete partially occluded pictures, so that the various completions that the system declares as consistent with the occluded image can be shown.

ACKNOWLEDGMENTS

We wish to thank Carlo Arcelli for helpful comments on an earlier version. One of the authors (MDG) is deeply grateful to Igor Aleksander who introduced him to the fascinating world of weightless systems.

REFERENCES

Aiello, A., Burattini, E., & Tamburrini, G. (1997). Neural nets and rule-based reasoning. In C.D. Leondes (ed.), *Fuzzy Logic and Expert Systems Applications*, Academic Press, Boston MA.

Aiello, A., Burattini, E., & Tamburrini, G. (1995). Purely neural, rule-based diagnostic systems. Part I. Production rules. *Int. J. of Intelligent Systems 10*, 735-749.

Ajjanagadde, V., & Shastri, L. (1993). From simple associations to systematic reasoning. *Behavioral and Brain Sciences 16*, 417-494.

Aleksander, I. (1988). Logical Connectionist Systems. In Eckmiller R. & von der Malsburg C. (eds.), *Neural Computers*, Springer Verlag, Berlin 189-197.

Aleksander, I., & Morton, E. (1990). *An Introduction to Neural Computing*. Chapman & Hall, London.

Aleksander, I., Thomas, W., & Bowden, P. (1984). WISARD, a radical new step forward in image recognition. *Sensor Review 4*, 29-40.

Burattini, E. (ed.) (1994). *Intelligenza Artificiale e Recupero Edilizio*, Istituto di Cibernetica CNR, I-80072 Arco Felice, Italy.

Burattini E., & De Gregorio, M. (1994). Qualitative Abduction and Prediction. Regularities over Various Expert Domains. *Information and Decision Technologies 19*, 471-481.

Damasio, A.R., & Damasio, H. (1994). Cortical systems for retrieval of concrete knowledge: the convergence zone framework. In C. Koch & J.L. Davis (eds.), *Large-Scale Neuronal Theories of the Brain*, 61-74. MIT Press, Cambridge MA.

De Gregorio, M. (1996). Integrating inference and neural classification in a hybrid system for recognition tasks. *Mathware & Soft Computing 3*, 271-279.

De Gregorio, M. (1997). On the reversibility of multi-discriminator systems. Technical report 125/97, Istituto di Cibernetica CNR, I-80072 Arco Felice, Italy.

Edelman, G.M. (1989). *The Remembered Present*. Basic Books, New York NY.

Farah, M.J., Gazzaniga, M.S., Holtzman, J.D., & Kosslyn, S.M. (1985). A left hemisphere basis for visual imagery? *Neuropsychologia 23*, 115-118.

Finke, R.A. (1985). Theories relating mental imagery to perception. *Psychological Bulletin 98*, 236-259.

Hilario, M. (1995). An overview of strategies for neurosymbolic integration. In *Connectionist-Symbolic integration: from unifies to hybrid approaches*, Working Notes, IJCAI '95, Montréal, 19-20 August, 1995.

Kosko, B. (1988). Bidirectional Associative Memories, *IEEE Transactions on Systems, Man, and Cybernetics 18*, 49-60.

Kosslyn, S.M. (1994). *Image and Brain*. MIT Press, Cambridge MA.

Kosslyn, S.M. (1995). Mental imagery. In D.N. Osherson & S.M. Kosslyn (eds.) *Visual Cognition*, vol. 2 of *An invitation to Cognitive Science*, 267-296. MIT Press, Cambridge MA.

Redgers, A., & Aleksander, I. (1992). Digital neural networks. In Warwick K. *et al.* (eds.), *Neural Networks for Control and Systems*, Peter Peregrinus, London, 13-30.

Rocha, A.F. (1997). The brain as a symbol-processing machine. *Progress in Neurobiology 53*, 121-198.

Sun, R. (1994). *Integrating Rules and Connectionism for Robust Commonsense Reasoning*. John Wiley, New York NY.

Tabachneck-Schijf, H.J.M., Leonardo, A.M., & Simon, H.A. (1997). CaMeRa: a computational model of multiple representations. *Cognitive Science* **21**, 305-350.

Ullman, S. (1996). *High-Level Vision*. MIT Press, Cambridge MA.

Influence of Mapping on Analog Access: A Simulation Experiment with AMBR

Alexander A. Petrov*

*New Bulgarian University
Department of Cognitive Science
21, Montevideo Str.
Sofia 1635, Bulgaria
apetrov@cogs.nbu.acad.bg

Boicho N. Kokinov#*

#Institute of Mathematics and Informatics
Bulgarian Academy of Science
Bl. 8 Acad. G. Bonchev Str.
Sofia 1113, Bulgaria
kokinov@cogs.nbu.acad.bg

ABSTRACT

This paper contrasts two views about the relationship between the processes of access and mapping in analogy-making. According to the modular view, analog access and mapping are two separate 'phases' that run sequentially and relatively independently. The interactionist view assumes that they are interdependent subprocesses that run in parallel. The paper argues in favor of the second view and presents a simulation experiment demonstrating its advantages. The experiment is performed with the computational model AMBR and illustrates one particular way in which the subprocess of mapping can influence the subprocess of access.

KEYWORDS

Analogy-making, interactionist approach, access, mapping, simulation experiment, hybrid cognitive architecture.

INTRODUCTION

A crucial point in analogy-making is the retrieval of a base (or source) analog. Accessing an appropriate base from the vast pool of episodes stored in the long-term memory is not only a logical necessity (one cannot make analogies without a source) but apparently is the most difficult and capricious element of analogy-making. Starting with the classical experiments of Gick and Holyoak (1980) it has been repeatedly demonstrated that people have difficulties in spontaneously accessing a base analog, especially when its domain is very different from that of the target problem. In the aforementioned study only about 20% of the subjects were able to solve the so-called radiation problem even though an analogous problem (with solution) was presented shortly before the test phase. When provided by an explicit hint to use this source analog, however, 75% of the subjects achieved the solution. This great difference between the two experimental conditions was attributed to the difficulty of analog access.

On the other hand, we know a lot of stories about great scientists making discoveries by spontaneously using remote analogies. We have also personal experience in everyday usage of remote analogies. A recent study by Wharton, Holyoak, and Lange (1996) has demonstrated that about 35% of their subjects were successfully reminded about a remote analog story studied 7 days earlier when cued by the target story. (They have used a directed reminding task, not a problem solving task, however.)

Researchers of analogical access have become interested in the features of a remote analog that facilitate retrieval. Most data in the field (Holyoak and Koh, 1987, Ross 1989) suggest that analogical access is almost exclusively guided by superficial semantic similarities between base and target—similar objects and relations, similar themes, similar story lines, etc. In contrast, analogical mapping is dominated by the structural similarity between target and base, i.e. having common systems of relations (Gentner, 1983, 1989). This explains why remote analogs are much more difficult to access than to map—they lack the superficial similarities needed for access but do have the (quasi)isomorphic relational structure necessary for mapping.

This clear separation stimulated the researchers in the field to build separate models of mapping and retrieval and even to claim that they are different cognitive modules. Thus Gentner (1989) claims that 'the analogy processor (the mapping machine) is a well-defined separate cognitive module whose results interact with other processes, analogous to the way some natural language models have postulated semi-autonomous interacting subsystems for syntax, semantics, and pragmatics.' Although she explicitly mentions in a footnote that this should not be considered in the Fodorian sense as innate and impenetrable, the actual models built are quite impenetrable. This line of research has generated a number of quite successful models that explained the data and made some new predictions. Typically, a model of mapping is coupled with a (separate) model of retrieval. The best-known examples are SME + MAC/FAC (Falkenhainer, Forbus, and Gentner, 1986; Forbus, Gentner, and Law, 1995) and ACME + ARCS (Holyoak and Thagard, 1989; Thagard, Holyoak, Nelson, and Gochfeld, 1990).

However, the experimental work soon revealed that the pattern is not that clear and straightforward. It has been demonstrated that superficial similarities do play an important role in mapping as well. In particular cross-mapping is difficult (Ross, 1989). This led Holyoak and Thagard to include syntactic, semantic, and pragmatic constraints in their model of mapping ACME (Holyoak & Thagard, 1989) and to develop their multi-constraint theory (Holyoak & Thagard, 1995).

There are also some indications that structural similarity might play a role in access as well. Thus Ross (1989) demonstrated that in some cases (when the general story line is similar) structural similarity plays a positive role in retrieval, while in other cases (when the general story line is dissimilar) it does not play any role or can even worsen the results. The results of Wharton, Holyoak, and Lange (1996) also support indirectly the hypothesis that structural correspondences might affect the access. This was reflected in the models being proposed. Both MAC/FAC and ARCS included a submodule of partial mapping in the module of retrieval, thus considering structural similarities at an early stage.

To sum up, the initial separation between retrieval and mapping was founded on their different psychological characteristics—semantic factors govern the retrieval, structural factors govern the mapping. Subsequent more precise experiments, however, cast doubt on this clear separation. These complications were accommodated by making patches to the original models. Finally, it was acknowledged that all kinds of constraints affected all phases of analogy-making, although to different extent (Holyoak & Thagard, 1995).

The experimental data themselves became more and more complex and controversial. These controversies can be explained in terms of more and more sophisticated classifications of the types of similarities involved in access and mapping (Ross, 1989; Ross & Kilbane, 1997). We argue, however, that these problems are resolved more parsimoniously by adopting a principally different view of analogy-making.

This resembles an episode of the history of astronomy. The geocentric system of Ptolemy started as a straightforward theory that described the observable movement of both stars and planets remarkably well[1]. As accuracy of measurement increased, however, discrepancies between theory and data crept in every now and then. It became routine for astronomers to deal with such 'anomalies' by adding more and more epicycles. But as time went on, it became evident that astronomy's complexity was increasing far more rapidly than its accuracy and that a discrepancy corrected in one place was likely to show up in another (Kuhn, 1970).

Back to the domain of analogy-making, most classical models assume sequential processing: *first* the retrieval process finds the base for analogy and *then* the mapping process builds the correspondences between the target and the retrieved base (Figure 1). Thus MAC/FAC+SME and ARCS+ACME are linear models separating retrieval and mapping in time and space. This view underlies most of the experimental work in the field as well. Researchers often contrast hint versus non-hint conditions in problem solving supposing that in the first case only mapping takes place, while in the second retrieval and mapping are running one after the other. However, as Ross (1989) has noted, even when explicitly hinted to use a certain analog subjects still must access the details of its representation. Another common experimental technique uses a memory task (typically recall) for studying access with the assumption that the same processes take place during analogical problem solving.

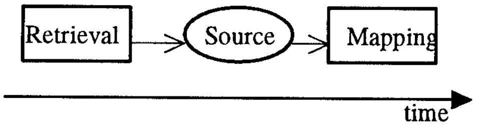


Figure 1. Dominating sequential models of analogy-making.

The limitations of both the models and experimental methods can be overcome by giving up the linearity assumption. This might look strange at first glance—how can you map the source analog onto the base if you have not even accessed it?! If, however, one reconsiders one more assumption—that there are centralized representations of situations/problems in human memory—then it becomes clear that whenever we have partial retrieval of the base (having recalled a few details) we can start looking for corresponding elements in the target. This allows us to conceptualize access and mapping as parallel processes that can interact (Figure 2). In this paradigm, access and mapping refer not to phases or other behavioral steps, but rather to separate mechanisms that both play a role in selecting and activating a base and in finding the correspondences between base and target.

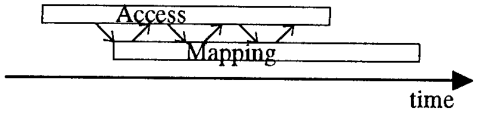


Figure 2. Parallel and interactive models of analogy-making.

The current paper explores the implications of the parallel and interactive view on access and mapping by running simulation experiments with an integrated model of human (analogical) reasoning called AMBR (Kokinov, 1994c, Petrov, 1997). These experiments provide a detailed example of how these two processes can interact and thus open space for new theoretical speculations as well as for new experimental paradigms. AMBR's predictions about the development of the process over time call for appropriate experimental methods capturing the dynamics of human analogy-making—RT studies, think-aloud protocols, etc. Some of the controversies around the role of superficial and structural similarities in access and mapping 'phases' can now be expressed in terms of the interactions between the two mechanisms.

[1] It is still used today as an engineering approximation.

A very important contribution of the simulation is that it demonstrates how the supposedly later 'phase' of mapping can influence the supposedly earlier 'phase' of access. A detailed example shows how the access process develops over time and how it is influenced by the concurrent mapping process. This is contrasted with the case of isolated access. Different results are obtained in the two cases. These results correspond to the data of Ross and Sofka (unpublished) which main conclusions are summarized in (Ross, 1989) as follows: '... other work (Ross & Sofka, 1986) suggests the possibility that the retrieval may be greatly affected by the use. In particular, we found that subjects, whose task was to recall the details of an earlier example that the current test problem reminded them of, used the test problem not only as an initial reminder but throughout the recall. For instance, the test problem was used to probe for similar objects, and relations and to prompt recall of particular numbers from the earlier example. The retrieval of the earlier example appeared to be interleaved with its use because subjects were setting up correspondences between the earlier example and the test problem during the retrieval.' The simulation data presented in the current paper (obtained absolutely independently and based only on the theoretical assumptions of DUAL and AMBR) exhibit exactly the same pattern of interaction.

We must admit that even in a highly parallel and interactive model such as AMBR the effects of interactions are not predominating. In the majority of cases the independent work of the access mechanism might well yield the same results as the interaction between mapping and access described above. That is why the classical linear models of analogy have been successful and have contributed a lot to our understanding of human analogy-making. However, exactly the few exceptional cases that do provide different results in a parallel model are the more interesting and those who make the interpretation of the experimental data look controversial if analyzed in the spirit of the sequential models.

There are a few other models that advocate a parallel, overlapping, and interactive view on analogy—Copycat (Mitchell, 1993, Hofstadter, 1995), Tabletop (French, 1995, Hofstadter, 1995), and LISA (Hummel and Holyoak, 1997). However, Copycat and Tabletop do not model retrieval at all—they model the parallel work and interaction between perception/representation building and mapping. LISA also integrates access and mapping and performs them in parallel. Thus the mapping mechanism (connectionist learning in this case) influences the access. As a result, LISA could in principle demonstrate effects similar to those reported here.

BRIEF DESCRIPTION OF THE ARCHITECTURE DUAL AND THE MODEL AMBR

The basis for the simulation experiment discussed in this paper is a model called AMBR (Associative Memory-Based Reasoning). It is built on the cognitive architecture DUAL. Space limitations allow only an extremely sketchy description of DUAL and AMBR here. The interested reader is referred to earlier publications (Kokinov, 1988, 1994a,b,c; Petrov, 1997).

DUAL is a multi-agent cognitive architecture that supports dynamic emergent computation (Kokinov,Nikolov, and Petrov, 1996). All knowledge representation and information processing in the architecture is carried out by small entities called *DUAL agents*. Each DUAL-based system consists of a large number of them. There is no central executive in the architecture that controls its global operation. Instead, each individual agent is relatively simple and has access only to local information, interacting with a few neighboring agents. The overall behavior of the system emerges out of the collective activity of the whole population. This 'society of mind' (Minsky, 1986) provides a substrate for concurrent processing, interaction, and emergent computation.

Each DUAL agent is a hybrid entity that has symbolic and connectionist aspects (Kokinov 1994a,b,c). On the symbolic side, each agent 'stands for' something and is able to perform certain simple manipulations on symbols. On the connectionist side, it sends/receives activation to and from its immediate neighbors. Thus, we may adopt an alternative terminology and speak of *nodes* and *links* instead of *agents* and *interactions*. The population of agents may be conceptualized as a network of nodes.

The long-term memory of a DUAL-based system consists of the network of all agents in that system. The size of this network can be very large. Only a small fraction of it, however, may be active at any particular moment. The active subset of the long-term memory together with some temporary agents constitutes the *working memory (WM)* of the architecture. The mechanism of spreading activation plays a key role for controlling the size and the contents of the WM. There is a threshold that sets the minimal level of activation that must be obtained by an agent to enter the WM. There is also a spontaneous decay factor that pushes the activation levels back to zero. As the pattern of activation changes over time, some agents from the working memory fall back to dormancy, others are activated, etc. Only active agents may perform symbolic computation. Moreover, the speed of this computation depends on the level of activation of the respective agent. This makes the computation in DUAL dynamic and context-sensitive (Kokinov et al., 1996; Kokinov, 1994a,b,c). One particular consequence of this dynamic emergent nature of the architecture is that, although all micro-level processing is strictly deterministic, the macroscopic behavior of a DUAL system can be described only probabilistically.

The AMBR model takes advantage of these architectural features to account for some phenomena of human reasoning and in particular reasoning by analogy (Kokinov, 1988, 1994c). Again, due to space limitations we will consider only a small fraction of model's mechanisms.

Analog access in AMBR is done by means of spreading activation by the connectionist aspects of the DUAL agents. In particular, only few of the many episodes stored in the long-term memory are active during a run and only they are accessible for processing. The episodes or 'situations' have decentralized representations—it is not a single agent but a whole *coalition* that represents the elements of a situation and the relationships among them[2]. Therefore, it is possible that an episode is only partially accessed because only some of the agents have entered the WM.

The process of analogical mapping is done in AMBR by a combination of three mechanisms—marker passing, constraint satisfaction, and structure correspondence (Kokinov, 1994c; Petrov, 1997). The main idea is to build a *constraint satisfaction network (CSN)* to determine the mapping between two situations. This network consists of *hypothesis agents* representing tentative correspondences between two elements. Consistent hypotheses support, and incompatible ones inhibit each other.

This is similar to other models of analogy-making and notably ACME (Holyoak and Thagard, 1989). AMBR differs from the latter model, however, in several ways: (*i*) the CSN is constructed dynamically, (*ii*) only hypotheses that have some justification are created, (*iii*) the CSN is incorporated into the bigger working memory network, and (*iv*) there is no separate relaxation phase so there is a partial mapping at each moment.

The implication of these four points is that, unlike ACME and most other analogy models, the processes of access and mapping run in parallel and influence each other in AMBR. In other words, the model departs from the classical 'pipeline' paradigm and aims at a more interactive account of analogy making.

The influence between the two subprocesses in AMBR goes in both directions. The present paper concentrates on the 'backward' direction—from mapping to access. The next section describes a simulation experiment that sheds light on this kind of influence.

SIMULATION EXPERIMENT METHOD

We performed a simulation experiment to contrast the two ways of combining access and mapping—parallel vs. serial. The experiment also tested whether the AMBR model was capable to access a source analog out of a pool of episodes, and to map it onto a target situation.

Design

The experiment consisted of two conditions. Both conditions involved running the model on a target problem. In the 'parallel condition', AMBR operated in its normal manner with the mechanisms for access and mapping working in parallel. In the 'serial condition', the program was artificially forced to work serially—first to access and only then to map. The target problem and the content of the long-term memory were identical in all runs. The topics of interest fell into two categories—the final mapping constructed by the program and the dynamics of the underlying computation. The latter was monitored by recording a set of variables describing the internal state of the system at regular time intervals throughout each run.

Materials

The domain used in the experiment deals with simple tasks in a kitchen. The long-term memory of the model contains semantic and episodic knowledge about this domain. It has been coded by hand according to the representation scheme used in DUAL and AMBR (Kokinov, 1994c; Petrov, 1997). The total size of the knowledge base is about 250 agents. It states, for example, that water, milk, and tea are all liquids, that bottles are made of glass, and the relation 'on' is a special case of 'in-touch-with'. The LTM also stores the representations of eight situations related to heating and cooling liquids. Two of these eight situations are most important for the experiment and are described below together with the target problem.

Situation A: *There is a cup and some water in it. The cup is made of china. There is an immersion heater in the water. The immersion heater is hot. This state of affairs causes that the water is hot.*

Situation B: *There is a glass and an ice cube in it. The glass is made of [material] glass. The glass is in a refrigerator. The refrigerator is cold. This state of affairs causes that the ice cube is cold.*

Target problem (situation T): *There is a glass and some cola in it. The glass is made of [material] glass. There is an ice cube in the coca cola. The ice cube is cold. What is the consequence of this state of affairs?*

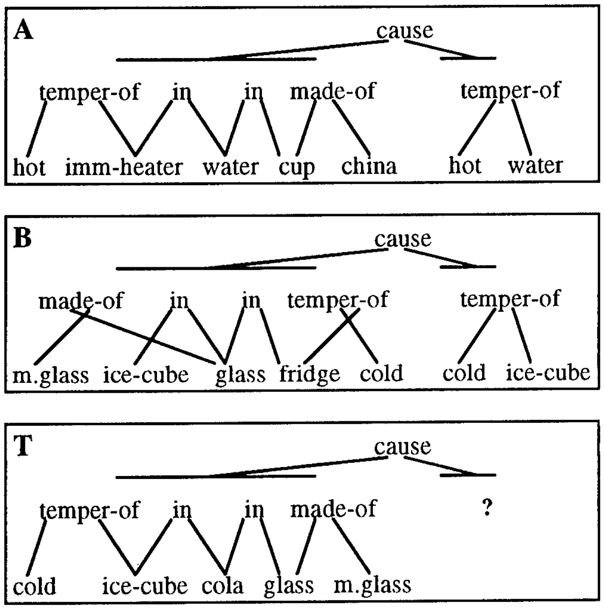


Figure 3. Simplified representations of situations **A**, **B**, and **T**. (The actual AMBR representations are more complex.) See text for details.

[2] This is one of the differences between the current version of the model (AMBR2) and the original proposal (AMBR1) as set forth by (Kokinov, 1994c).

As evident from Figure 3, both situations may be considered similar to the target problem. There are some differences, however. Situation **B** involves the same objects and relations as the target but the structure of the two are different. In contrast, situation **A** involves different objects but its system of relations is completely isomorphic to that of the target. According to Gentner (1989), the pair **A-T** may be classified as analogy while **B-T** as mere appearance. Thus it was expected that situation **B** would be easier to retrieve from the total pool of episodes stored in LTM. On the other hand, **A** would be more problematic to retrieve but once accessed it would support better mapping.

Procedure

The Common Lisp implementation of the AMBR model was run two times on the target problem. The two runs carried out the 'parallel' and the 'serial' conditions of the experiment, respectively. The contents of the long-term memory and the parameters of the model were identical in the two conditions.

Recall that situations have decentralized representations in AMBR. The target problem was represented by a coalition of 13 agents standing for the ice-cube, the glass, two instances of the relation 'in' and so on. 11 of these agents were attached to the special nodes that serve as activation sources in the model. This attachment was the same in the two experimental conditions.

In the parallel condition, the model was allowed to run according to its specification. That is, all AMBR mechanisms ran in parallel, interacting with one another. The program iterated until the system reached a resting state. A number of variables were recorded at regular intervals throughout the run. Out of these many variables, the so-called *retrieval index* is of special interest. It is computed for each situation and is based on the average activation level of the respective coalition. More concretely, the retrieval index is calculated by the formula:

$$\mathrm{RI}\,(t) = \frac{\sum a_i(t)}{0.5 + N} ,$$

where N is the total number of agents in the coalition and $a_i(t)$ is the activation level of agent_i at moment t.

In short, at the end of the run we had the final mapping constructed by the program as well as a log file of the retrieval indices of all eight situations from the LTM.

In the serial condition, the target problem was attached to the activation source in the same way and the same data were collected. However, the operation of the program was forcefully modified to separate the processes of access and mapping. To that end, the run was divided in two steps.

During step one, all mapping mechanisms in AMBR were manually switched off. Thus, spreading activation was the only mechanism that remained operational. It was allowed to work until the pattern of activation reached asymptote. The situation with the highest retrieval index was then identified. If we hypothesize a 'retrieval module', this is the situation that it would access from LTM.

After the source analog was picked up in this way, the experiment proceeded with step two. The mapping mechanism was switched back on again but it was allowed to work only on the source situation retrieved at step one. This situation was mapped to the target. Thus, at the end of the second run we had the final mapping constructed at step two, as well as two logs of the retrieval indices.

RESULTS AND DISCUSSION

In both experimental conditions the model settled in less than 150 time units and produced consistent mappings. By 'consistent' we mean that each element of the target problem was unambiguously mapped to an element from LTM and that all these corresponding elements belonged to one and the same base situation. Stated differently, the mappings were one-to-one and there were no blends between situations.

In the parallel condition, the target problem was mapped to situation **A**, revealing the isomorphism illustrated in Figure 4. One element from the source situation remained unmapped—the agent representing that the water becomes hot. This proposition is a good candidate for inference by analogy. *Mutatis mutandis*, it would yield the conclusion that the cola becomes cold. (In the current version of AMBR the mechanisms for analogical transfer are not implemented yet.)

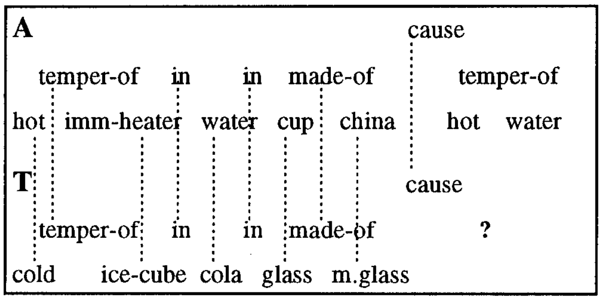


Figure 4. Correspondences constructed by the model in the parallel condition.

In the serial condition, situation **B** won the retrieval stage. This is explained by the high semantic similarity between its elements and those of the target—both deal with ice cubes in glasses, cold temperatures, etc. The asymptotic level of the retrieval index for **B** was more than three times greater than that of any other situation. In particular, situation **A** ended up with only 4 out of 14 agents passing the working memory threshold.

According to the experimental procedure, situation **B** was then mapped to the target during the second stage of the run. The correspondences that emerged during the latter stage mapped consistently the chains of two interlocking relations 'in' and the higher-order relation 'cause' (Figure 5). This structural alignment was achieved, however, at the expense of the semantic similarity between objects—the two glasses did not correspond, which in turn violated the structural constraint regarding the arguments of the relation 'made-of'.

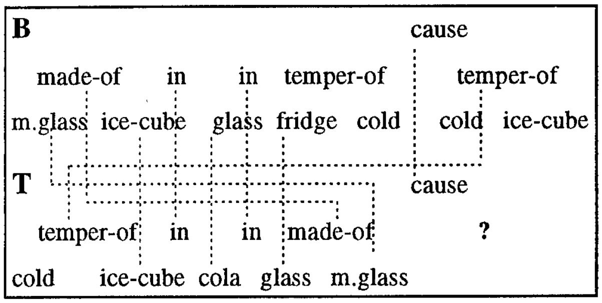


Figure 5. Correspondences constructed by the model in the serial condition.

It might be argued that these flaws of the mapping are not very serious, especially in the light of the structure mapping theory (Gentner, 1983). If we consider the material of the glasses an attribute, it is permissible to give it little attention. There is, however, a more serious flaw in the set of correspondences. The proposition 'temperature-of(ice-cube, cold)', which is a *premise* of the relation 'cause' in the target, is mapped to the proposition 'temperature-of(ice-cube, cold)', which is a *consequence* in the source. Therefore, the whole analogy between the target problem and the situation **B** could hardly generate any useful inference.

To summarize, when the mechanisms for access and mapping worked together, the model constructed an analogy that can potentially solve the problem. On the other hand, when the two mechanisms were separated, the retrieval stage favored a superficially similar but inappropriate base. The mapping stage then worked hard to produce an acceptable set of correspondences. Still, the final result was seriously flawed.

The presentation so far concentrated on the final result produced by the model. We now turn to the dynamics of the computation as revealed by the time course of the retrieval indices. Figure 6 plots the retrieval indices for several LTM episodes during the first run of the program (i.e. when access and mapping worked in parallel).

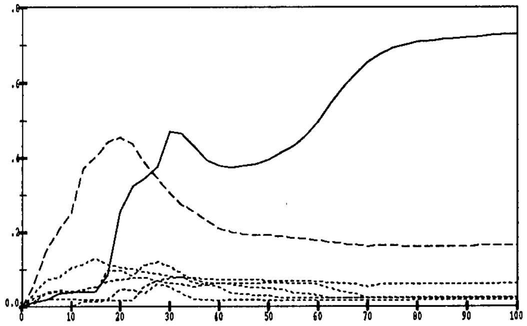


Figure 6. Plot of retrieval indices versus time for the parallel condition. Situation **A** is in solid line, **B** in dashed. The dotted lines at the bottom correspond to other situations from LTM.

This plot tells the following story: At the beginning of the run, several situations were probed tentatively by bringing a few elements from each into the working memory. Of this lot, **B** looked much more promising than any of its rivals as it had so many objects and relations in common with the target. Therefore, all agents representing situation **B** were rapidly activated and they began trying to establish correspondences between themselves and the target agents. The active members of the rival situations were doing the same thing, although with lower intensity. At about 15 time units since the beginning of the simulation, however, situation **A** (with the immersion heater) rapidly gained strength and eventually overtook the original leader. At time 30, it had already emerged as winner[3] and gradually strengthened its dominance.

The final victory of situation **A**, despite its lower semantic similarity compared to situation **B**, is due to the interaction between the mechanisms of access and mapping in AMBR. More precisely, in this particular case it is the mapping that radically changes the course of access. To illustrate the importance of this influence, Figure 7 contrasts the retrieval indices with and without mapping.

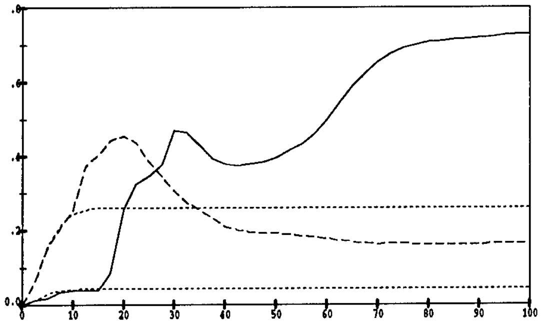


Figure 7. Retrieval indices for situations A and B with and without mapping influence on access. See text for details.

The dotted lines in Figure 7 show the retrieval indices for the two situations when mapping mechanisms are suppressed. Thus, they indicate the 'pure' retrieval index of each situation—the value that is due to the access mechanism alone. The index for situation **B** is much higher than that of **A** and, therefore, **B** was used as source when the mapping was allowed to run only after the access had finished.

In the interactive condition, however, the mapping mechanism boosted the retrieval index via what we call a 'bootstrap cascade'. This cascade operates in AMBR in

[3] The 'hump' in the graph is a side effect of the mapping mechanism which is too complex to be detailed here. In a nutshell, it involves transforming 'embryo hypothesis agents' into 'mature hypothesis agents' (Petrov, 1997).

the following way. First, the access mechanism brings two or three agents of a given situation into the working memory. If the mapping mechanism then detects that these few agents can be plausibly mapped to some target elements, it constructs new correspondence nodes and links in the AMBR network. This creates new paths for the highly active target elements to activate their mates. The latter in turn can then activate their 'coalition partners', thus bringing a few more agents into the working memory and so on.

The bootstrap cascade is possible in AMBR due to two important characteristics of this model. First, situations have decentralized representations which may be accessed piece by piece. Second, AMBR is based on a parallel cognitive architecture which provides for concurrent operation of numerous interacting processes. Taken together, these two factors enable seamless integration of the subprocesses of access and mapping in analogy-making.

CONCLUSION

The simulation experiment reported in this paper provides a clear example of mapping influence on analog access and of the advantages of the parallel interactionist view on analogy-making. Furthermore, the computational model AMBR provides a theoretical framework for explaining the controversies in the psychological data on access and reminding. It is possible to explore in which cases the interaction between access and mapping produces results different from a sequential and independent processing. It provides also a framework for generating more precise hypotheses and new experimental designs for their testing. Thus, for example, the detailed logs of the running model might me used for comparison with protocols of think-aloud experiments.

Analogy-making has certainly no clear cut boundaries. Most literature has concentrated on explicit analogies, i.e. consciously retrieving an analog and noticing the analogy. However, there are other cases which might be called implicit or partial analogies, e.g. subconsciously accessing part of a previously solved problem and mapping it to part of the target description without consciously noticing the analogy. The decentralized representations of situations in AMBR make it possible to model the process of partial access, access with distortions, blending (Turner & Fauconnier, 1995), and interference. A previously solved problem can influence the course of problem solving in an even more subtle way by priming some concepts or situations which then trigger a particular solution (Kokinov, 1990, Schunn and Dunbar, 1996). The AMBR model can be used to analyze such cases. It has already been successfully applied for predicting priming and context effects (Kokinov, 1994c).

Priming effects are an example of the influence of access on mapping which is the opposite direction of the one discussed in the current paper. Order effects are another kind of effect that goes in 'forward' direction. Such effects may be due to non-simultaneous perception of the elements of the target problem (Keane, Ledgeway, & Duff, 1994) and/or non-simultaneous retrieval of relevant pieces of information from LTM. Thus the mutual influence between analog access and mapping offers many opportunities for investigation.

REFERENCES

Falkenhainer, B., Forbus, K., and Gentner, D. (1986). The structure-mapping engine. *Proceedings of the Fifth Annual Conference on Artificial Intelligence.* Los Altos, CA: Morgan Kaufman.

Forbus K., Gentner D., and Law, K (1995). MAC/FAC: A model of similarity-based retrieval. *Cognitive Science, 19,* 141-205.

French, R. (1995). *The subtlety of sameness: A theory and computer model of analogy-making.* Cambridge, MA: MIT Press.

Gentner, D. (1983). Structure-mapping: A theoretical framework for analogy. *Cognitive Science, 7,* 155-170.

Gentner, D. (1989). The mechanisms of analogical learning. In S. Vosniadou and A. Ortony (Eds.), *Simiarity and analogical reasoning.* New York, NY: Cambridge University Press.

Gick, M.L. and Holyoak, K.J. (1980). Analogical problem solving. *Cognitive Psychology 12* (80), 306-356.

Hofstadter, D. and the Fluid Analogies Research Group (1995). *Fluid concepts and creative analogies: Comuter models of the fundamental mechanisms of thought.* New York: Basic Books.

Holyoak K. and Koh K. (1987). Surface and structural similarity in analogical transfer. *Memory and Cognition, 15* (4), 332-340.

Holyoak K. and Thagard P. (1989). Analogical mapping by constraint satisfaction. *Cognitive Science, 13,* 295-355.

Holyoak, K. and Thagard, P. (1995). *Mental leaps: Analogy in creative thought.* Cambridge, MA: MIT Press.

Hummel, J. and Holyoak, K. (1997). Distributed representation of structure: A theory of analogical access and mapping. *Psychological Review, 104,* 427-466.

Keane, M., Ledgeway, K., and Duff, S. (1994). Constraints on analogical mapping: A comparison of three models. *Cognitive Science, 18,* 387-438.

Kokinov, B. (1988). Associative memory-based reasoning: How to represent and retrieve cases. In T. O'Shea and V. Sgurev (Eds.), *Artificial intelligence III: Methodology, systems, applications.* Amsterdam: Elsevier.

Kokinov, B. (1990). Associative memory-based reasoning: Some experimental results. *Proceedings of the Twelfth Annual Conference of the Cognitive Science Society.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Kokinov, B. (1994a). The context-sensitive cognitive architecture DUAL. *Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society.* Hillsdale,NJ: Lawrence Erlbaum Associates.

Kokinov, B. (1994b). The DUAL cognitive architecture: A hybrid multi-agent approach. *Proceedings of the Eleventh European Conference of Artificial Intelligence.* London: John Wiley & Sons, Ltd.

Kokinov, B. (1994c). A hybrid model of reasoning by Analogy. In K. Holyoak and J. Barnden (Eds.), *Advances in Connectionist and Neural Computation Theory. Vol. 2: Analogical Connections.* Norwood, NJ: Ablex Publishing Corp.

Kokinov,B., Nikolov,V., and Petrov,A. (1996). Dynamics of emergent computation in DUAL. In A. Ramsay (Ed.), *Artificial Intelligence: Methodology, Systems, Applications.* Amsterdam: IOS Press.

Kuhn, T. S. (1970). *The Structure of Scientific Revolutions* (second ed., enlarged). Chicago: The University of Chicago Press. (First edition published 1962.)

Minsky, M. (1986). *The society of mind.* New York: Simon and Schuster.

Mitchell, M. (1993). *Analogy-making as perception: A computer model.* Cambridge, MA: MIT Press.

Petrov, A. (1997). *Extensions of DUAL and AMBR.* M.Sc. Thesis. New Bulgarian University, Cognitive Science Department.

Ross, B. (1989). Distinguishing types of superficial similarities: Different effects on the access and use of earlier problems. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 15,* 456-468.

Ross, B. and Kilbane, M. (1997). Effects of principle explanation and superficial similarity on analogical mapping in problem solving. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 23* (2), 427-440.

Ross, B. and Sofka, M. (1986). [Remindings: Noticing, remembering, and using specific knowledge of earlier problems]. Unpublished manuscript.

Schunn, C. and Dunbar, K. (1996). Priming, analogy, and awareness in complex reasoning. *Memory and Cognition, 24,* 271-284.

Thagard, P., Holyoak, K., Nelson, G., and Gochfeld, D. (1990). Analog retrieval by constraint satisfaction. *Artificial Intelligence, 46,* 259-310.

Turner, M. and Fauconnier, G. (1995). Conceptual integration and formal expression. *Metaphor and Symbolic Activity, 10* (3), 183-204.

Wharton, C., Holyoak, K., and Lange, T. (1996). Remote analogical reminding. *Memory and Cognition, 24* (5), 629-643.

Modelling the interpretation of verbal commands with fuzzy logic and semantic networks

Christophe Brouard
Bernadette Bouchon-Meunier
LIP6
Université Paris 6, Case 169
4 place Jussieu,
75252 Paris cedex 05, FRANCE
+33 1 44 27 70 03
Christophe.Brouard@lip6.fr
Bernadette.Bouchon@lip6.fr

Charles A. Tijus

Laboratoire de Psychologie Cognitive
Université Paris 8
2 rue de la liberté
93526 St Denis Cedex 02, FRANCE
+33 1 49 40 64 84
tijus@univ-paris8.fr

ABSTRACT

This study deals with the interpretation of verbal commands for action. After an experimental study of human interpretation of instructions for drawing geometrical figures, we have devised a model whose computerized version is called SIROCO. This model represents an attempt to simulate category construction for interpretation. The use of fuzzy logic and circumstantial semantic networks allows emphasizing the importance the situation plays in completing and clarifying propositions expressed in natural language. Finally, a simulation shows quite good results for the model.

Keywords

natural language, command, action, fuzzy logic, semantic network, situation, categorisation.

INTRODUCTION

When you look at the content of verbal commands, they appear to be poor, ambiguous and elliptic. Nevertheless, they are in fact efficient as measured by the fit of actions carried out by an operator to the speaker's (the person who formulates the command) intended goal. In summary, a few words are enough to elicit complex and precise actions. How can the power of utterances be explained ?

A partial explanation lies in the fact that the operator has mental models of situations, scenarios and procedures at his disposal. These comprise a general knowledge which allows him to complete the information received, to activate other knowledge in order to understand what is being asked of him and finally, to carry out the action. When, for example, someone is asked to post a letter, he knows that the letter needs a stamp, an address, and that it should be dropped in a mail box or taken to the post office. Modelling the operator, (here, the person asked to mail the letter) calls for describing and representing the kind of general knowledge we have just described. This is what a number of recent systems have attempted to do, including CARAMEL (Sabah & Briffault, 1993) for understanding stories, CAMILLE (Hasting & Lytinen, 1994) for describing scenarios, and KA (Peterson, Mahesh, & Goel, 1994) for technical specifications.

Pragmatic explanations might also be useful in explaining the power of utterances. Sperber and Wilson's communicational implications (1986) and Grice's maxims (1975) come to mind. Thus, in the above example, lacking any indication as to the cost of the stamps, the operator might rightly assume that the letter should be sent at a standard rate; because if it were to be sent express or recommended, this very relevant bit of information would surely have been provided. Modelling the operator thus calls for integrating pragmatic rules as well as general knowledge into the comprehension system. This is what has been done with DIABOLO, a system for analysing and generating dialogue (Vilnat, 1995).

The situated action approach[1] provides a more circumstantial way of explaining the efficiency of speech. The proponents of situated action place less emphasis on the notion of internal representation and more on situational cues and action. For Olson (1970), who rejects the linguistic approach to studying the comprehension of verbal utterances, the meaning of an utterance should not be looked for in the proposition, but in the situation to which the utterance refers. This is the approach we are taking here : the power of language resides in its relation to a given situation. Important clues that allow completing vague and elliptical utterances are provided by (i) the environment, (ii) the information that has already been communicated (what we will call the "background") and (iii) the task (what must be done with the elements provided by the environment).

We thus propose that a system for interpreting verbal commands must be able to cope with the incompleteness and the imprecision of language by analysing situations. The system we have devised to do so is called SIROCO. Though it is currently outfitted to interpret verbal commands for drawing geometrical figures, it could be adapted to interpret other kinds of verbal commands. We have used it to study how operators interpret commands and make decisions. In the case of incompleteness, the system has to identify the instructor's intended categories. In the case of imprecision, it has to define the fuzzy boundaries of the categories. To this end, we used two tools for representing information that is incomplete or imprecise, namely: circumstantial semantic networks and fuzzy subsets.

The study we present here was done in three phases: An experimental phase in which a human subject-operator was asked to interpret and carry out instructions for drawing geometrical figures given in natural language by a subject-instructor. The second phase consisted in designing a

[1] See Norman (1993), for an introduction to this situated action approach .

model of the subject-operator. Finally, a simulation allowed comparing SIROCO's responses to those of the subject-operator.

EXPERIMENT

Objectives

The aim of this experiment was to provide empirical data on the degree of precision with which people interpret verbal commands for drawing geometrical figures. More importantly, it aimed at providing information on how missing information is completed and, more generally, on how concrete situations influence the precision with which a command is carried out. All data relative to instructor commands and operator actions was collected automatically to provide a precise record of input and output for the simulation.

Method

Participants

Thirty five instructors were recruited from the undergraduate population of the University Paris 8, St Denis-Vincennes. A single operator was recruited from the same population, his responses provided the data we analysed.

Materials

A set of 35 drawings (8.2 cm large and 14.8 cm high), one for each instructor, were created with a drawing software. Each drawing was composed of three simple geometrical figures. The set was designed to provide a wide range of property combinations for the geometrical figures. The different figure-properties were: rectangle, circle and square, for the shape; red, green and blue, for the color; small, medium and large, for the size (from 1 cm up to 6.2 cm for width, from 0.6 cm up to 6.46 cm for height); top, center and bottom for the vertical position (from 0.01 cm up to 11.65 cm on the Y coordinate); and, finally, left, middle and right for the horizontal position (from 0.31 cm up to 5.98 cm on the X coordinate). The complexity of these combinations, from the point of view of the corresponding lattice (see next section), was maximal in all cases. In other words, any two geometric figures have both common and distinctive features.

The computer apparatus consisted of two large monitors placed back to back on a long table (fig. 1). Thus, the instructor and the operator, each behind a monitor, were hidden from each other. The instructor could only communicate through verbal commands, the operator could not see the original drawing the instructor had in his hand.

General procedure

Each one of the thirty five drawings was given to an instructor. The instructor was asked to make the operator reproduce this picture through verbal commands only. The operator, who was not allowed to speak to the instructor, typed each verbal command[2] he received into the word processor and then carried it out. The graphic interface on which the operator worked was of the same size as the instructor's picture. On his screen, the instructor saw what the operator was drawing. After the operator had finished carrying out a command, the instructor could correct the drawing with a new verbal command and so on, until the instructor was satisfied with the drawing the operator had produced.

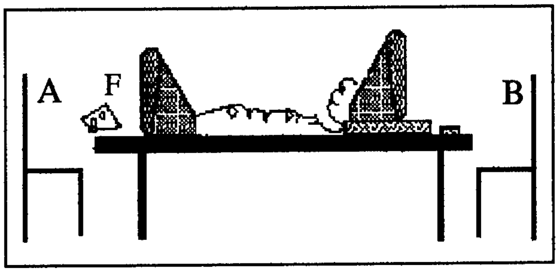


Figure 1. The instructor was placed in A and had in his hands a drawing (F). The operator was placed in B.

Automatic data collection

All action related to writing verbal commands (on the word processor) and drawing figures (on the graphic interface) was recorded with "spy" software.

Results and discussion

The Power of utterances

On average, 9 commands were necessary for a satisfactory reproduction of the original drawing. The minimum was 4, the maximun was 18 for a single drawing. On average, 3 commands were required for reproducing each figure. More precisely, 2 commands were sufficient to correct the first attempt to draw a figure. This may seem very few when one considers that there were four continuous factors which defined each figure (size, shape, vertical and horizontal position).

The precision of commands for discrete properties

There were just a few lateralisation errors ("not on the left, I said on the right"). Though information on size and color was not always given, these were correctly reproduced by the operator. The semantic structure of the properties of the figures already in place (see MODEL section) did allow completing the missing property information in a command. Thus, our hypothesis was globally satisfied. Nevertheless, the operator seemed to hesitate between an average value and a value inferred from properties already in place. The effect of the extracted regularity of the properties already in place would certainly have been greater if the objects were more numerous (in our experiment there are only three figures per drawing).

The precision of commands for continuous properties

From a statistical point of view, there were no significant differences between the figure-values for the continuous properties of the operator's finished drawings and the corrresponding values of the original drawings: for the X coordinates of the figure's top-left corner p>.96, for the Y coordinates (of the same point) p>.17, for width p>.94 and for height p>.08. The correlation between the operator's drawings and the originals one was .86 for the X coordinate (p<.0001), .93 for the Y coordinate (p<.0001), .70 for W (width) and .86 for H (height)

[2] Verbal commands were expressed in French. For the purpose of this article we translated some of them.

(p<.0001).The average correlation for each of the first seven commands is given in table 1.

Table 1: The average correlation for each of the first seven commands

	C1	C2	C3	C4	C5	C6	C7
X:	.75	.95	.70	.80	.91	.91	.91
Y:	.88	.99	.76	.97	.98	.94	.97
W:	.96	.76	.27	.55	.53	.96	.50
H:	.94	.93	.80	.50	.93	.97	.88

It is clear that the operator faithfully reproduced the original values quite rapidly, because from the first try on, the commands were executed with an overall precision of 4% for X, 1% for Y, 3 % for W and 2% for H. When the operator's figures did not fully correspond, a few more verbal commands were all that was needed to correct them. In summary, long-distance geometric figure drawing in this experiment was extremely precise.

MODEL

The results of the experiment show that the situation is indeed an aid in interpreting commands. The present model replicates the way in which the situation provides information by taking advantage of the dynamicity of circumstantial semantic networks and the flexibility of fuzzy subsets.

General Description

For SIROCO, interpreting verbal commands means using the situation to construct the instructor's intended categories. Thus, when one or more of a figure's properties is not explicitly indicated, it is inferred from the property network based on the figures that have already been drawn. Often property-categories are specified with absolute utterances such as "large" or "at the top" but sometimes compound utterances such as "rather square" or "smaller" are used. The meaning of compound utterances must be constructed from the meaning of the absolute ones. Indications and corrections given prior to a new command (the background) must also be taken into account. Finally, all of this information is represented in the form of fuzzy subsets and integrated through a procedure which aims at finding the solution that best satisfies all the constraints including space constraints.

Incompleteness processing with circumstantial semantic networks

The propositional meaning of an instruction is first analysed as to the objects and their associated properties. Subsequently, objects and properties are used in order to construct a semantic network which reflects an understanding of the proposition (Zibetti & Tijus, 1997; Poitrenaud, 1995).

In this network, properties shared by several objects are grouped together in order to constitute categories (figure 2). The underlying mathematical structure of this property network is the Galois lattice (Barbut & Monjardet, 1970). This network allows different logical operations. For example, if among different geometrical coloured figures, all the squares are black, it is possible to predict the black property from the square property because of the inherited properties of the square category in the semantic network. Otherwise, in certain context an object can be designated by a single property such as "the white one" in figure 2 to refer the white circle, or as "the black one" to refer the black circle because given that there are two black objects, the instructor might want to designate a figure that differs from others in the same category by being black.

Finally, the lattice allows some operations which can explain and simulate categorisation processes (Tijus & Moulin, 1997). For example, it is always a problem to categorise an incompletely described new object. A good solution (from the point of view of modelisation) consists in choosing or constructing a category that alter the structure of the network as little as possible. For example, if a white square has to be drawn, without any specification as to its size, in the situation described in figure 2, it will be small. More generally, this semantic network represents the property structure of a given situation which can be very useful for modelling (Richard, J.F., Poitrenaud, S., & Tijus, C.A., 1993).

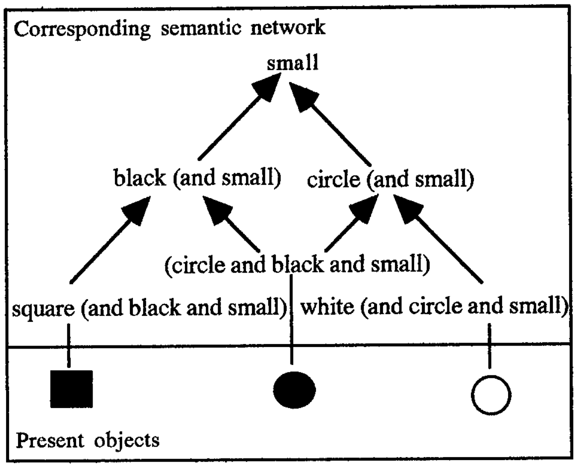

Corresponding semantic network
small
black (and small)
circle (and small)
(circle and black and small)
square (and black and small)
white (and circle and small)
Present objects

Figure 2. Example of a semantic network constructed from the object properties of the situation.

Representing a command with fuzzy subsets

A drawing command specifies size, shape, colour and position categories. Except for colour categories which are precisely defined (there is just one kind of blue, green and red available), the other kinds of categories (for example, large, rectangle) have imprecise boundaries. Thus an element (like a value corresponding to a surface in square centimetres) can have an intermediate degree of membership between 0 and 1 in a category. So we have chosen to represent these categories with a fuzzy subset. The concept of fuzzy subsets (Zadeh, 1965; Bouchon-Meunier, 1995) is a generalisation of the concept of sets. A fuzzy subset is characterised by its membership function (figure 3).

An important issue lies in the choice of reference variables (in figure 3, the choice is surface area as measured in square centimetres). This choice has to be made such that the variable is well suited to determine whether or not an element belongs to the represented category. Ideally, this variable has to correspond to a psychologically relevant perceptive dimension. Psychophysics, which studies the relations between physical and perceptive dimensions, could provide this kind of variable. As we are focusing upon general principles, we chose simple variables (like surface area for size categories, and abscissa for horizontal position) and trapezoidal fuzzy subsets.

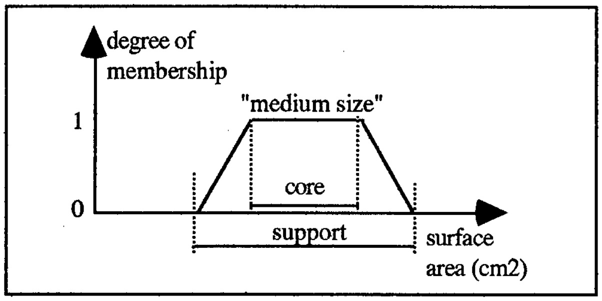


Figure 3. The membership function for a fuzzy subset representing "a medium size" category. Note that the "core" is comprised of elements which belong to the fuzzy subset with a membership degree equal to 1, and that the "support" is comprised of elements which belong to the fuzzy subset with a non-zero degree.

We represent a command by associating a fuzzy set to each dimension of the description. Zadeh (1975) introduced the concept of linguistic variables which consist of a variable, a universe in which the variable is defined (real numbers for example), and a set of fuzzy subsets which represent different characterisations of the variable (for example, small, medium and large for a size variable).

Here we use four linguistic variables to represent a command: (i) the size which is the surface area of the figure and which is characterised by "small", "medium" and "large", (ii) the elongation which is the width/height ratio and which is characterised by "upright" "equal" and "reclining", (iii) the horizontal position on the abscissa which is characterised by "left", "middle" and "right" and, finally, (iv) the vertical position which is on the ordinate and which is characterised by "top", "centre" and "bottom". Two discrete variables complete this representation: colour which can be blue, green or red and shape, which can be rectangular or elliptical.

Because there is an odd number of characterisations for each variable (exactly three), there is always a central category. Moreover the fuzzy subsets that represent these characterisations are such that they constitute a fuzzy partition of the universe. Which means that for each element, the sum of its membership degrees in all the different characterisations for a given variable is 1. Thus the slopes of the trapezia intersect at midpoint (see figure 4).

Applying a linguistic modifier

Our aim here is to represent utterances like "very large" or "toward the left", that is to say modified versions of categories. Zadeh (1972) associates to each linguistic modifier ("very", "rather",...) a mathematical transformation which allows constructing new fuzzy subsets from initial ones. The initial fuzzy subset represents an initial category ("large"). The new fuzzy subset represents a modified version ("very large") of the initial category.

Since Zadeh's pioneering work, numerous new modifiers have been introduced. Here, we use modifiers (Bouchon & Yao, 1992) which exploit the distribution of defined categories in a single universe (size, for example). The mathematical transformation corresponds to a shift whose amplitude and direction can be deduced automatically. We chose them for the way they can readily be applied to all different kinds of properties (see figure 4).

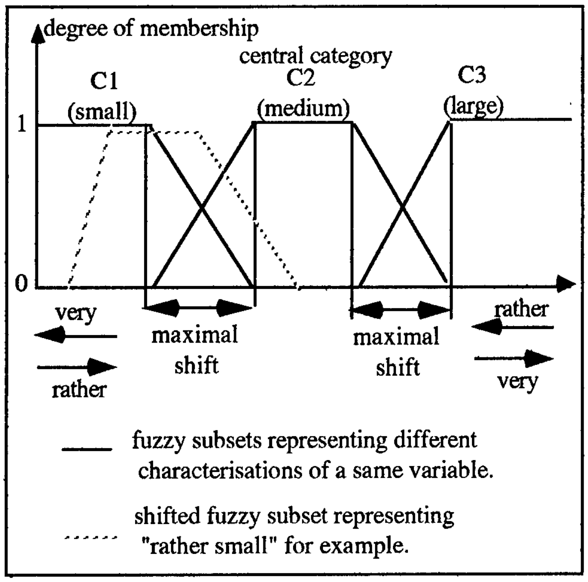

degree of membership
central category
C1
(small)
C2
(medium)
C3
(large)
1
0
very
rather
maximal shift
maximal shift
rather
very
fuzzy subsets representing different characterisations of a same variable.
shifted fuzzy subset representing "rather small" for example.

Figure 4. Illustration of linguistic modifier mechanisms.

From a given characterisation and a given modifier, simple mechanisms yield the shift to be applied. Thus, for modifiers like "very" the direction of the shift is toward an extreme and for modifiers like "rather" the direction is toward the centre (figure 4). The amplitude of the shift is defined as a proportion of the maximal shift which corresponds to the distance between initial category cores. Thus a modified category will never overlap upon a neighbouring category. Moreover, the maximal shift automatically defines a scale regardless of the type of variable. Finally, it is possible to use modifiers of different strengths. Thus "very very" is a modifier of the same kind as "very" but the amplitude of the shift associated to it is larger. To be more precise, the coefficient associated to "very very" is larger than the one associated to "very".

Applying a fuzzy relation

Utterances like "larger" or "a little bit less to the left" can be represented with fuzzy relations. The concept of fuzzy relations (Zadeh, 1971) is a generalisation of the concept of relation as it allows intermediate degrees (between 0 and 1) of relation between elements. It corresponds again to a fuzzy subset. In contrast to the preceding case concerning modifiers, this fuzzy set will not be constructed from a fuzzy set but from a value (the surface area of the figure, if the command is "larger"). We can divide this kind of command into two parts: the relation part which is, for example; "much more", "less" or "same" and a category part which is, for instance, "on the left" or "large".

It is possible to define mechanisms such that from a given relation and category, the fuzzy set representing the utterance can be constructed. First, after having defined a sign for a relation ("less" relations will be negative and "more" relations, positive), a category (positive categories are to the right of the central category, negatives are to the left) the direction (increasing, decreasing) indicated by the utterance (for example, "less big") is calculated by multiplying the relation sign and the category sign. When the category is the central one, the direction depends on the position value compared to the middle value of the category (when the command, for example, is "rounder", the question to be asked is whether the figure is an upright or a reclining ellipsis). Like modifiers , different coefficients are associated with each relation expressing a different strength. "Much more" indicates a stronger variation than "more" (figure 5).

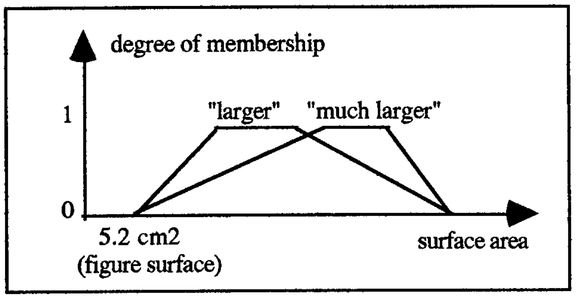


Figure 5. Illustration of fuzzy subset construction for utterances containing "larger" and "much larger" (than 5.2 cm2).

Softening inferred categories

As we mentioned above, when no characterisation is specified for a given variable in a drawing command, it is inferred from the semantic network. This tacit information is not as constraining as explicit information. We therefore chose to represent it by allowing all values, that is to say, by taking a support (for the constructed fuzzy subset) equal to the entire universe of the variable. Moreover, taking the results of experimentation into account we softened the inferred category by applying a modifier.

Background communication

At any point in a verbal exchange involving commands, what has already been said and done constitutes the background communication so decisive for interpretation. For example, what "larger" means can vary according to whether it is an initial correction whose aim is to get the operator to draw a figure of roughly the right size or whether it is a final correction aimed at precision. The background thus allows the commands to be interpreted with increasing precision. Indeed, without background, instructions like "a little bit larger" followed by "a little bit smaller" would consist of nothing more than commands for switching back and forth from an initial value.

Background construction

During communication, various indications and corrections are given. This can be represented by a list of slopes of different constructed trapezia in the prior commands. For each variable, there is one background. Fixing a maximal length for this list allows taking the operator's limited memory into account.

Making background operational

Only two slopes are useful for each variable. They correspond to the more restrictive constraints (right, left constraints could be for instance, respectively, much smaller than 10.3 square centimetres and larger than 5.4 square centimetres) and allow constructing a fuzzy set. So, background is accounted for by intersecting this last fuzzy subset with the current command associated to the fuzzy subset. When this intersection is small (under a given threshold), we can decide to forget the background in order to produce an appropriate response despite contradictory commands.

Choosing an appropriate solution

Choosing a relevant point

According to the specified position in the drawing command, the relevant point varies. For example, if the command calls for drawing a figure at the top left corner, the top left corner of the figure is the relevant point. Which means, that it is the point which will be taken into account for characterising the figure's position (in the example, the more the top-left corner of the figure is at the top and to the left, the more the position of the figure is acceptable). If the command is "To the left of the square, draw a ....", the right and centre (vertically) point is relevant.

From the 3 vertical position characterisations and the 3 horizontal position characterisations, we defined 9 relevant points. Relevant points allows simplifying the decision procedure. We could have chosen a more sophisticated variable that might have been more psychologically relevant, but as we are focusing upon general principles, we did not do so.

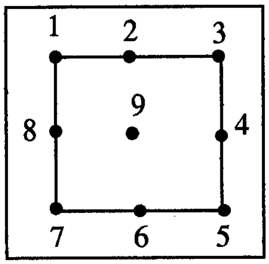


Figure 6. The nine possible relevant points of a figure.

Defining the degree of acceptability for all points of the drawing area.

For a point p of the drawing area, the acceptability degree is computed by aggregation of two intermediate degrees $d_1(p)$ and $d_2(p)$. We chose the min operator for expressing conjunctions:

$$d(p) = \min (d_1(p), d_2(p)),$$

where $d_1(p)$ indicates the degree to which point p (the relevant point) is a good point from which to begin drawing the figure specified in the command ("in the top-left corner" or "near the circle") and where $d_2(p)$ indicates the degree to which it is possible to place at p a figure of the size and shape corresponding respectively to the size and the shape of the characterisations of the command. It is computed as :

$d_2(p) = \sup\{(\min(\mu_{size}(l), \mu_{shape}(l,h)) / 0 < l < lmax, 0 < h < hmax(l)\}$,

where lmax and hmax are respectively the largest possible width and height taking into account the figures already present and μ_{size} and μ_{shape} are respectively the membership functions of the size and shape fuzzy subsets constructed from the command (figure 7).

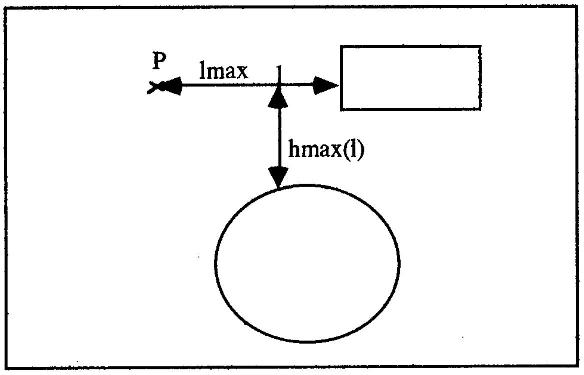


Figure 7. Illustration of the d2(p) calculation when p is the top-left corner of the figure.

Computing d(p) for all drawing-area points allows defining favourable areas (figure 8).

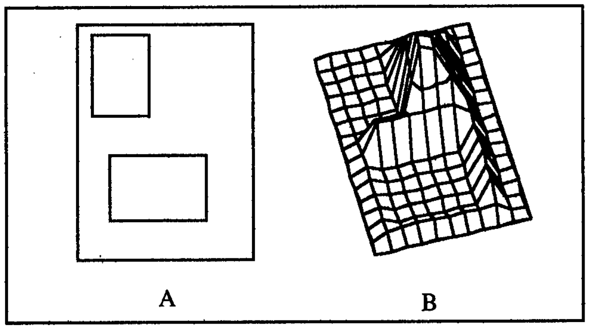


Figure 8. A) Figures already present.B) Visualisation of favourable areas for drawing "a large circle at the centre of the drawing area".

A solution suited to the situation
The general optimisation procedure allows choosing a solution suited to the situation without explicitly describing the situation beforehand (figure 9).

VALIDATION

The above model has been computerised and called SIROCO. This system allowed simulating the operator-subject in order to validate the model by comparing system responses to the operator responses.

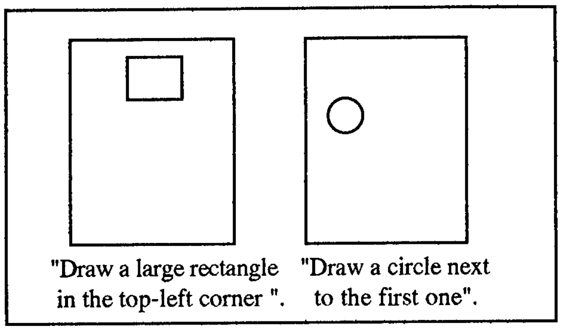

"Draw a large rectangle in the top-left corner ".
"Draw a circle next to the first one".

Figure 9. In these two situations, the optimisation procedures decide respectively to draw an upright rectangle and a circle to the right of the first one.

Model parametrization

The experiment provided thirty five communication records. Ten records were kept in order to test the model. The others were used for teaching the fuzzy subsets of the different characterisations, the modifiers and relation parameters to the system. More precisely, the first drawings for each communication (which correspond to a minimal context) allowed defining the cores for all characterisations. Supports were then defined in order to construct a fuzzy partition for each variable (see above). Analysing experimental results allowed defining modifier and relation coefficients. Relation coefficients express similarity, these similarity relations are not necessarily linear. For example, for an equal difference of surface, the smaller the two compared surfaces are, the more they are perceptively different. However, we considered these relations to be linear, and chose average coefficients because the experimental material did not allow inferring their exact shape[3].

Simulation

A description of SIROCO
Developed in C++, SIROCO includes a graphic interface for visualising system and subject drawing responses. It also allows running a commands file, typing commands interactively and readjusting the system's responses to the subject's responses at will. Finally, it allows visualising favourable drawing areas (by creating a matlab file).

Definition of a minimal language
The commands that were kept in order to test the model were translated into a minimal language with a limited number of words and with a strict structure[3]. Most of these words indicate the linguistic variable characterisations, and also, the modifiers and relations often used in commands. This language aims at representing commands without interpreting them. For example, "nearer the edge" is not translated as "more

[3] We should define, as in FILIP (Zemankova, 1989), these relations from the outside of the system.
[4]The commands which could not accurately be translated into the minimal language were excluded from the results. Variables and objects (like edges) need to be added to the language to make it more expressive. However our translation tables do allow expressing most of the commands.

toward the left" (if the figure is near the left edge) but by "more extreme" ("extreme" is automatically replaced with the object category, in this case, the "left"). Likewise, "make the rectangle longer" is translated by "more extreme". Thus, if the shape category is upright, the height will be increased. If the category is reclining, the width will be increased. We also use the OR connector in order to express utterances like "rectangle" (an upright rectangle OR a reclining rectangle) or "next to" (to the left OR to the right of another figure).

Simulation with readjustment

For this simulation, the operator-subject comparison was made command by command. Each of the system's responses was automatically readjusted to the subject's responses just before the next command was interpreted. The communication background was also readjusted. Thus, for each new command, the system was placed in precisely the same interpretive situation as the subject.

Results and discussion

In order to evaluate model validity, we compared the figures the human operator drew with the ones the system drew. More precisely, we compared the X and Y co-ordinates of the figure's top-left corner, the width W and the height H. We measured the error margin and the correlation for each of these variables.

Table 2: The average error margin in centimetres for the first seven commands.

X:	Y:	W:	H:
0.25	0.35	0.30	0.28

Compared to the figure variance in the initial drawings, there is no significant difference between subject and system drawings for the X coordinate (p>.2) and for width W (p>.32). On the other hand, we found differences for the Y coordinate (p=.02) and for height H (p <.01). Positions and sizes have a very important correlation: .93, .84, .93 and .92, respectively for X, Y, W, H. From the first to the third figure, the correlation is shown in table 3.

Table 3: The correlation from the first to the third figure.

X:	.94	.93	.91
Y:	.93	.98	.30
W:	.91	.90	.97
H:	.83	.90	.97

We can see that simulation becomes more and more precise as communication progresses (table 4).

Table 4: The correlation from the first to the seventh command.

	C1	C2	C3	C4	C5	C6	C7
X:	91	.94	.97	.97	.95	.92	.96
Y:	.86	.79	.8	.99	.99	1	1
W:	.35	.9	.97	.80	.86	.86	.99
H:	.67	.84	.93	.85	.95	.97	.99
av:	.70	.87	.92	.90	.94	.94	.98

The small error margin with which the system operated might be, but is not necessarily, due to the model. The system chose one solution from a set of equally possible solutions and, under the same conditions, a human operator might also give different responses. In order to explain these differences, we should add that the system, as opposed to a human subject, does not make mistakes and does not forget information. Overall, the response given by the system is always acceptable and it is difficult to distinguish it from the human operator's response. Otherwise, softening the category the system had infered from the semantic network also gave good results.

GENERAL DISCUSSION

A small number of combined cues are enough to enable us to define a precise solution. Other more elaborate experiments could reveal other important cues. Even in the particular case of this experiment, we do not pretend to have tackled all the facets of command interpretation. Category learning (Omri, 1994), that is to say the adjustment of interlocutor categories, is not taken into account here. Nonetheless, its affect would probably have been insignificant because communication between the operator and the instructor took place very quickly (the instructor was replaced for each new drawing).

As we mentioned in the introduction, our study is about a particular contextual explanation of the power of language. Thus, some implicits of communication were not taken into account, whereas their effects were not negligible from the point of view of the results. For example, when the command was to draw a figure on the left and there already was a figure on the left, the system chose to place the new figure very near the first one (it placed it as far to the left as it could). The implicit information in this command is that the two figures can not be stuck together, because if they were, this information would be given. To explain this kind of implicit principle of relevance introduced by Sperber and Wilson (1986) seems well suited. It could be implemented with semantic networks and fuzzy sets.

In summary, we have shown here a set of mechanisms for constructing the meaning of utterances from the basic category meanings. We have associated fuzzy subsets with semantic circumstantial networks and it appears that these representational tools are complementary as they cope with two different kinds of knowledge imperfection (imprecision and incompleteness) (Bouchon-Meunier, 1992). We could talk about "fuzzy semantic networks" even if category inclusion is not gradual as in Rossazza's networks (1992).

Unlike Hersh and Caramazza (1976), we are not only interested in representing the meaning utterances, we wanted to make it work, which is much more challenging. The method we follow, first, determination of fuzzy meaning for a set of variables, and second, definition of a solution maximizing the satisfaction degree of all variable constraints and integrating all environnement constraints, seems well adapted to model action. Compared with a rule system where the rules have to cover all situations and have to be explicited, this method appears more adaptative and more simple to implement, the main work consisting in constructing variables.

CONCLUSION

The aim of this interdisciplinary study was double. On the one hand, our goal was to model the processes of command interpretation (through cognitive psychology) and on the other hand, it was to create a system capable of responding consistently to verbal commands, of detecting implicit information and of adapting itself to a given situation (through artificial intelligence). These two aspects of the study are by no means opposed because devising a system that models a human subject has every chance of being a system whose behaviour is adequate. This is all the more true given that verbal communication is a specifically human activity.

There already exist certain mobile remote control apparatuses, equipped with a camera, for inspecting places that humans, for one reason or another, cannot enter. The operator who controls the apparatus must constantly specify the angle and speed at which the apparatus moves. Though the interface may be user friendly and, for instance, allow guiding the apparatus with a joystick rather than explicitly indicating angle and speed, there are still disadvantages. Namely, the constant supervision that the system requires calls for technical mastery as well as taxing levels of alertness and watchfulness on the part of the human operator. These disadvantages could be partially compensated for by redesigning the system to respond to natural language.

AKNOWLEDGEMENTS

This research was funded by DGA/DRET Contract N° 93-166. The authors express their thanks to E. Hamilton for her assistance in research and translating.

REFERENCES

Barbut, M., & Monjardet, B. (1970). Ordre et classification: algèbre et combinatoire. Paris: Hachette.

Bouchon-Meunier, B. & Yao, J. (1992). Linguistic modifiers and gradual membership to a category. International Journal on Intelligent Systems, 7, 25-36.

Bouchon-Meunier, B. (1995). La logique floue et ses applications. Paris: Addison-Wesley France.

Bouchon-Meunier, B. (1992). Représentation et traitement de l'incertitude, Journées Science et Défense, La Villette.

Grice, H.P. (1975). Logic and conversation. In P.Cole and J.L. Morgan (Eds.). Syntax and semantics, Vol 3: Speech acts, New York: Academic Press.

Hasting, P.M., & Lytinen, S.L. (1994). Objects, Actions, Nouns, and Verbs. In Ashwin Ram & Kurt Eiselt (Eds.). Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society.(pp. 397-402). Hillsdale, NJ: LEA

Hersh, H. M. & Caramazza, A. (1976). A Fuzzy Set Approach to Modifiers and Vagueness in Natural Language. Journal of Experimental Psychology: General, Vol 105, 3, 254-276.

Norman, D. A. (1993). Cognition in the head and in the world : an introduction to the special issue on situated action. Cognitive Science, 17, 1-6.

Olson, D. R. (1970). Language and Thought: Aspects Of a Cognitive Theory Of Semantics. Psychological Review, 4, 257-273.

Omri, M-N. (1994). Système interactif flou d'aide à l'utilisation de dispositifs techniques : le système SIFADE. Thèse, Université Paris VI.

Peterson, J, Mahesh, K., & Goel, K. (1994). Situating natural language understanding within experience-based design. International Journal of Human-Computer Studies, 41, 881-913.

Poitrenaud, S. (1995). The Procope Semantic Network : an alternative to action grammars. International Journal of Human-Computer Studies, 42, 31-69.

Richard, J.F., Poitrenaud, S., & Tijus C.A. (1993). Problem-solving restructuration: elimination of implicit constraints. Cognitive Science, 17, 497-529.

Rossazza. J-P. (1992). Une représentation centrée objet. Deuxième Congrès National sur les Applications des Ensembles Flous, Nîmes.

Sabah, G., & Briffault (1993). CARAMEL: step towards reflexion in natural language understanding systems. Actes IEEE International Conference on Tools with Artificial Intelligence, Boston, 258-265.

Sperber, D. & Wilson, D. (1986). Relevance : Communication and cognition. Cambridge, MA: Harvard University Press.

Tijus, C.A., & Moulin, F. (1997). L'assignification de signification à partir de textes d'histoires drôles. L'Année psychologique, 33-75.

Vilnat, A. (1995). STANDIA: A pragmatic-driven Man-Machine System, the structure of multi-modal dialogue. Amsterdam: North-Holland.

Zadeh, L.A. (1965). Fuzzy sets. Information and Control , 8, 338-353.

Zadeh, L.A. (1971). Similarity relations and Fuzzy Orderings, Information Sciences, 3, 177-200.

Zadeh, L.A. (1972). A Fuzzy-Set Theoretic Interpretation of Linguistic Hedges, Journal of Cybernetics, 2, 2, 4-34.

Zadeh, L. A. (1975). The concept of a linguistic variable and its application to approximate reasoning. Information Sciences, 8, 199-249, 301-357.

Zemankova (1989), FILIP : a Fuzzy Intelligent Information System with Learning Capabilities, Information Systems, 14, 6, pp. 473-486.

Zibetti, E., & Tijus, C.A. (1997). l'effet des propriétés d'objet sur l'interprétation de l'action. Actes du Colloque Sciences Cognitives JIOSC 97: La perception: du naturel à l'artificiel. Orsay: 1er et 2 décembre 1997.

Chess Players' Early Recall of Chess Positions: An Empirical and Simulative Investigation

Pertti Saariluoma
Cognitive Science
University of Helsinki
P.O. BOX 13, Helsinki, Finland
+358 9 191 23458
psa@utu.fi

Tei Laine
Cognitive Science
University of Helsinki
P.O. BOX 13, Helsinki, Finland
tei@iki.fi

ABSTRACT

Novice acquisition of skilled recall of chess positions was studied in an experiment in which two novices studied a series of five hundred chess positions during a period of several months. They spent fifteen minutes to half an hour a day teaching themselves chess positions. As a result their skills in recalling chess positions rose from an average sixteen percent to somewhere between forty to fifty percent. The learning curve proved to be a logarithmic function in which learning is very fast at first but after some 100-150 studied positions the speed of learning decreases substantially.

A computer simulation was used to model the results. Two alternative ways of thinking were tested. In the first model chunk construction was assumed to be based on neighbourhood of associated pieces. The second model assumed a frequency based correlative association process. Although the learning curves of the two models are very similar by shape to that of the subjects, the frequency based associative model gave better explanation for the data. This is why it is natural to suggest that common co-occurrence is one mechanism in associative processes during chess players learning of chess specific chunks.

Keywords

Cognitive modeling, novice skill acquisition, chunking

INTRODUCTION

One can argue that research on chess players' memories is relevant only when the top level skills are considered. When all the basic skills training is focused on people that are very far from having ten years experience in the field, it should be interesting to investigate what are the major properties relevant to early learning in chess. The first hours of chess training are close to any basic course in some symbolic subject matter. Therefore, it would be good to pay more attention to these earliest stages of information processing. An important shift to the direction of early learning was made by Fisk and Lloyd (1988) when they studied the acquisition of skilled visual search in chess with absolute novices.

Fisk and Lloyd's (1988) study showed that a skill develops very rapidly at first, but later the speed of learning decreases substantially. By studying some later stage of skill development, they could not have made this observation (see also Newell and Rosenbloom 1981, and Rosenbloom and Newell 1987 for parallel findings in different task environments). If a similar pattern of skill development to the one concerning the reaction time results of Fisk and Lloyd (1988) could be found in chess recall task, it might explain why the development of skilled memory takes so much time. Though it is easy to achieve one level, each new step takes more and more effort.

To help resolve the problems above, two students with only very elementary knowledge in chess were asked to study hundreds chess positions ten to twenty minutes a day for four to six months in order to recall the positions as well as they can. The development of their recall was tested several times during this period. The aim was to determine the form of the learning curve for a later simulation analysis.

By using computer simulation we wanted to study the nature of the chunking mechanisms in early learning. Chase and Simon (1973) suggested that a number of chess specific relations such as colour, kind, threat, defense, and proximity are important in chunk construction. Here, we are interested in an even simpler factor. This is general associativity, and a good approach to it is to use a simple correlative measure. If the pieces that commonly co-occur are used in building new chunks (the idea that general associativity is important), one should get the best fitting simulation outcome by chunking pieces with high correlation.

In the next section the settings and the results of experiments in which two novice chess players learnt and tried to recall game and random chess positions are described. In the following sections a computer simulation designed to model the experimental conditions and results are presented and analyzed.

METHOD

Subjects

Two graduate psychology students participated in the experiment. One was a woman, NT, who had played only few games of chess in her life. The second subject was a man, MQ, with the same background, who had played chess a little more often, but he had neither chess ambitions nor qualifications. Neither of the subjects had ever visited a chess club or participated in a chess competition. Both were thus absolute novices.

Task and Procedure

NT, the first subject, was asked to study five hundred middle game positions from a book of combinations. She studied four to five positions for approximately fifteen minutes a day. When studying the positions she put the pieces on the board according to the illustration and tried to learn the location of each piece. She concentrated, however, only on the patterns and did not study the moves at all. She was tested five times: before the experiment began, after 110, 250, 365, and 500 positions. Her involvement in the whole experiment lasted about four months.

The experiment involving MQ was made a couple of months after the end of the experiment with NT. This second experiment took six months as MQ wanted to spend more time per a position than NT. He also studied five hundred middle-game positions from a book of middle game combinations and his method of study was the same as NT's. It was possible to test MQ somewhat more frequently than NT. His recall was tested eight times: at the beginning, after 30, 60, 175, 220, 270, 350 and 500 studied positions. The irregularities in testing intervals were due to certain practical problems involved in running this long experiment such as compulsory exams, Christmas leave, etc. Each test consisted of a standard de Groot (1965, 1966) experiment. Subjects were shown ten game and ten random positions with 18 to 28 pieces in each. The presentation time was five seconds per position and the presentation order was random. The positions presented in the various testing sessions were always different.

The test positions were made by using chess print transfers which were then photographed as slides. They were shown with a slide projector. The subjects sat at a distance of 150 cm from a display. The size of chess boards on the display was 40 × 40 cm.

RESULTS AND DISCUSSION

The results of the experiment are presented in figure 1. The x-coordinate represents the number of studied positions and the y-coordinate the percentage of recalled pieces in a test session. The percentages are the mean percentages of correctly placed pieces calculated for each test session.

The effect of learning is clear. The subjects were able to increase their percentage of recalled pieces from roughly fifteen to somewhere between forty and fifty percent, which was a rise of 25-35 percentage points. However, in recalling random positions the effect was substantially smaller averaging about five percentage.

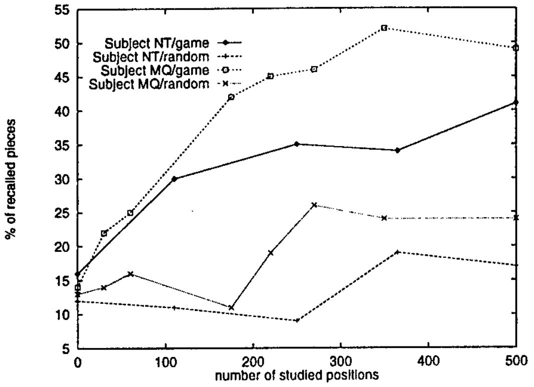


Figure 1: Percentage of recalled pieces when the subjects were tested 5 or 8 times during learning.

The profile of learning curve was very similar for both subjects. When studying the first hundred to one hundred and fifty positions they achieved most of the total increase in recall percentages. The increase was far slower from two hundred positions onwards. There was also some increase in recalling random positions, but the profile is very different from game positions, the increase being more linear throughout the whole learning period.

The learning curves of NT and MQ have a standard form. They are like many other learning curves: at first it is very steep reflecting a sharp improvement in learning. However, after a short period of time the speed decreases and the gain in performance level becomes smaller per training unit (Fisk and Lloyd 1988, Newell and Rosenbloom 1981, Rosenbloom and Newell 1987). This kind of curve can be called negatively accelerating or logarithmic.

SIMULATIVE ANALYSES

The results of the experiment provide very rough information. As such they do not tell very much about chunking as a method of learning. However, by using a computer simulation it would be possible to associate the previous experimental data with some other information about chess players' chunking. Thus the present experiment may be used for a theoretical discussion of some aspects of chess players' information chunking and to estimate the number of chunks needed for very high performance.

Several properties of chunks and chunking in chess which should be taken into account in any attempt to model chess players' recall, have been noted during the last twenty five years. The main function of chunking is to avoid the capacity limitations of human working memory. The more and larger the chunks a chess player has in his long term memory, the greater the probability of him being able to avoid the limits of his working memory and achieve a high level of performance.

This should not, however, be interpreted so that the chess players' working memory is seen as a box with a few slots. The reason for capacity should rather be sought in the integration of information as the chess positions are stored in the long-term memory rather than in working memory (Charness 1976, Frey and Adesman 1976, Lane and Robertson 1979, Lories 1987, Simon 1976).

Chase and Simon (1973) observed that skilled chess players do not necessarily recall more chunks than novices, but the sizes of their chunks are larger. The increase in chunk size is not an all-or-nothing phenomenon but it rather takes place incrementally. Good players do not learn new and longer chunks at one time but their chunks slowly lengthen and their recall improves (Chase and Simon 1973, Newell and Simon 1972). This is also the way chunk learning is assumed to occur by some theories of cognitive skills (Newell and Rosenbloom 1981, Rosenbloom and Newell 1987).

In addition to chunk size the factors behind the coherence of chunks have deemed important. Chase and Simon (1973) found five chess-specific relations (same kind or colour, threat, defense and adjacent locations on board) that increase the probability of successive pieces to belong in the same chunk. Another issue affecting the recall ability of a chunk is its location, not just the chess-specific relations between the pieces in it (Saariluoma 1984). It is easy to find strongly associated piece patterns in random positions, but they are very seldom correctly located. Finally, the speed of information intake must be taken into account. Ellis (1973) and Saariluoma (1984, 1985) have shown with very different procedures that skilled chess players are faster to extract information from chess positions than less skilled (Charness 1988). Chase and Simon (1973) have also noted that the more skilled the subjects the faster they learn chess games.

All these properties of chunks must be built into any model attempting to explain chess players' recall of chess positions. The model must contain an initially almost empty long term memory with a large number of simple chunks and it must be able to incrementally learn larger chunks. The pieces in chunks must have a number of chess specific relations between them and they must also be located in precise positions on a chess board. The speed of learning must also increase. The importance of building this kind of model is in testing the logic of theories. It has been known since the original study by Simon and Gilmartin (1973) that chunking can be studied in this way. Their model, however, was not a learning program and therefore it was not suitable in explaining the early learning curve. It is thus necessary to build a model, which is able to simulate the dynamics of the learning process. In the simulation model described below only the aspects of learning and precise location of pieces when building chunks are addressed, other chess specific heuristics are not considered.

Structure of the Simulation Program

Two versions of computer simulation programs were built to model the chunk construction strategies of novice chess players in the experiments described above. The models were programmed in the object-oriented language Java. Their functionally separate cognitive components are implemented as different object classes, instances of which are created during run time. The main classes are *piece, chunk, long term memory (LTM), short term memory (STM)* and *subject* that controls learning and recalling. The chess board is coded as two dimensional array of strings which present piece type and color. As the size of the array was the same as a real chess board's, the location of every piece was presented explicitly. For the chunking chess pieces (location, color and type) were coded as integers, and memory chunks were lists of these integers. The class hierarchy, and class or object methods were not intended to model human cognitive architecture or algorithms. The system was only to predict the development of the learning curve of an unexperienced chess player due to accumulation of new chess chunks in memory when her/his recall of unfamiliar chess positions is tested regularly during the learning phase.

Learning and Recalling

In the beginning the simulation systems have in LTM 768 chunks, which present every possible one piece chunk that can be formed, i.e. every piece type (12) on each location (64) on the board. So it was assumed that the subjects can trivially recognize single isolated pieces wherever they are situated on the board. After the initial situation the systems form new chunks in LTM from every shown study position. The size of chunks stored in LTM increases due to the systems' experience; in the beginning chunks of size two pieces are built, but later on the systems memorize larger chunks as they notice that possibly all the two (three, four, five, etc.) piece chunks are already known to them. The amount of chunks learnt from one position and used in recalling one position is limited by the capacity of the short term memory. Overlapping chunks are not constructed from a single position.

Unlike Simon's and Gilmartin's (1973) EPAM-based learner, the systems can find a chunk in memory independent of the piece around which it is built, so the chunk is identified by the pieces in it and no duplicates of the chunks are stored. Simon's and Gilmartin's chunks were identified by the focal pieces around which the chunks were built. However, the simulation systems do not examine the chess positions as a whole but process them one chunk at a time, starting with a specific or a random piece on a board. In the test phase the systems first build a proper chunk of the pieces on a test position, and then look for a corresponding chunk in LTM. So they do not reconstruct positions on empty boards like in Gilmartin and Simon (1973), but try to cover the pieces on the board with corresponding chunks in LTM if they are found. If the chunk cannot be found, the systems try a one piece smaller or a totally new chunk, otherwise they add the chunk to STM and mark the corresponding pieces on the board as recalled. Finally the recall score is calculated as a percentage of pieces explained by chunks in STM of all the pieces in the position.

Pieces or chunks that are not seen in learning phase are never memorized or retrieved, so the models make no commission errors. Once they have learnt something they never forget it, nor retrieve any incomplete or wrong chunk from memory. The models did not learn any chunks from the test positions, either.

Chunking Heuristics

The first version of the simulation is a naive one. It uses a **random neighbourhood heuristic**. In the learning phase and in the recall phase it always processes the chess positions in random order. It starts building a chunk from a random piece (focal piece), and when expanding a chunk to its neighbours the system proceeds to a random direction. It should be noted that only the pieces in adjacent location can form a chunk. The pieces that do not have any immediate neighbours can only form a one piece chunk.

The other simulation model uses a **correlation heuristic** when constructing chunks. Its decision about which pieces belong to a chunk is based on the frequency of co-occurrence of those pieces. The system chooses the most commonly seen piece as a focal piece around which it tries to form a chunk. Next the system adds to the chunks the most common neighbour of this focal piece, and then expands the chunk to the most common neighbour of this piece. However, in the learning phase the system starts examining the board and building chunks from random pieces. In this way it is guaranteed that the diversity of learnt chunks is high; not merely the most frequent pieces or chunks around them are exploited. Additionally, the multiplicity of chunks was thought to be of some use in recalling random positions.

The correlation model keeps record of the occurrences of single pieces and two piece combinations in a matrix like table. Note, that the system calculates frequencies of those pieces only that it memorizes in learning phase.

Neither of the models take into account the possibility of building very oddly shaped chunks. Despite of the chunking heuristics their structure and functioning is identical.

Simulation Results

The conditions in which the simulation models were tested were similar to those of the subjects MQ and NT. The models were taught 500 chess positions and the recall of unfamiliar game and random positions was requested within the same intervals as MQ's, in the beginning, after 30, 60, 175, 220, 270, 350 and 500 studied positions. Every test session consisted of ten real game and ten random positions. The random positions were permutations of the real chess positions used in the tests. They included just the same number and type of pieces, only in different locations.

For curiosity, test runs were run with short term memory sizes 4, 7, 10 and 12 chunks, because it was not very clear in the beginning whether it was just the quality of the chunks, not the number of them used in recalling that could improve the performance most. The short term memory size of 4 produced quantitatively the most similar results to the novice human subjects. When the STM size was over 7 chunks, the performance reminded that of experts'. The learning curves with STM size 4 for real game positions and random positions, and for both models are presented in figure 2. The curve is plotted such that for every test session the average recall percents of game and random positions are calculated, and then the whole test sequence is averaged over 20 independent runs. Similarly, the learning curves for all the short term memory sizes and both chunking heuristic in real game conditions are presented in the figure 3.

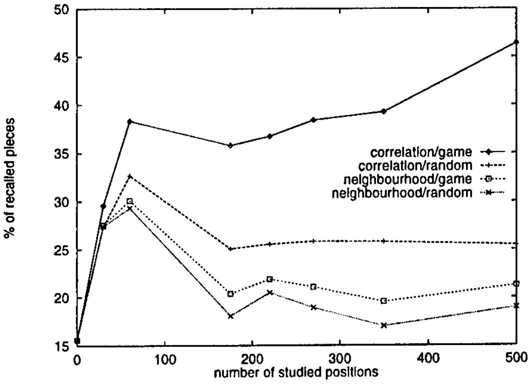


Figure 2: Percentage of recalled pieces when the simulation systems were tested 8 times during learning. Short term memory size was 4 chunks.

When evaluating the simulation versions it is very clear that the correlation model can exploit the regular patterns seen on chess board much better than the random neighbourhood model. Hence it is able to memorize the most useful piece combinations which help it to recall more pieces in the test situations. The other model stores too much redundant information. With the same amount of stored chunks it could recall much less of the game positions. The both simulations performed worse in the random test condition than in real game test condition, but the correlation model performed significantly worse than in the real game condition. With the neighbourhood model the difference was not so big. Still in the random con-

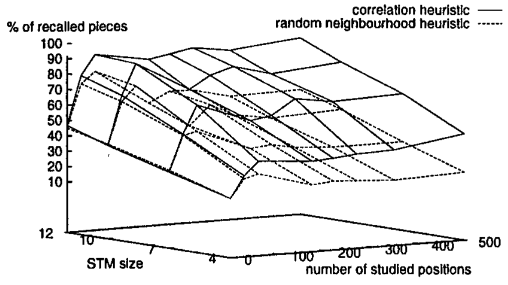


Figure 3: Learning curves for the short term memory sizes 4, 7, 10 and 12.

dition the correlation version reached better results than the neighbourhood version in the real game condition.

The Effect of the Amount of Learnt Chunks vs. Short Term Memory Size

The hypothesis was that the sharp increase at the beginning and the modest increase later on in the learning curve is caused by early accumulation of relevant chunks in LTM. Although the chunk amount goes up at a quite constant rate throughout the learning time the recall score does not seem to reflect it very well. It seems like after some turning point remarkably more chunks would be needed to enhance the performance. Note that the program using a neighbourhood heuristics could produced no more than about 700 chunks while the other one discovered a little over 2000 chunks. The accumulation of LTM chunks for different STM sizes and the two chunking heuristics is plotted in figure 4.

The size of the simulated short term memory played more drastic role in the performance than learning the relevant chunks. With the minor size (4 chunks) it was impossible to reach the results that were quite easy to obtain with STM sizes 10 or 12, as can be seen in figure 3. Otherwise a huge amount more learning would be demanded, say 10000 or 50000 LTM chunks (which may be normal for expert chess players). Our simulation did learn only about 2000 chunks at its best, and the amount of formed chunks did not, somewhat surprisingly, vary with STM size, when only 500 positions were studied. However, the amount kept on increasing linearly when the number of studied positions was doubled to 1000 (the results of these runs are not reported here, because the recall score did not improve at all). If the formation of overlapping chunks had been allowed, the amount of stored chunks may have been remarkably larger, but the recall scores somewhat smaller, because the same pieces might have been included in several chunks when recalling a single position.

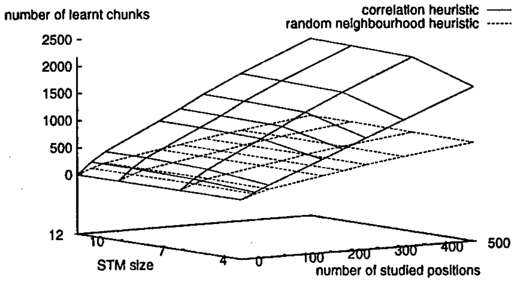


Figure 4: Accumulation rate of long term memory chunks during learning for both heuristics, run over STM sizes 4, 7, 10 and 12.

The Effect of the Chunking Method vs. the Size of Chunks

It was noticed that not merely the size of the chunks was important for the performance but the quality of them, although it was hypothesized that the learning is due to the gradual accumulation of bigger and bigger chunks. In practice the system could not exploit much larger chunks than five pieces on average, because it could learn only a fraction of the combinatorial alternatives, as the amount of them grows exponentially with the size. For this reason the longer chunks were harder to match to the game positions as they were seen so rarely during the learning phase. In the figure 5 the average sizes of the chunks used by the model using correlation heuristic in recalling real game positions are presented. The curve is plotted as an average of the largest chunks used to recall the ten test positions in every test session.

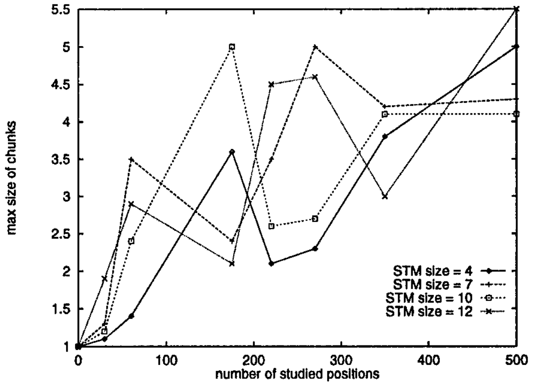


Figure 5: The average sizes of the largest chunks used in recalling real game positions by correlation model, plotted for all STM sizes.

It was also noticed that the overall method used in building chunks produced significant differences in performance. The method that incrementally builds larger chunks adding one adjacent pawn to an existing chunk improved the performance considerably compared to simple accumulation of chunks of different sizes. Hence it is advantageous to add one pawn to the earlier memorized chunk than to memorize two almost separate chunks that do not have many pieces in common. The latter method forms more variable chunks but it does not take into account the nature of real game positions. Although the test positions were not consecutive positions from a single game, especially the model using correlation heuristic was superior in exploiting regular patterns in positions it had seen in learning phase. The incremental method building chunks uses additionally the idea that it does not really matter that one pawn has changed place, it can still retrieve the chunk partially i.e. recall a one pawn smaller chunk that it has possibly built earlier. It may not have seen all the one piece smaller chunks previously, but in practice quite a few of them, anyway.

GENERAL DISCUSSION

These empirical results are very clear. A negatively accelerating learning curve was found. In this aspect chess is similar to many other symbolic and motor skills: The first steps are always the fastest in acquiring any skill. During this period one learns the most basic but also most common aspects of the domain. In chess this means the very familiar chunks such as castling or standard pawn chains. In random positions the absence of similar regularities makes it impossible for them to find equally common piece configurations and chunking is much less effective. Later in skill development the number of pieces in a chunk will increase and the number of combinatorial possibilities also increases exponentially. Consequently, it is very logical that the increase in the number of recalled pieces decreases respectively.

By a computer simulation we investigated various possibilities to interpret the empirical data. Of course, simulation has several weaknesses as a method. It is far from being unambiguous, because it is possible to construct several different types of models to investigate possible interpretations of data. Nevertheless, one should not forget that it is still better than mere intuition. The formal dimension of thinking is better controlled by using modeling than by relying on intuition. Therefore, a simulation, though having undeniably speculative sides, can be beneficial.

In our simulation, we were at first interested in the form of learning curve. Therefore, we let chunks grow incrementally, and indeed, our assumption was correct. There was a clear difference between game and random positions. This means that the incremental growth of chunks is a very good conjecture for the explanation of the negatively accelerating learning curve in chess. The exponential growth in the number of possible chunks as a consequence of the increase in the required chunks' length effectively explains the form of the learning curve.

It was interesting to notice that we got the best fit with the data when the size of working memory was kept at 4. On one hand this piece of evidence fits extremely well with the classic models of chess players' memory suggested by Chase and Simon (1973), for example, in which chunks are stored into short term working memory. However, the empirical research since Charness (1976) has clearly shown that experts do not store information into their short term working memory but in long term working memory (Ericsson and Kintsch 1995). This is apparently very problematic, but we must remember that in this experiment we investigated early learning. Therefore, there is nothing strange in that people would use their short term working memories to store chess specific information. The development of retrieval structures typical to masters takes a decade.

Finally, we were interested in the nature of associative connections between the chunk elements. The two simulation models suggest very evidently that frequency based correlative chunks provide much better a model for the data than random neighbourhood model. Indeed, this fact is also in harmony with the classic theory of associations, which assumes that frequent co-occurrence is sufficient explanation for many associations.

The ultimate point of simulation is the analysis of the interconnections between various phenomena and cognitive mechanisms. In that way simulation allows us to provide global theoretical concepts with more accurate contents than is possible when basically intuitive theoretical notions are used. This is an important point when the foundations of psychological argumentation is considered (see Saariluoma 1997). Here, the main problem is to find, how the learning curve, chunks growth, ST-WM capacity are interrelated and what is the significance of these finding in global psychological terms.

The model suggest that chess skill is essentially based on associative piece configurations and the basic learning mechanism is a gradual construction of them. The problem in improving memory recall is to resolve the combinatorial problem of getting sufficient number of chunks to get full coverage of standard real game piece configurations. The learning curve shape is thus simply a consequence of required number of chunks on each level of length. More chunks of length five need to be stored than chunks of length three. Consequently, model suggests that the shape of early learning curve is a consequence of combinatorial properties of the materials and limited capacity of the system.

In global terms, one can argue that chunking is one form of knowledge construction. As it is well known, the major contemporary global learning theory is called constructivism. It is predominant way of thinking as well in clinical as in social and educational psychology (Resnick 1987). The crucial theoretical problem in this way of thinking is the notion of construction itself. What does it mean, in concrete terms, that people construct their knowledge bases. The simulation of early learning provides one alternative. It is frequency based construction of associative and prelinguistic patterns.

A problem in this context is the precise role of learning results and chunking mechanism. de Groot and Gobet (1996, p.117) criticise Chase's and Simon's (1973) chunking explanations relying on Ericsson's and Harris's (1990) experiment in which they showed how a novice, by using mnemonic techniques, can improve his/her performance with no improvement in chess skills. The point is that chunks only do not suffice in chess, but knowledge about moves is also required. The authors are naturally correct in their thinking.

Nevertheless, one can argue that the memory mechanism of chunking is not bound to static piece patterns, but moves are sequences of spatio-temporal chunks. Thus chunking in position recall tasks utilizes the same underlying mechanisms that all learning of chess knowledge. Blindfold game recall strongly speaks for this interpretation (Saariluoma 1989). The problem is that one must learn all relevant types of chunks, i.e. piece configurations and moves, to improve one's chess skills. The concept is relevance (de Groot and Gobet 1996, Saariluoma 1995). If people do not learn relevant spatio-temporal chunks their skill construction is biased. Indeed, much of our conceptual knowledge is in these tacit patterns and therefore it is so important to understand these knowledge construction mechanism also in early learning (Saariluoma 1995, 1997).

ACKNOWLEDGMENT

This work was supported by Foundation of the 350th Anniversary of University of Helsinki grant to Tei Laine.

REFERENCES

Charness, N. (1976). *Memory for chess positions: Resistance to interference.* Journal of Experimental Psychology: Human Learning and Memory, 2, pp. 641-653.

Charness, N. (1988). *Expertise in chess, music, and physic: A cognitive perspective.* In L. Obler and D. Fein (Eds.), The exceptional brain. New York: Guilford press.

Chase, W. G. and Simon, H. A. (1973). *The mind's eye in chess.* In W. G. Chase (Ed.), Visual information processing. Academic Press: New York.

Ellis, S. H. (1973) *Structure and experience in the maching and reproduction of chess patterns.* Unpublished doctoral dissertation, Carnegie-Mellon University, Pittsburgh. Citated from Charness (1988).

Ericsson, K.A., Harris, M.S. (1989), cited in de Groot and Gobet (1996).

Ericsson, K.A. and Kintsch W. (1995). *Long term working-memory.* Psychological Review 102, pp. 211-245.

Fisk, A. W. and Lloyd, S. J. (1988). *The role of stimulus-to rule consistency in learning rapid application of spartial rules.* Human Factors, 30, pp. 35-49.

Frey, P. W. and Adesman, P. (1976). *Recall memory for visually presented chess positions.* Memory and Cognition 4, pp. 541-547.

de Groot, A. D. (1965). Thought and choise in chess. The Hague: Mouton.

de Groot, A. D. (1966). *Perception and memory versus thought: Some old ideas and recent findings.* In B. Kleinmuntz (Ed.), Problem solving: Research, methods and theory. New York: Wiley.

de Groot, A. D., Gobet, F. (1996), Perception vs. Memory in Chess. Assen: van Gorcum.

Lane, D. M. and Robertson, L. (1979). *The generality of the levels of processing hypothesis: An application to memory for chess positions.* Memory and Cognition, 7, pp. 253-256.

Lories, G. (1987). *Recall of random and non random chess positions in strong and weak chess players.* Psychologica Belgica, 27, pp. 153-159.

Newell, A. and Rosenbloom, P. (1981). *Mechanisms of skill acquisition and law of practise.* In J. R. Anderson (Ed.), Cognitive skills and their acquisition. Hillsdale, N. J.: Erlbaum.

Newell, A. and Simon, H. A. (1972). Human problem solving. Engelwood Cliffs, N. J.: Prentice-Hall.

Resnick, L.B. (1987), *Learning in school and out.* Educational Researcher, December.

Rosenbloom, P. and Newell, A. (1987). *Learning by chunking: A production system model of practise.* In D. Klahr, P. Langley and R. Neches (Eds.), Production system models of learning and development. Cambridge, Mass.: MIT Press.

Saariluoma, P. (1984). *Coding problem spaces in chess.* Commentationes scientiarum socialium, vol 23. Turku: Societas scientiarum fennica.

Saariluoma, P. (1985). *Chess players' search for the task relevant cues.* Memory and Cognition 13, pp. 385-391.

Saariluoma, P. (1989), *Chess players' recall of auditorily presented chess positions.* European Journal of Cognitive Psychology 1, pp. 309-320.

Saariluoma, P. (1995), Chess Players' Thinking: A Cognitive Psychological Approach. Routledge, London.

Saariluoma, P. (1997), Foundational Analysis. London: Routledge.

Simon, H. A. (1976). *The information storage system called human memory.* In M. Rosenzweig and E. Bennett (Eds.), Neural mechanisms in learning and memory. Cambridge, Mass.: MIT-Press.

Simon, H. and Gilmartin, K. (1973). *A Simulation of memory for chess positions.* Cognitive Psychology, 5, pp. 29-46.

Problem Solving with Incomplete Information: Experimental Study and Computer Simulation

Nathalie Chaignaud
LIPN - CNRS UPRES-A 7030
Université Paris XIII
Avenue J.B. Clément
93430 Villetaneuse - FRANCE
nat@lipn.univ-paris13.fr

Anh Nguyen-Xuan
Laboratoire de Psychologie Cognitive
Université Paris VIII
2, rue de la liberté
93526 Saint Denis Cedex - FRANCE
anguyen@ext.jussieu.fr

ABSTRACT

The aim of this study was to understand some particular human methods of problem solving in everyday situations. In this aim, we designed an experiment to obtain individual protocols. A cognitive model was based on the notions of *phases* and *states of mind* that evolved during the problem-solving process. The proposed model was then implemented in IGGY, a system which uses a blackboard architecture, and the validity of the model was tested by a Turing-like test and by a statistical analysis.

Keywords

Cognitive modelling, problem solving, incomplete information, model validation, blackboard system.

INTRODUCTION

In everyday life, people frequently encounter incompletely described situations where common sense reasoning and planning are essential. In most of these situations, the complexity of the reasoning process comes both from the fact that the state space in which constraints have to be satisfied is so large that no combinatoric approach can be used, and from the fact that some information is missing and hence must be collected. A simplistic example of such a situation is when someone wants to organise a party with friends. To this end, several constraints must be satisfied and the problem cannot be solved instantaneously because information is missing.

The way the solution is built up depends in particular on how strictly people pay attention to the constraints and how well they gather and use information. In this kind of problem, people may choose to use sophisticated reasoning that resembles planning, by optimally articulating the pieces of information already known, inferring the best way to gather the missing information, and anticipating the different possible outcomes. On the opposite, they may decide to avoid paying an important cognitive cost and adopt a simple behaviour, driven more by reaction than by planning. However, although Agre and Chapman (1987) stated that this reactive behaviour was cognitively plausible, there is, to our knowledge, no experimental evidence of such activity for human subjects.

In this interdisciplinary study, at the intersection of Cognitive Psychology and Artificial Intelligence, our purpose is to understand and simulate the way human beings elaborate plausible conclusions in imperfectly described everyday situations. To this end we have chosen to carry out a psychological experiment dealing with an *ill-structured problem* (Simon, 1973; Voss & Post, 1988; Goel, 1992). In this class of problems, people have to reason on incomplete or uncertain knowledge. Design problems (Guindon, 1990; Visser, 1990; Ball et al., 1997) form a particular subclass of this class.

In our work, a *bottom-up* approach has been adopted: an experiment was conducted in which subjects were to solve individually the so-called "hi-fi system problem". The set of experimental protocols obtained were analysed to extract the different behaviours, and from this analysis a computational model was built and implemented in order to have a better understanding of the human problem-solving process. Finally the output of this simulation was compared with the human protocols in order to validate the proposed model.

THE PROBLEM-SOLVING SITUATION

The task was designed with the following characteristics:

- the set of constraints could be satisfied in a large state space, and a pure combinatoric solution could not be considered;
- information initially available had to be incomplete in order to compel the subjects to reason in an uncertain environment;
- it had to be of sufficient complexity so as to obtain a large range of behaviours; however it had to be simple enough to be manageable.

The problem consisted in configuring a hi-fi system. A complete system comprised five different items: an amplifier, a tuner, a record player, a tape recorder and a compact disc reader. The subject could choose between three models of amplifiers and between four models for each of the other items. The amplifier had a special status insofar as the other items had to be compatible with it[1]. The price of the items, the maximum amount allowed and the compatibility between the amplifiers and the components[2] were given to the subjects at the beginning of the experiment. However, the subjects did not know which items were available at the beginning. This information had to be acquired by making a phone call for each chosen item and the subject knew that s/he would be told the number of calls allowed in due time. Thus, there were four types of constraints: the total price of the system, the compatibility between the amplifier and the components, the availability of the chosen items and the number of phone calls.

In order to record all the actions performed by the subjects, the task was simulated via a computer program.

[1] In the remainder of this article, the amplifier is thus differentiated from the other items called "components".

[2] Some components were compatible with more than one amplifier.

The user interface was designed to serve as an external memory store and a calculator of the total amount spent on the chosen items. In order to instigate a large variety of behaviours, three versions of the problem were built, which differed by price and compatibility table. Forty seven female and male students took part individually in the experiment. Each subject was asked to solve the problems by thinking aloud. The subjects' verbalisation was tape recorded and all their actions were automatically recorded by the simulation program. Therefore, a *protocol* was a list of all the subject's actions (and results of actions) and verbalisations during a problem-solving process.

Fifteen of the 141 individual protocols were eliminated from data corresponding to certain subjects that did not understand the instructions. Therefore the raw data comprised 126 protocols.

BUILDING A FRAMEWORK TO MODEL THE PROTOCOLS

The model had to be realistic, complete and simple. This needed a lot of comings and goings between the analysis of the experimental data and the building of the model. Therefore, our cognitive model was built by successive approximations.

An ideal strategy

An optimal way for handling the problem situation consists in selecting three configurations, based on three different amplifiers, and in choosing components that are as multi-compatible as possible. By doing so, one can make sure that the constraints of price and compatibility are satisfied. Information gathering (i.e. by phoning) will be undertaken only when the configurations are completed. This *ideal strategy* is based on parallel planning and can be characterised as being "opportunistic" when item choice and information gathering exploit the idea of multi-compatibility. Such a strategy is similar to a breadth-first search observed in expert designers (Ball et al., 1997), because it takes into account the three possible alternatives at the same time and leads to a solution with a minimum risk of backtracking.

A rough characterisation of the protocols

The subjects could solve the problem at different levels of reasoning, from the most sophisticated to the simplest mode. We characterised a mode of reasoning as being sophisticated when (i) the subject built in parallel three configurations based on the three amplifiers, then explored in depth the configuration that appeared the most promising at a given time; (ii) the subject explored at the same time several solutions with a single amplifier; (iii) the subject used a strategy similar to the *focusing* one (Bruner et al, 1956) by phoning in order to reduce the set of possible solutions quickly. This mode of reasoning resembled the ideal strategy presented above. The reasoning was characterised as being shallow when the subject tried to build up only one configuration at a time. This mode of reasoning is similar to the depth-first search approach observed in novice designers. It bore two characteristics: (i) the subject abandoned the current solution only when forced to do so (i.e. one or more constraints were violated); (ii) the subject phoned for a component of a given category then shifted to another category as soon as s/he got a positive answer.

To our surprise, after a first superficial analysis of the protocols, we did not find much sophisticated reasoning. Only 13 of the 126 protocols can be characterised as adopting a sophisticated mode of reasoning. In the remaining 113, subjects focused on building up one configuration at a time.

Moreover, the subjects did not respect simultaneously all the constraints of the problem.

Despite the fact that the observed behaviours were simpler than expected, there did not exist two identical protocols. The question that arose was then how different were they? To answer this question we needed to define a framework for analysing the protocols more precisely.

This analysis takes into account the protocols that adopt a shallow mode of reasoning.

From the 113 protocols, three sets of protocols were drawn at random. The first set comprised 30 protocols which have been carefully analysed to determine the main ingredients of the cognitive model and to build up a precise method of analysing the protocols by hand. The second set of 43 protocols was randomly drawn from the remaining protocols to validate the completeness of the hand analysis method. The third comprising the 40 remaining protocols was used to validate the implemented model: they were not used for the setting up of the parameters of the implementation.

The ingredients of the model

Our model was based on the notions of phases, states of mind, strategies and tactics.

The Notion of Phases

The configuration building process rarely developed smoothly and some "obstacles" arose which had to be overcome. Two kinds of obstacles were distinguished: either they were not really bad and the situation needed only a few corrections or they were more serious and constituted deadlocks which needed to be removed. From this point, we differentiated between the situations considered as being *normal* and those considered as *abnormal*, with two degrees of *abnormality*.

In every abnormal situation, the configuration building process was interrupted and the subjects undertook either a correction task or a deadlock-solving task. After the obstacle had been solved, they either returned to their previous task or tested their configuration if they thought they had found a solution. Thus, each protocol could be divided into *phases* characterised by the current task. These phases were *configuration building*, *correction*, *deadlock solving* and *test*. Figure 1 represents all the possible relations between the phases.

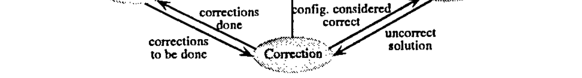


Figure 1: the possible relations between phases

The Criteria Taken into Account by the Subjects: the State of Mind

The decision to perform any particular action depended on the attention paid to the different constraints. Thus, we defined the notion of *criteria* taken into account by the subjects. They were related to the four constraints of the problem, and they were given the same names:

- *compatibility criterion*: the subjects focused on the compatibility between the different items,
- *availability criterion*: the subjects tried to check as soon as possible the availability of items,
- *price criterion*: the subjects took into account of the price of the items,
- *phone calls criterion*: the subjects were careful how many phone calls they made.

The criteria were different from the constraints: satisfying a constraint meant making sure that it was not violated, whereas complying with a criterion only meant that the subjects had this constraint in mind while taking decisions, in order to reduce the chances of violating it.

The subjects did not necessarily take into account all criteria at the same time but only a subset of them that varied as new information was acquired. This subset of criteria was called *state of mind*. It evolved according to the problem-solving situation and its changes triggered modifications in the subject's behaviour.

The Possible Strategies

In a configuration building phase, the subjects could perform two different kinds of actions: choose the items that will form a configuration and gather information about the availability of items. We distinguished between *item choice strategies*, and *information acquisition strategies*.

There were two possible item choice strategies:

- *amplifier centred strategy* (Strategy 1): choose the components by focusing on only one amplifier,
- *component centred strategy* (Strategy 2): choose the components without having in mind a predetermined amplifier, so that the determination of a single amplifier was delayed as long as possible.

Three possible information acquisition strategies, probably related to the user interface used in the experimentation, emerged from the protocols:

- *select then phone strategy* (Strategy 4): choose several items and then phone for each of them,
- *phone then select strategy* (Strategy 5): give a sequence of phone calls for a series of items and then build up a solution by selecting only available items,
- *phone and select simultaneously strategy* (Strategy 6): select each item and then phone immediately for it (if available, the item was kept, else it was discarded, and the subject selected another item).

Moreover, the subjects could forget that it was necessary to know the availability of all the elements of a chosen configuration. In this case a *null strategy* (Strategy 7) was attributed.

In a deadlock-solving phase, the subjects had to change the flawed configuration by deciding to focus the deadlock-solving process on either an amplifier or a component. All the strategies, except Strategy 4 and Strategy 7, could be applied.

Finally, since the correction phase concerned the items that violated the constraints, no strategy on item choice was necessary. This is the reason why a *null strategy* on item choice (Strategy 3) was attributed to any error correction phase.

Table 1 summarises all the possible strategies in the different phases.

Item choice strategies	**CB**	**C**	**DS**
1. amplifier centred strategy	X		X
2. components centred strategy	X		X
3. null strategy		X	
Information acquisition strategies	**CB**	**C**	**DS**
4. select then phone strategy	X	X	
5. phone then select strategy	X	X	X
6. phone and select simultaneously strategy	X	X	X
7. null strategy	X	X	

Table 1: strategies for each phase
(CB: configuration-building phase, C: correction phase, DS: deadlock-solving phase)

Instantiating the Items Choice Strategies: the Tactics

In the configuration building and deadlock-solving phases, the same strategy on item choice could be instantiated through different atomic actions. In order to differentiate between these different choices, we introduced the notion of *tactics*.

	CB		**DS**		**C**
Tactics on amplifier choice	**St. 1**	**St. 2**	**St. 1**	**St. 2**	**St. 3**
1. cheapest amplifier	X		X		X
2. medium-priced amplifier	X		X		X
3. amplifier compatible with the most components	X		X		X
4. amplifier compatible with the fewest components	X				X
5. amplifier most compatible with the configuration	X		X		X
6. amplifier compatible with available components	X		X		X
7. amplifier compatible with the cheapest components	X		X		X
Tactics on components choice	**St. 1**	**St. 2**	**St. 1**	**St. 2**	**St. 3**
8. cheapest comp. compatible with a given ampl.	X				X
9. medium-priced components compatible with a given ampl.	X				X
10. available comp. compatible with a given ampl.	X				X
11. cheapest comp. compatible with at least 2 amplifiers		X			X
12. components compatible with the most amplifiers		X			X
Tactics on key-component choice	**St. 1**	**St. 2**	**St. 1**	**St. 2**	**St. 3**
13. cheapest key-component		X		X	
14. key-component of blocking category				X	
15. available key-component				X	

Table 2: tactics in terms of phases and strategies

From an informal analysis of the 30 protocols, 15 different tactics were identified, which depended on the current phase, on the state of mind and on the item choice strategy. We distinguished between tactics for choosing an amplifier and tactics for choosing the components. In the latter case, there was an additional distinction between situations where a set of components had to be chosen and situations where only one component, called *key-component*, had to be chosen in order to start or restart a configuration. Table 2 presents the fifteen tactics and their domain applicability.

METHOD TO ANALYSE THE PROTOCOLS

The hand analysis was performed in parallel by two "judges"[3] and the infrequent disagreements between them were easily solved after a discussion with a third judge. The method to analyse the protocols was applied to 73 protocols: the set of 30 that were used to build up the framework of the cognitive model, and the set of 43 that were reserved to verify the completeness of the hand analysis method.

Decomposing the protocols into phases

Identification of the test phase was straightforward: it boiled down to the "test" action.

A deadlock-solving phase was usually a short sequence of actions that eventually led to a change of amplifier. It began when the subjects considered that they would not be able to find a solution with the current amplifier. After possibly checking the availability of some items, the subjects chose either a new amplifier or a component compatible with another amplifier. This choice ended the deadlock-solving phase.

A correction phase, by contrast, was a phase where the subjects checked the availability of items and/or replaced components that violated the constraints. It could be triggered by a negative test or simply by the subjects' noticing one or more errors in the solution.

A configuration building phase was simply defined as a phase that was none of the three phases defined above.

Recognising the strategies and the tactics

Once the phases had been identified, strategies and tactics were rather easy to detect. But the identification of strategies and tactics could not be conducted separately. Most actions, when taken out of context, were compatible with more than one strategy and more than one tactic according to the possibilities given in tables 1 and 2. It was often necessary to take into account a sequence of actions in order to narrow the range of possibilities. As mentioned earlier, the verbal utterances were good clues to help choose among the possible hypotheses. Thus the approach we used to identify strategies and tactics was a hypothesis-and-test approach.

Identifying the state of mind

The last ingredient to be identified in the protocols was the state of mind. To this end, we assumed that any criterion that appeared in the state of mind had a visible effect in the protocol.

The information acquisition strategies depended on the availability criterion . Strategy 5 and Strategy 6 favoured an early discovery of the availability and therefore required the presence of the availability criterion, whereas Strategy 4 and Strategy 7 were inconsistent with it.

Each tactic corresponded to one or two criteria, and some tactics excluded a criterion. The compatibility criterion was also attributed when the subject explicitly referred to it when choosing items.

The subjects could change their states of mind, strategies or tactics during a configuration building phase. This meant that a configuration building phase could be divided into several episodes. An episode was defined as a sequence of actions characterised by the same set of phase, state of mind, strategies and tactics. The end product of the analysis of a protocol is a *skeleton* which partitions the protocols into successive episodes.

GENERATING ARTIFICIAL PROTOCOLS

Our aim was to build a computational model to simulate human reasoning in this particular problem-solving task. Protocol analysis made it possible to identify, for each protocol, the successive phases, strategies, tactics, states of mind and their changes during the problem-solving process. However, the model did not "explain", why different subjects adopted different states of mind, strategies and tactics and why some of them made more careless mistakes than others. In order to introduce this inter-individual variability, individual characteristics had to be taken into consideration. This led us to introduce the notions of *observation* and of *personality.*

Linking the episodes: observations as triggers of the episode changes

During problem solving, any change of ingredient corresponds to a new episode, which depends on the subjects' interpretation of the current situation. From a generative point of view, our aim was to simulate not only behaviours inside episodes but also the inferences the subjects made from the current situation in order to go ahead. For this we used the notion of *observation*, which corresponds to the explanations about the problem-solving process that would be present in the subjects' verbalisations if these were complete. The subjects' verbal utterances are thus considered as a sample of their observations and they play an important role to explain the changes of episode.

Observations may concern the current configuration as a whole (e.g. "the configuration is too expensive") or a particular element (e.g. "tuner 1 is available"). They can have an impact on the phase (e.g. deadlock, correction to be done or configuration to be tested), on the state of mind (e.g. number of phone calls already made too high: take into account the phone calls criterion), on strategies (e.g. an element compatible with several amplifiers: Strategy 2) and/or on the current tactics (e.g. expensive configuration: "cheapest components" tactic).

From the 73 protocols that have been hand analysed, all the verbal utterances were picked out except for the meta-cognitive statements. The list of useful utterances can be grouped into 22 observations. In our model, the observations are represented by predicates with or without arguments. They are not described here due to space limitation.

The personality of the subjects

The second notion we needed in order to simulate the diversity of the observed behaviours was *personality.*

For each of the 73 protocols that have been analysed, we determined the personality of the subject by 5 orthogonal features:

- *careful*, for the frequency of careless mistakes made by the subject,
- *thrifty*, for the importance attached by the subject to the price of the configuration,
- *opportunistic*, for the subject's ability to use information flexibly,

[3] We thank Jean-Marc Meunier for having done a very efficient job as one of these two judges.

- *systematic*, for the subject's choice of elements that followed more or less strictly the order in which the categories were presented in the price and compatibility table,
- *good appraiser*, for the subject's aptitude to estimate a situation correctly.

All of them can take the values *poorly*, *fairly* and *very*.

In the process of generating artificial protocols, these parameters are given as data of the problem-solving process and observations and choices are made essentially according to them.

IMPLEMENTATION OF THE COGNITIVE MODEL: THE IGGY SYSTEM

From a psychological point of view, the implementation of our cognitive model aims to validate the model and, from an AI point of view, to show the feasibility of such a computational model. IGGY, a system written in Common Lisp, implements the model: it is a protocol generator that takes as input a personality and gives as output a protocol corresponding to this personality.

We need an architecture allowing the specialisation of knowledge and the sequentiality of actions. The *blackboard architecture* (Engelmore & Morgan, 1988, Hayes-Roth, 1993) with a *hierarchical control* is well suited to our needs.

IGGY's components

IGGY is a hybrid system which contains three elements: a blackboard, an *executor* and an *engine* (see Figure 2). The engine co-ordinates the other two elements in a "perception-decision-action" loop in disguise, where the perception and decision tasks are accomplished by the blackboard and the actions are performed by the executor, which generates protocols.

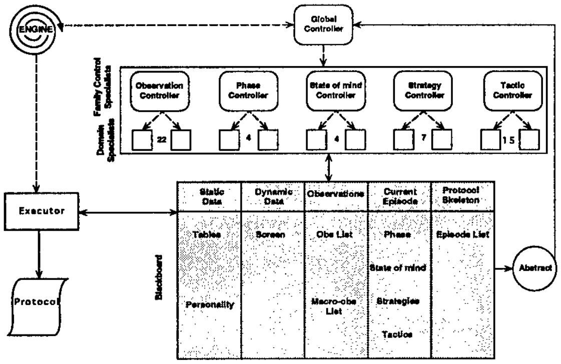


Figure 2: the system IGGY

The Blackboard System

It includes a blackboard, domain specialists grouped in five families, family controllers and a global controller.

The blackboard

It includes five thematic panels: the static parameters of the problem (compatibilities, the price of the items and the personality of the simulated subject), the dynamic data of the problem (the availability of the items and the configuration), the current observation list, the current episode and the history of the different episodes of the problem solving (the skeleton).

An *abstract*, updated at each change of the blackboard, informs the system on the nature of the new information (phase, state of mind, strategies, tactics, observations or action).

The domain specialists

There are fifty two domain specialists grouped into five families representing the ingredients. Each specialist corresponds to a possible choice in its family. Thus, there are twenty two specialists for the observations, four for the phases, four for the states of mind, seven for the strategies and fifteen for the tactics.

They are represented by "condition/action" rules: their condition concerns the state of the blackboard and their action consists in writing a new instantiation of an ingredient on the "current episode" panel, except for the "observation" family that writes on its own panel.

The family controllers

Each control specialist, called *family controller*, concerns one family and knows the list of domain specialists that it supervises. At the beginning of the problem-solving process, the observation controller is triggered to initialise the observation list. Observations are generated according to the personality. From this list of observations, the first episode is calculated by the phases controller, the states of mind controller, the strategies controller and the tactics controller. Then the executor performs just one action according to the episode. After each executed action, the observations controller is triggered and either new observations are written in the observations list, or no observation is made. In the first case, a new episode is calculated. In the second case, the executor continues with its job, and so on.

The family controllers are specified by condition/action rules. The condition concerns the state of the abstract and is defined so that family controllers are triggered cyclically in the following order {observations, phases, states of mind, strategies and tactics}. The action is threefold: send a call for proposals to the domain specialists, choose one of the candidates and trigger it. If there is no candidate then the next family controller is triggered.

The global controller

The global controller supervises all the family controllers. It is reduced to the action part: as a control specialist, it send a call for proposals to the family controllers, then chooses one candidate and triggers it. When a new observation is made that raises a conflict between family controllers, the global controller chooses in priority the "phase" family controller, but if no domain specialist proposes a change, then comes the turn of the "state of mind" family controller, and so on.

The Executor

This module generates the sequence of actions corresponding to the current episode chosen by the blackboard system. It executes only one action at a time and gives the control back to the engine.

VALIDATION OF THE MODEL

Validation consists in comparing a set of real protocols with a set of simulated ones. The first (real) set is the 40 protocols that had been put aside to validate the implemented model. The second (simulated) set of protocols has been provided by IGGY: 73 protocols have been generated, each having the personality of one of the 73 analysed protocols. Then, 40 of the simulated protocols have been randomly selected to constitute the artificial sample.

Turing-like test

The first validation method was based on a Turing-like test (Turing, 1950).

We have adapted this test in the following way: from the two sets of protocols described above, we randomly drew two samples of 15 protocols. The 30 protocols were given to the psychologist[4] who had already analysed the 73 real protocols. We asked him first to hand analyse them (using the same analysis framework to derive 30 skeletons), and second to classify them according to their origin (human or artificial). The results were very good since he misclassified half of the protocols: 8 artificial and 7 real protocols were classified wrongly by the psychologist.

Statistical comparisons

The comparisons were based on a set of observables obtained from the protocols of the two samples (40 protocols for each sample). The significant level we adopted was p=.05.

We counted separately the number of episodes for the four different phases: configuration building, correction, deadlock solving and test.

Table 3 shows the data concerning the configuration building episodes: data in the cells represent the number of protocols in which there were one, two, three, four or five configuration building episodes. By combining the "4 episodes" with the "5 episodes" cells, we obtained $\chi^2(3)=27.75$ (p<.05).

Prot.	1 epis.	2 epis.	3 epis.	4 epis.	5 epis.
IGGY	1	6	15	13	5
Real	4	25	9	2	0

Table 3: Number of protocols by number of configuration building episodes.

Table 4 shows data concerning the test phase: in the cells are the numbers of protocols in which there were one, two, three, or four test episodes. In order to use the χ^2 test, we considered two categories, "one episode" and "two-or-more episodes": $\chi^2(1)=12.29$ (p<.05), the difference was significant.

Prot.	1 epis.	2 epis.	3 epis.	4 epis.
IGGY	36	3	1	0
Real	22	15	2	1

Table 4: Number of protocols by number of test episodes.

Tables 5 and 6 show data concerning, respectively, the deadlock solving and the correction phases. As in tables 3 and 4, data in the cells represent the number of protocols. For these two types of data, the differences between the two groups are not significant: $\chi^2(2)=4.64$ (p=.10) and $\chi^2(2)=1.04$ (p=.59) respectively (the last 3 categories of Table 6 have been combined).

Prot.	0 epis.	1 epis.	2 epis.
IGGY	7	20	13
Real	10	25	5

Table 5: Number of protocols by number of deadlock-solving episodes.

Prot.	0 epis.	1 epis.	2 epis.	3 epis.	4 epis.
IGGY	22	11	6	0	1
Real	20	15	4	1	0

Table 6: Number of protocols by number of correction episodes.

We also counted the number of protocols in which a given strategy or tactic was observed at least once. For Strategy 1 and Tactic 15, no statistical test was needed; for the artificial and real protocols there were, respectively, 40 and 39 Strategy 1, and 1 and 0 Tactic 15. For the other data (six strategies and twelve tactics), Tactic 11 and Tactic 12 being combined, we used the χ^2 test when possible; else the Fisher exact probability test was used. Except for Strategy 6, the obtained p-values were either very small or very large, as shown in tables 7 and 8.

Note that data presented in tables 3 to 8 are obtained from the same two groups; they were a kind of "repeated measures". In this case, we should use the Bonferroni χ^2 statistic (Jensen et al., 1968), instead of the classical χ^2 statistic. It turned out that the Bonferroni statistic gave the same conclusions (except for Strategy 6, where p>.05) as the classical χ^2 statistic and Fisher's test on accepting and rejecting the null hypothesis.

From the twenty possible types of strategies and tactics, artificial and real protocols differed only for two of them (marked by "*"): Strategy 4 and Tactic 5.

Prot.	St. 2	St. 3	St. 4 *	St 5	St. 6	St. 7
IGGY	12	18	25	4	40	5
Real	10	20	9	7	35	4
p-value	.61	.65	.0003	.33	.03	.72

Table 7: Number of protocols (out of 40) in which the strategies were observed at least once

Prot.	Ta. 1	Ta. 2	Ta. 3	Ta. 5 *	Ta. 6	Ta. 7
IGGY	21	3	5	28	4	9
Real	26	9	6	9	5	4
p-value	.26	.06	.75	<.0001	.72	.13

Prot.	Ta. 8	Ta. 9	Ta. 10	Ta. 11/12	Ta. 13	Ta. 14
IGGY	39	5	8	7	8	7
Real	37	7	11	5	4	6
p-value	.37	.53	.43	.53	.21	.09

Table 8: Number of protocols (out of 40) in which the tactics were observed at least once

The last comparison between artificial and real protocols concerns the states of the mind in the first, penultimate and last configuration building episodes. It is clear that the diversity of the states of mind increases during the solving process. Table 9 gives the distribution of the states of mind.

[4] Jean-Marc Meunier

	1st epis.		Penult. epis.		Last epis.	
State of mind	IGGY	Real	IGGY	Real	IGGY	Real
Availability (A)	nil	nil	nil	nil	nil	nil
Compatibility (C)	nil	nil	nil	nil	nil	nil
Price (P)	10	9	1	5	nil	nil
Telephone (T)	nil	nil	1	0	1	0
AC	3	1	0	1	0	1
AP	24	25	24	28	11	19
AT	nil	nil	1	0	2	8
CP	1	0	2	1	1	0
CT	nil	nil	nil	nil	2	1
PT	nil	nil	2	0	15	5
ACP	2	5	6	1	4	5
ACT	nil	nil	nil	nil	2	1
CPT	nil	nil	2	0	2	0

Table 9: Number of states of mind at different episodes

In order to perform statistical comparisons, we combined the rows of Table 9 in the following ways:

- For the first episode, the χ^2 was calculated on a table with 2 columns and 3 rows: P; AP; CP+AC+ACP (the shared criterion was C);
- For the penultimate episode, the only combined table of observed data that allowed the use of a statistic test was a 2X2 table. Row 6 was left alone, and the remaining rows were combined;
- For the last episode, we chose the following combinations of rows: AP; CP+AC+ACP (the shared criterion was C); T+PT+AT+CT+CPT+ACT (the shared criterion was T).

For the 3 episodes, the differences between artificial and real protocols were not significant: $\chi^2(2)=.073$ (p=.96), $\chi^2(1)=2.32$ (p=.13), and $\chi^2(2)=4.30$ (p=.12), respectively.

Discussion

The results we obtained are generally good, since there were only four cases where the difference was significant. The results of statistical tests showed that, the p-value for a test was either very large or very small. It did not seem reasonable to appeal merely to the notion of *sampling error*. The differences concerned the number of episodes in the configuration building phase, the number of test phases, the number of protocols in which there was at least one application of Strategy 4 (select then phone strategy) or Tactic 5 (choice of the amplifier that was the most compatible with the configuration). Reasons must be found to explain some of these differences.

First, the IGGY'S protocols had more episodes in the configuration building phase than the real protocols. This difference can be attributed to an important difference between human subjects and IGGY as far as verbalisation is concerned. Unlike IGGY's reasoning, which is explicitly visible through the evolution of its internal state, the activity of the subjects is only known through their observed behaviour. Consequently there can be changes of episode within a configuration building phase that cannot be detected in the analysis of the real protocols, due to the lack of a proper verbalisation. On the contrary, in the case of IGGY, if all the conditions for an observation to be made are satisfied, then the observation is effectively made. In this respect, IGGY can be considered as a subject who verbalises all her/his actions. This explanation is coherent with the following finding. We counted the number of the verbal utterances in the real protocols that belonged to the 22 verbalisations we used in the model, and the number of observations in the artificial protocols. The mean number of verbal utterances were respectively 6.65 (S.D.=2.38) and 8.32 (S.D.=2.31). The p-value for the Student-t test was .002.

The second difference concerned the finding that human subjects were more likely to test the configuration when it was not yet a satisfying solution. This difference suggests that the simulated subjects were somewhat better appraisers than the human subjects. Thence, being better appraisers, the simulated subjects were more able to establish relationships between the amplifier to be chosen and a set of components already put in a column. Consequently, Tactic 5 was more often observed in the IGGY's protocols. However, IGGY and human subjects were equally efficient in detecting errors. These findings, together with the difference concerning the use of Strategy 4, suggested that the number of personality features we introduced as free parameters of IGGY was somewhat too small, and that these features were probably correlated.

CONCLUSION

We were interested in *complex problems* that belong to a semantically rich domain and which did not give from the outset all the information that was necessary to reach a solution.

The principal characteristics that differentiate our problem from the puzzle problems are: (i) the problem space is very large (more than one million nodes); (ii) the problem is more likely to be an *arrangement*[5] problem than a *transformation* one so that general heuristics such as means-ends analysis cannot be applied; (iii) the subjects must ask for information to find a solution. These characteristics are common to both our problem and design problems, although the latter are much more difficult, take more time, and are usually much more ill-defined.

In our experiment, we deliberately intently built the user interface in order to allow the subjects to use different ways of solving the problem, from a sophisticated reasoning mode similar to that of an expert in the domain of design, to a very reactive mode where the subject tries to build up one solution at a time, without considering the possible alternatives. It turned out that the majority of our subjects adopted an approach rather similar to that of the novice designers.

This result is coherent with studies about human reasoning, which demonstrated a general tendency to depart from *sophisticated* reasoning behaviour. For instance, research work on deductive reasoning and decision making has shown that people do not usually reason following an "apparently appropriate normative system" (Evans & Over, 1997, p.2) such as standard rules of logic or mathematical models. Several factors have been called upon to justify this departure from a normative behaviour, such as Simon's *bounded rationality* and *satisficing principles*, bias introduced by

[5] The goal is a state of the world that satisfies certain requirements. The anagram problem is typically an arrangement problem.

the way the subject builds up a representation of the problem situation, memory load, etc.

However, adult subjects cannot be dubbed as being incapable of sophisticated reasoning, because around 10% of the observed protocols showed an approach that was similar to the ideal strategy. For the remaining protocols, the fact that all of them reached a solution that met the goal requirements suggests that the subjects' behaviour can be considered as based on a *rationality of purpose* rather than a *rationality of process*. Evans and Over (1997) argued that the first kind of rationality is more generally and more spontaneously applied than the second one.

Finally, from the Artificial Intelligence point of view, although we did not find as much sophisticated reasoning and anticipative behaviour as we expected, this study brings out a number of interesting points. We have already shown (Chaignaud & Levy, 1996) that a parallel could be established between our cognitive model and recent trends in Artificial Intelligence such as knowledge compilation or constraint satisfaction.

We think that our model, and particularly the notions of phase, state of mind, strategy, tactic and personality, is general enough to be used in a whole class of problems that we have called *configuration problems*: variables have to be instantiated among a set of values that have to satisfy several constraints. Moreover the data of the problem is incompletely described. Possible examples include timetable problems in a school subjected to the constraint of availability of the classrooms, when the other obligations of the teachers are not known in advance; the travel agency problem where one has to schedule multimodal journeys to go from one point to another, and the availability of the seats is not known in advance. The dichotomy between normal and abnormal situations arises in most problems with incomplete information, and our system is able to manage two degrees of abnormality (simple errors and deadlock) and to react according to the situation.

By using the notions of personality and state of mind, our model accounts for observed individual differences that cannot be explained otherwise. Moreover, it captures the dynamic aspect of the problem-solving process.

REFERENCES

Agre, P. E. & Chapman, D. (1987). Pengi: an implementation of a theory of activity. *American Association for Artificial Intelligence*, Seattle, 268-272.

Ball, L. J., Evans, J. St. B., Dennis, I. & Ormerod, T. C. (1997). Problem-solving strategies and expertise in engineering design. *Thinking and Reasoning*, *3*, 247-270.

Bruner, J. S., Goodnouw, J. J. & Austin, A. (1956). *A study of thinking*. New York: Wiley.

Chaignaud, N. & Lévy, F. (1996). Common sense reasoning: experiments and implementation. *European Conference on Artificial Intelligence*, Budapest, 604-608.

Engelmore, R. & Morgan, T. (1988). *Blackboard systems*, Addison Wesley Publishing Company.

Evans, J. St. B. & Over, D. E. (1997). *Rationality and Reasoning*. Hove, UK: Psychology Press.

Goel, V. (1992). "Ill-structured representations" for ill-structured problems. *14th Annual Conference of the Cognitive Science Society*. Hillsdale, NJ: Lawrence Earlbaum, 130-135.

Guindon, R. (1990). Designing the design process: exploiting opportunistic thoughts. *Human Computer Interaction*, *5*, 305-344.

Hayes-Roth, B. (1993). Intelligent control. *Artificial Intelligence*, *59*, 213-220.

Jensen, D. R., Beus, G. B. & Storm, G. (1968). Simultaneous statistical test on categorical data. *Journal of Experimental Education*, *36*, 46-56.

Simon, H. A. (1973). The structure of ill-structured problems. *Artificial Intelligence*, 4, 181-201.

Turing, A. M. (1950). Computing machinery and intelligence. *Mind*, *59*, 433-460.

Visser, W. (1990). More or less following a plan during design: opportunistic deviation in specification. *International Journal of Man-Machine Studies*, *33*, 247-278.

Voss, J. F. & Post, T. A. (1988). On the solving of ill-structured problems. In M. T. H. Chi, R. Glasser & M. J. Farr (Eds.). *The nature of expertise*. Hillsdale, NJ: Earlbaum.

Symposia

Cognitive Models at Work Symposium

Symposium Aims

Cognitive models are used in the design of aircraft and industrial plant; operator tasking; user · interface design; and for operations research into the behaviour of complex sociotechnical systems. The purpose of their use is to account for human performance in shaping work environments, developing cognitive aids, evaluating systems and designs, and predicting the outcomes of courses of actions. These models come from a number of intellectual traditions, and the papers included here are from and across disciplines. Rather than focusing on a particular model, this symposium seeks to explore some of the uses to which cognitive models are put, to find which models are being used and to draw some conclusions as where advances has been made and the technical challenges still in front of the cognitive modelling community.

Simon Goss

Programme

1. Introduction
2. *Learning and Testing Cognitive Models*
   Simon Goss[1], Sam Waugh[1], Adrian Pearce[2], and Tim Menzies[3]
   [1]DSTO-Air Operation Division,
   [2]Computer Science-Curtin University
   [3]Computer Science- UNSW, Australia
3. *The Cognitive Cockpit: The Application of a Cognitive Model,*
   Simon Finne and Robert Taylor
   DERA Centre for Human Sciences;
   Farnborough, UK
4. *Military Applications of Cognitive Models with COGNET*
   Floyd Glenn
   CHI Systems, Inc., Spring House, PA, USA
5. *Exploiting Knowledge Engineering for the Construction of Cognitive Models*
   Nigel Shadbolt
   Psychology, Nottingham University, UK
6. Wrap-up

Learning and Testing Cognitive Models

Simon Goss (1), Sam Waugh(1), Adrian Pearce(2), and Tim Menzies(3)

(1) DSTO-Air Operation Division,
GPO Box 4331, Melbourne, 3001 Australia
{first.last}@dsto.defence.gov.au

(2) Computer Science-Curtin University
GPO Box 1987U , Perth, 6001, Australia,
adrianp@cs.curtin.edu.au

(3) Department of Artificial Intelligence,
School of Computer Science &Engineering, University of. NSW, Sydney, Australia, 2052,
tim@menzies.com

Introduction

In the military environment the physical systems components are there to facilitate operator mission objectives. While analysts have traditionally paid considerable attention to fidelity when modelling physical entities, the physical characteristics of system components are not sufficient to describe operational systems in sociotechnical environments; the human operators contribute significantly to systems outcomes [1]. In supporting operational usage after a capacity acquisition it is in the domain of mission parameters and operator procedures that the scope for change to improve performance lies. In operational research there has been a shift in focus from modelling an operator performing a task in an environment to modelling an entity with a social role performing actions in a dynamic social environment. This involves the recognition of the intentions of other entities. It could be said that the focus has shifted from computational theory of mind to computational theory of other minds.

Two aspects of modelling users are addressed in this presentation : the first, "grey box modelling" is applied to documenting a user's model of simulation software; the second concerns a method for learning to recognise the intentional behaviour of players or simulated agents in an agent-oriented virtual environment.

Testing Models

Grey box modelling, the process whereby a user by means of exploration develops a causal model of a partially understood system is the problem of legacy code maintenance and black box model commissioning. It is the process of acquiring expertise with a system to the point of function practicality. In our current work a method developed for the verification of knowledge based systems is applied to the testing and documentation of a developing user model of software [2-4]. The context is operations research where large models are used; often with large components externally sourced and less than well documented. Considerable investment of staff time is required in learning and using these systems. An explicit documentation of the mental model the user has of the system has significant potential as a guide and aid to the acquisition of expertise, and the retention of this expertise independent of staff movement.

Learning Plans

In this work the experimental aim was to demonstrate a method of constructing procedures from spatio-temporal data which describe action plans of agent/entities in a virtual environment [5-6]. These are required for testing candidate operator intentions against operator action history, and are interpretable as partial instantiations of intentionality. The capacity of situation awareness possessed by human operators in dynamic social systems requires is the recognition of plans whilst in execution in addition to than casual physical processes in train. A desirable incidental benefit is a summary method for the massive amount of data obtainable in a human-in-the loop simulation.

We explore this experimentally in the context of flight simulation, and offer a method for learning action plans. This requires three components: an appropriate ontology (model of operator task performance), an appropriate virtual environment architecture (accessibility of data and image generation databases) and a learning procedure (which relates the data stream to the domain ontology).

In simple terms, we are looking at the domain of circuit flight. We have a task analysis for circuit flight. The flight simulator has an authentic flight model for a PC9 aircraft, and a cockpit with generic throttle and stick controls. It also has a particular software architecture conferring special data recording properties. A relational learning technique is used to relate the data from the flight simulator to the task analysis. We build relations which describe generalised flight plan segments.

In practise these run in real-time and announce attributed plan segments *while* the pilot is executing them. This is a compelling demonstration of the feasibility of real-time recognition of intention in a user interface to an immersive virtual environment task. We assert that our results have wider significance and may form part of the foundation for the construction of agent-oriented simulations, and more broadly, virtual environments

References

[1] Ian Lloyd and Simon Goss , *Simulating Human Characteristics for Operational Studies*, Proc of Australian Cognitive Science Conference , 1997,.

[2] T. Menzies and S. Goss, *Applications of Abduction #3: "Black-box" to "Gray-box" Models* in Proceedings of AI in Defence Workshop, AI'95, DSTO general document GD-0077, 1997.

[3] Tim Menzies and Simon Goss Vague Models and Their Implications for the KBS Design Cycle, Proceedings PKAW '96: Pacific Knowledge Acquisition Workshop, 1996.

[4] Waugh S.; Menzies T.J; Goss, G. Evaluating a Qualitative Reasoner. Advanced Topics in Artificial Intelligence: 10th Australian Joint Conference on AI; Abdul Sattar (Ed.); ISBN 3-540-63797-4. Springer-Verlag, pp 505-514, 1997..

[5] Adrian Pearce, and Simon Goss, *Learning Action Plans in a Virtual Environment*., In L. J. Hettinger and M. W. Haas, editors, Psychological Issues in the Design and Use of Adaptive Virtual Interfaces, Erlbaum Publishers Inc., Hillsdale, N.J., USA, (to appear 1998.)

[6] Adrian Pearce, Terry Caelli, and Simon Goss, On *Learning Spatio-Temporal Relational Structures in Two Different Domains*, Third Asian Conference on Computer Vision - ACCV98 Hong Kong, January 1998.

The Cognitive Cockpit:
The Application of an Adaptive Cognitive Model

SE Finnie and RM Taylor
Centre for Human Sciences
F131, DERA Farnborough
Hampshire, GU14 6TD, UK
+44 1252 394282
{sfinnie, rtaylor}@dra.hmg.gb

Abstract

In an increasingly complex and automated aircraft environment, aircrew tasks are now more cognitive than physical in nature. This has led to interest in the requirements for cognitive quality in aircrew systems, and the need for engineering principles to guide the design of cognitive tasks. In symbiotic systems where both human and system cognitive quality is necessary for effectiveness, research is needed into the requirements for cognitive control (strategic, opportunistic) and compatibility (usability, intuitiveness). Such joint cognitive systems require reliable, and adaptive, cognitive models.

DERA CHS is currently developing such a cognitive model which will provide guidance on pilot-system dialogue structures, and cognitive task specification. The model attempts to encapsulate the relationship between human and machine at different levels of control, communication, awareness, and behaviour, and draws upon contemporary psychological theories such as: Rasmussen (1983); Hollnagel (1996); Taylor (1988). The model will provide guidance on the nature of the relationship between human and system. For example, the model will indicate that at no time should the system remove the pilot's control. Instead, a process of critiquing is preferable where the system is able to critique the pilot's errors, and similarly, the pilot is able to critique, and improve, the Cognitive Cockpit's advice. This paper outlines the adaptive cognitive model and the factors that ensure that it is a practical, applicable, framework for implementing automation in the DERA CHS Cognitive Cockpit.

Rasmussen J (1983), *Skills, Rules, and Knowledge: Signals, Signs, and Symbols, and other distinctions in human performance models,* IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-13, No 13, 3.

Hollnagel E (1996), *Modelling the Orderliness of Human Action*, In "Cognitive Engineering in the Aviation Domain", Amalberti R, and Sarter N (Eds.)

Taylor MM (1988), *Layered Protocols for Computer-Human Dialogue I: Principles,* In "International Journal of Man-Machine Studies", 28, 175-218.

Military Applications of Cognitive Models with COGNET

Floyd Glenn
CHI Systems, Inc.
716 N. Bethlehem Pike
Spring House, PA, USA 19002
215-542-1400
floyd_glenn@chiinc.com

ABSTRACT

This paper presents an overview of three cognitive models developed with the COGNET (for Cognition as a Network of Tasks) methodology and toolset. The examples illustrate the broad range of applications for which such models are suitable. They include a model for an air defense gunner which was developed for the purpose of crewstation design evaluation. The second example is a set of models for the watchstanders in an advanced ship's combat information center which are being developed as part of an embedded intelligent training system. The last example is a model of an airborne anti-submarine warfare sensor operator which is being developed to support an intelligent interface for the sensor operator.

Keywords

cognitive model, design evaluation, training, intelligent interface, COGNET, CIC watchstander, air defense gunner, sensor operator.

INTRODUCTION

Development of a cognitive model for a person operating a complex system is always a daunting effort. At a minimum, the cognitive modeler must define the task procedures for system operation, the complete knowledge base that is relevant to performance of these tasks, including both general and task-specific knowledge, and the various component performance models which characterize each aspect of human task performance. Construction of these cognitive models typically entails use of specialized AI programming languages such as LISP and accordingly requires the support of highly trained computer scientists. The COGNET methodology and toolset for cognitive modeling (Zachary, Ryder & Hicinbothom, in press; Zachary et al., 1992) has been developed in order to facilitate the development of cognitive models with a minimum need for support from such computer specialists. COGNET offers an integrated model development environment with a graphical interface for goal and task representation. This paper presents an overview of the COGNET toolset and descriptions of three distinctly different types of application of COGNET for military systems. The three COGNET applications include the primary alternatives that have been conceived for applications of executable cognitive models — (1) detailed performance prediction for design evaluation, (2) an embedded cognitive model for an intelligent training system, and (3) an embedded cognitive model for an intelligent operational interface.

DESIGN EVALUATION - GUNNER MODEL

The application of COGNET for design evaluation concerns the development of a simulation model for the operator of the U.S. Army's mobile air defense weapons system known as Avenger. The Avenger is an operational mobile Forward Area Air Defense (FAAD) element consisting of a High Mobility Multipurpose Wheeled Vehicle (HMMWV) having a rotatable turret and eight ground-to-air Stinger missiles. Avenger is manned by a driver and operated by a gunner. The gunner sits in the turret where he searches for air targets through a transparent windscreen and also with a forward-looking infrared (FLIR) display. Upon detecting a target, the gunner aims the turret optical site at the target by rotating the turret using a control yoke, and then, upon verification of a hostile identification, a missile can be fired using control buttons on the yoke. The simulation of the Avenger gunner was developed to operate in the software environment of a simulation-based trainer for the Avenger system, called the Avenger Institutional Conduct of Fire Trainer (Avenger ICOFT). This simulation effort was originally conducted in order to enable simulation-based evaluation of contemplated modifications to Avenger, but interest has also developed in the potential use of this simulation for DIS applications. The gunner model includes distinct component performance models for visual search, target detection and identification, target tracking, and associated equipment operation and decision making. The model was developed initially through a task analysis of gunner performance in the ICOFT and later through collection of detailed performance time data in the ICOFT to use for model parameter calibration.

TRAINING APPLICATION - CIC MODELS

The COGNET application for intelligent training involves a series of models to simulate both the behavioral and cognitive activities of the watchstanders comprising the Anti-Air Warfare (AAW) team in the Combat Information Center (CIC) on an Aegis-based Cruiser (see Zachary et al., 1997 for a more detailed summary). This was done to construct simulation-based Advanced Embedded Training Systems for shipboard team training. The various members of the AAW team must identify and appropriately respond to air tracks so as to maintain the self-defense of own-ship and any protected assets. This is a particularly difficult task in complex geo-political environments characterized by low-intensity conflict such as the Persian Gulf. Models for four different watchstanders have been developed — the AAW Coordinator (AAWC), the Tactical Action Officer (TAO), the Electronic Warfare Supervisor (EWS), and the Identification Supervisor (IDS). Each simulation model is able to:

* operate the actual watchstation (or a high-fidelity simulation) in the same manner and with the same level of performance as a human expert;

* generate and respond to voice interactions with other members of the AAW team and other CIC personnel;

* reason about and solve tactical problems as they arise; and

* take appropriate tactical actions.

The simulations were built to support shipboard team training based on the embedded training simulation capability of the Aegis Weapon Control System. While an Aegis embedded simulation is running, the behavioral models work the simulation scenarios in parallel to human trainees, generating expert level behaviors that are used as a dynamic benchmark for diagnosing both the behavioral and cognitive performance of the trainees. This diagnosis is then used to provide (real-time or deferred) feedback. Other direct applications of these models include supporting mission rehearsal and tactics development.

INTERFACE APPLICATION - SENSOR OPERATOR MODEL

COGNET has been used in a planned design for an intelligent interface for the U.S. Navy's new SH-60R multi-mission helicopter, designated as the Task-Oriented Interface Layer (TOIL). TOIL is envisioned as separate from the basic crewstation interface planned for the SH-60R and is intended as an alternative means for the sensor operator SO to accomplish essentially all functions provided by the crewstation. TOIL is offered as an option to the SO who may use it as much or as little as seems appropriate given the knowledge and skills that that operator has with the crewstation and tactical domain. Thus, TOIL represents an alternative interface layer for operator interaction with the system. TOIL is implemented in the form of various menu and data windows that appear in the data strip region of the SO's mission display. TOIL is structured to provide interface options and guidance that are specifically tailored to the momentary tactical and task context, and is hence task-oriented. Additionally, TOIL will incorporate intelligent agent software which will enable automation of some interface or tactical functions as part of TOIL.

CONCLUSIONS

The three example applications of COGNET described above provide an indication of the diverse ways that cognitive models are beginning to contribute to complex military systems. With the emergence of such "main stream" applications, it is becoming increasingly important to provide tools and methods to facilitate such model developments. COGNET represents a candidate model development environment that is designed to support modeling of the kinds of real-time, multi-tasking jobs involved in military crewstations such as are described here, but it is also designed to be relatively easy to use by people with minimal special computer training. Clearly, there are still many further enhancements and aids that are feasible and warranted for COGNET, as well as other similar cognitive modeling environments, to improve their usability. Two such areas of particular current interest are a graphical visualization tool for the COGNET declarative knowledge base (i.e., blackboard) and an aid for COGNET-oriented knowledge engineering (e.g., for guiding interviews and supporting information management).

ACKNOWLEDGMENTS

The applications of the COGNET model described in this paper were supported by contracts N61339-97-C-0023 and N61339-96-C-0055 with the Naval Air Warfare Center Training Systems Division, by contract N00421-97-C-1133 with the Naval Air Systems Command, and contract DAAH01-95-C-R124 with the Army Missile Command. All views expressed in this paper are strictly those of the author.

REFERENCES

Zachary, W., Ryder, J., & Hicinbothom, J. (in press). Cognitive Task Analysis and Modeling of Decision Making in Complex Environments. In J. Cannon-Bowers & E. Salas (Eds.), *Decision Making Under Stress: Implications for Training and Simulation*, Washington, DC: APA.

Zachary, W., Ryder, J., Hicinbothom, J., Bracken, K. (1997). The Use of Executable Cognitive Models in Simulation-based Intelligent Embedded Training. *Proceedings of Human Factors Society 41st Annual Meeting*. Santa Monica, CA: Human Factors Society, pp. 1118-1122.

Zachary, W., Ryder, J., Ross, L., and Weiland, M. (1992) Intelligent Human-Computer Interaction in Real Time, Multi-tasking Process Control and Monitoring Systems. in M. Helander and M. Nagamachi (Eds.). *Human Factors in Design for Manufacturability*. New York: Taylor and Francis, pp. 377-402.

Modelling Conceptual Changes in Mechanics: An Interdisciplinary Perspective

Stella Vosniadou[1], Christos Ioannides[1], Aggeliki Dimitracopoulou[1], Daniel Kayser[2], Marc Champesme[2], Floriana Esposito[3], Giovanni Semeraro[3], Donato Malerba[3], Stefano Ferilli[3]

ABSTRACT

The work reported here has been led in a collaboration which took place in the framework of "taskforce 1: Representation Change" of a european project "Learning in Humans and Machines" sponsored by the European Science Foundation during the years 1994-97.

This interdisciplinary and international collaboration has gathered Psychologists and Education Scientists, who collected and analyzed data about the knowledge of students in elementary mechanics, and who hypothesized mental models explaining the data; Computer Scientists specialized in Knowledge Representation who designed a language tailored to express the above models; and Computer Scientists specialized in Machine Learning, who investigated the behaviour of two systems on (part of) the data collected, and evaluated the relevance of the formal study of theory revision to the modelization of the conceptual changes that take place in students.

Keywords

conceptual change, force, knowledge representation, machine learning, mechanics, mental models, validation of cognitive models

INTRODUCTION

We report on a collaborative and interdisciplinary work, which took place in the framework of a project called "Learning in Humans and Machines" sponsored by the European Science Foundation. The objective of the authors in this research was to effectively bring together the know-how from the fields of cognitive psychology and machine learning in view of the fulfilment of two main goals. The first, mostly relevant for cognitive psychology, is to propose a theory of the development of knowledge acquisition in mechanics, with the help of computational models, clearly formalised and precisely testable. The second one, relevant for machine learning, is to obtain powerful guidelines for a more effective design strategy of learning systems, starting from the very basic issue of what knowledge they should handle and how to represent it.

All the research works that are presented have been conducted from the same data that has been collected on Greek students of various ages concerning their concept of force. The format of the present paper will be as follows. It will start with a short presentation of the empirical data which led to the identification of a small number of mental representations of force in students ranging in age from 5 to 15 years of age. It will continue with a presentation of a computational model which tries to reproduce the conceptual changes that take place when children develop the concept of force with reference to the theoretical framework proposed by the psychology team. We will then proceed with another short presentation of a process model of the solution of three problems in mechanics by elementary school students before and after a six week experimental program of instruction in mechanics. It will be followed by a presentation of a computer model designed to represent accurately the characteristics of the psychological model.

MENTAL MODELS OF FORCE

(Christos Ioannides & Stella Vosniadou)

The purpose of the research reported in this section was to investigate the development of the concept of force in young children and propose a theoretical framework within which we can explain this development.

A total of 105 children ranging in age from 5 to 15 years were interviewed individually using a questionnaire consisting of 27 questions. Each question referred to a drawing depicting objects of different weights and sizes (e.g. big stones and big balloons vs. small stones and small balloons), some stationary and some moving, and were asked about the kinds of forces that were exerted on these objects. Following a methodology developed by Vosniadou and Brewer (1992; 1994), we were able to identify eight mental models of force which were used consistently by 70.6% of the students in order to answer the questions. The mental models of force are presented in Table 1. The younger children in our sample used mental model 1, according to which there is an *internal force* within big and heavy objects regardless of their kinetic state or position. Older children's responses (about 9 - 10 years), could be mostly explained by assuming that they had used mental model 4, according to which there is an *acquired force* only within moving objects. This force was thought to be imparted to the objects by an external agent which set them in motion and serves to explain this motion. Mental models 2 and 3 were based on combinations of the above two interpretations of force (internal and acquired). In contrast to model 4, the

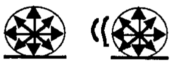
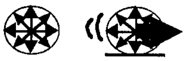
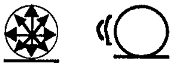
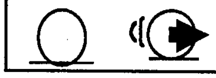
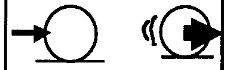
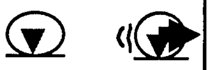
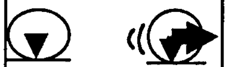
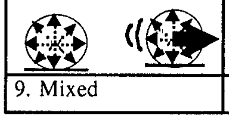
[1] {svosniad@compulink.gr - cioannid@atlas.uoa.gr - adimitr@atlas.uoa.gr} - Department of Philosophy and History of Science - National and Capodistrian University of Athens - 37, John Kennedy street - Athens 161 21 - Greece

[2] {Daniel.Kayser@lipn.univ-paris13.fr - Marc.Champesme@lipn.univ-paris13.fr) - LIPN - Institut Galilée - Université Paris-Nord - Av. J.-B. Clément - 93430 Villetaneuse - FRANCE

[3] {esposito@lacam.uniba.it} - Dipartimento di Informatica - Università degli Studi di Bari - via Orabona, 4 - 70126 Bari - Italy

children who used mental model 5 believed that there is a *force of push or pull* exerted on objects pushed or pulled by an agent (even in the absence of movement). The same interpretation of force is also present in model 7. Most of the children who had received instruction in mechanics developed mental model 6 according to which the *force of gravity* is exerted on all the objects. The force of gravity model was usually added to an already existing *acquired* force model. Various alternative interpretations of the word gravity have been identified. For example, some children believe that the force of gravity increases with the stability of the objects, others that gravity increases as the distance of an object from the ground becomes greater.

Table 1: Frequencies and percent of mental models of force as a function of grade.

Mental Models of Force	Kind/ garten	4th grade	6th grade	9th grade
1. INTERNAL FORCE: There is an Internal Force within stationary and moving heavy objects 	40%	6.7%	0%	0%
2. INTERNAL and ACQUIRED FORCE: There is an Internal Force within stationary heavy objects - There is an Internal and an Acquired Force within heavy objects that are moving 	20%	26.4%	20%	0%
3. INTERNAL FORCE IN STATIONARY OBJECTS: There is an Internal Force within stationary heavy objects only 	13.3%	6.7%	0%	0%
4. ACQUIRED FORCE: There is an Acquired Force within moving objects only 	0%	6.7%	30%	10%
5. ACQUIRED FORCE and FORCE OF PUSH/PULL: There is an Acquired Force within moving objects - Force from an external agent on all objects that are pushed or pulled by the agent 	0%	0%	13.3%	16.5%
6. GRAVITATIONAL FORCE and ACQUIRED FORCE: There is the force of *gravity* on all stationary and falling objects - There is the force of *gravity* and an *acquired* force within moving objects 	0%	3.3%	0%	39.6%
7. FORCE OF PUSH/PULL: There is a force only on objects that are pushed or pulled by an agent 	0%	0%	0%	3.%
8. SUSPENDED and ACQUIRED FORCE: Objects at high and unstable positions have "more force" then objects at a lower or more stable position (Suspended Force)- There is an *Acquired Force* within moving objects 	6.7%	16.5%	13.3%	3.3%
9. Mixed	20%	33.3%	23.3%	26.4%

Mental models are assumed to be formed as children attempt to explain their everyday observations and information (verbal or other) they receive from the culture under the constraints of certain ontological and epistemological presuppositions, such as that force is a property of physical objects, and that the motion of an inanimate object requires an explanation in terms of a causal agent (see Figure 1). The process of conceptual change seems to be a slow affair that proceeds through the gradual suspension or revision of well entrenched beliefs and presuppositions. For example, the older children seem to have differentiated weight from force. Nevertheless, the presupposition that force is a property of physical objects and that the motion of physical objects requires an explanation, do not seem to have been changed in the conceptual system of the 9th grader, despite the fact that these students have been exposed to systematic instruction in Newtonian mechanics.

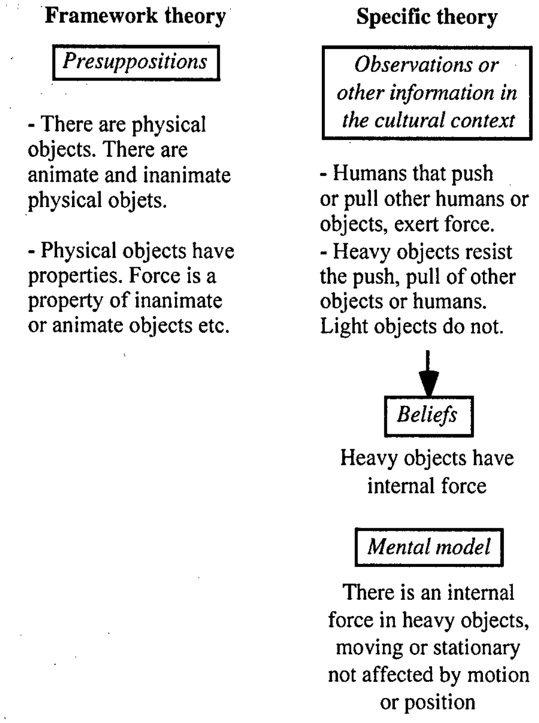

Framework theory
Specific theory
Presuppositions
Observations or other information in the cultural context
- There are physical objects. There are animate and inanimate physical objets.
- Physical objects have properties. Force is a property of inanimate or animate objects etc.
- Humans that push or pull other humans or objects, exert force.
- Heavy objects resist the push, pull of other objects or humans. Light objects do not.
Beliefs
Heavy objects have internal force
Mental model
There is an internal force in heavy objects, moving or stationary not affected by motion or position

Figure 1: Hypothesised conceptual structure underlying the internal force mode.

ELABORATION OF A MACHINE LEARNING MODEL

(Floriana Esposito, Giovanni Semeraro, Donato Malerba and Stefano Ferilli)

From the Machine Learning perspective, the research focuses on the elaboration of a computational model which tries to reproduce the conceptual changes that take place when children develop the concept of force with reference to the theoretical framework proposed by the psychologist team.

A fundamental characteristic in the use of mental models concerns the possibility of verifying the general validity of a reasoning process based on examples by generating a sequence of significant examples and by applying revision procedures on the models (Johnson-Laird, 1983). Revision processes triggered by inductive mechanisms are an important aspect of learning. The research focused on the elaboration of a computational model of learning based on theory revision. The main objective of the work was to prove the validity of two particular Machine Learning systems: whether they are able to simulate the very complex phenomena related to the process of acquiring concepts of naive physics by creating these conceptualizations and refining them on the ground of new evidences. This could be useful in order to supply an artificial experimental laboratory where to test some of the mental models proposed by the psychologists, by observing the variations in the behaviour of the computational models, monitoring the process of concept acquisition and refinement.

The proposed computational model considers learning as a process of formation and revision of a logical theory, where a logical theory is viewed as a set of conditions that are necessary and sufficient to explain a number of observations in a given environment. To be useful a theory must be able to explain past events and also predict future situations in the same environment.

A set of concept definitions constitutes a theory: theories are represented as sets of facts and rules, both strict and defeasible, sufficient for proving or explaining how an instance of a concept meets the concept definition. The instances from which a theory is inferred are called the training examples: these may be positive or negative. If we assume that the only source of knowledge available is represented by a set of examples and no prior knowledge can be exploited, the process of formulating a new theory is bound to be progressive. Starting from contingent observations, it is not possible to define concepts that are regarded as true. New observations can point out the inadequacies in the current formulation of the concepts. In such a case, a process of theory revision should be activated. Revisions of a logical theory are caused by a shift in the language and a change in the number and meaning of the involved concepts. In the proposed computational model the theory is refined incrementally: this is necessary when either incomplete information is available at the time of the generation of the initial theory or the nature of the concepts evolves dynamically.

Artificial learning systems can be roughly subdivided into batch and incremental, depending on whether all the examples used to train the system are completely available at learning time (batch) or not (incremental). Incremental learning systems maintain the inferred set of concept definitions consistent with all data (examples or observations) and have to store all previous data as soon as any backtracking is involved. In order to simulate the cognitive models of conceptual change in children learning elementary dynamics, two Machine Learning systems were used: ATRE and INTHELEX. The former is a batch system, while the latter is a fully incremental learning system.

The aim of the study was to see whether learning systems which learn from positive and negative examples by inductive inferential mechanisms could reproduce the changes in the concept of force observed in children. It has been suggested that children develop their concept of force on the basis of their interpretations of observations and information from their cultural background. Given some empirically derived hypotheses about the development of the concept of force, it was possible to extract the kinds of observations and/or information that are needed for the development to take place. These observations were used to validate the inferential power of the above mentioned learning systems.

Two experiments were run. In the first experiment, since ATRE can realize a shift in the representation language, the aim was that of discovering whether the system was able to relate the concept of force corresponding to "internal force in stationary and moving objects" to that corresponding to "acquired force in moving objects only". For humans the problem of learning dependent concepts is related to the possibility of having an ontology. For machine learning systems the two ways of solving this problem are to supply the system a graph representing the concept dependency or to leave the system discover the dependency while it learns the concepts. ATRE uses both the approaches and some interesting results have been obtained related to the autonomously defined order by which it generalizes the concepts.

The second experiment concentrated on the process of theory revision; INTHELEX was used to emulate the transitions occurring in the human learning process when, starting from an empty theory and providing just an observation a time, it is possible to model and to monitor the refinement process of a theory. Some initial interesting results have been obtained although a direct comparison with the children acquisition mechanisms is not possible.

The batch system allowed us to point out how the formulation of the naive concepts of force is based in part on everyday experience, observations and verbal information and to prove that the dependence between some basic concepts of force can be modeled by a shift in the representation language. The incremental learning system, compared to the batch learning system, seems to be more accurate in performing the conceptualization process, for two basic reasons:

a) changes of the initial theory caused by a new observation go through a process of refinement and it is not necessary to re-start the whole learning process from the beginning when a new instance is presented;

b) it can take into account temporal relations albeit in a very simplistic way.

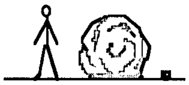
Both learning systems were able to produce from examples concepts related to the two models of "internal" and "acquired" force which were found in the developmental studies, although it is clear that students create their concepts only on the basis of observations or only being told about "force". The two systems tried to develop the two models of force through generalization and specification mechanisms. This may be compared with the phenomenon of "tuning" in conceptual change: indeed, both systems try to maintain coherence in the logical theory through their operators. This is an initial very simple form of conceptual change, although only a process of restructuration of knowledge should be considered a real conceptual change.

A PSYCHOLOGICAL PROCESS MODEL OF THE SOLUTION OF MECHANICS PROBLEMS BY ELEMENTARY SCHOOL STUDENTS

(Stella Vosniadou, Christos Ioannides and Aggeliki Dimitracopoulou)

The present project is based on collaborative and interdisciplinary work with a team of computer scientists from the LIPN (Daniel Kayser and Marc Champesme). The psychology team worked on a model that explained the solution of mechanics problems by elementary school children while and the computer science team validated this model by constructing a computer program. In previous work (Ioannides and Vosniadou, 1993; submitted) we used the theoretical model and methodology described in a series of studies on knowledge acquisition in astronomy (Vosniadou and Brewer, 1992; 1994; Vosniadou, 1994), to investigate the development of the concept of force. The results showed that it is possible to classify approximately 70% of the students in our sample as making use of one relatively well-defined mental model of force which they used in a logically consistent way to answer a number of questions about force. More specifically, six mental models of force were identified. These models were used in different frequencies by students ranging in age from 5 years (kindergarten) to 15 years (9th grade). These models are described in the previous section "Mental models of force".

In the present work we used these models to see if they could explain 5th grade students' responses to problems in mechanics, such as the one presented in Figure 2.



These two stones are just standing there. Is there a force exerted on them ?

Figure 2: Question 1.

The results showed that children's responses could be explained by assuming that the students used one of four models of force. When they gave responses such as "Yes, a force is exerted on the stones because they are big/heavy, etc." we assumed that they used the *internal force* model. On the contrary, if they said "No force is exerted on the stones because they do not move", we assumed that they used the *acquired force* model. Some students said that there is no force exerted on the stones "because the man does not push them". We assumed that these responses indicated use of the *push/pull model*, according to which a force is exerted when an (usually animate) object pushes or pulls another (usually inanimate) object. Finally, some students said that the force of gravity is exerted on the stones (*force of gravity model).* Students' responses to question 1 and the assumed mental models assigned to their responses are presented in Table 2.

Table 2: Students' responses to question 1.

	Response types	**Assumed mental model**	**%**
1.	Yes, a force is exerted on both stones because they are big/heavy they have weight/force	Internal force	18.4%
2.	No force is exerted on the stones because they are not moving	Acquired force	26.5%
3.	Yes, a force is exerted on both stones because the earth pulls/attracts them	Force of gravity	20.3%
4.	No force is exerted on the stones, because the man does not push them/exerts effort.	Push/pull	30.6%
6.	Other		4.2%

The results of this study also showed that the above mentioned models of force were not mutually exclusive and that the probability of activating them was influenced by the verbal and pictorial content of the specific questions asked. There were noticeable changes in the frequency of use of the different models in the different questions by the same subject population. The co-existence of different models of force in the same subject raised the issue of internal consistency. In previous work (Vosniadou & Brewer, 1992; 1994; Ioannides & Vosniadou, submitted) we have argued that students are consistent in their use of not more than one mental model of the earth, of the day/night cycle, or even of force. Are the present findings contradictory to our previous claims?

We believe that it is possible to explain the present findings if we assume that in the conceptual system of the 5th grader force is categorised differently for animate versus inanimate objects, as shown in Figure 3. If the child considers the question to apply to animate objects, then the *push/pull* model is used. If, on the other hand, the question is interpreted to belong to animate objects, the *internal* or *acquired force* models are used. Such an interpretation would make it possible for the same child to use either the "animate" or the "inanimate" models of force in different contexts, but not in the same context. Our results confirmed this prediction.

The possibility of internal inconsistency still is possible, however, in the case of use of the two inanimate models of force even in different contexts. However, an examination of the data showed that only one child made use of both the *internal* and the *acquired* models of force (the internal force model in questions 1 and 2, and the acquired force model in question 3). All the other children were consistent in their use of only one "inanimate" model of force.

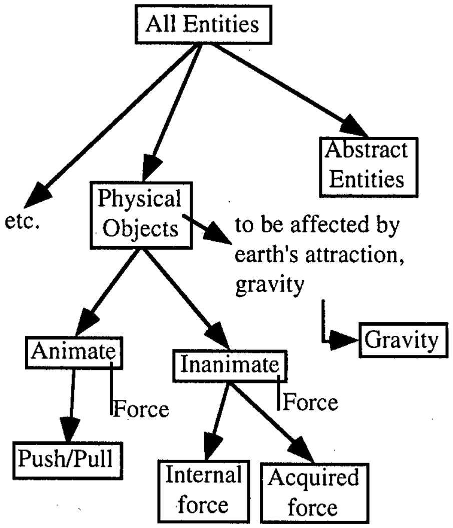


Figure 3: Assumed Categorization of the Concept of Force (for elementary school children).

The above leave unsolved the problem of how the mental model of the force of gravity is used. It appears to us that the gravity model comes through instruction to be added to the existing conceptual system of the 11 year old child and to be interpreted to apply to physical objects in general. Thus, the gravity mental model can theoretically take the place of any animate or inanimate model of force. When contextual cues lead to the activation of the gravity model, the search stops there and the other mental models of force are not utilized. We understand that this is a very preliminary treatment of the notion of gravity. We know from previous work that there are various misconceptions of gravity. These issues are further investigated in ongoing work.

To conclude, force can be categorised in different places in the conceptual system of a 5th grader. It can appear under inanimate objects either as an inherent internal property (internal force) or as an acquired property (acquired force). It can appear under animate objects as the force exerted by a person's push or pull. Finally it may appear as a property of physical objects to be affected by the earth's attraction (gravity). These alternative representations of force become available as information comes through observation and from the culture in the form of systematic or unsystematic instruction. In previous work we described some of the beliefs and presuppositions about force that underlie these representations. In the present work we note that the different representations of force are associated with different contexts of use. Depending on the nature of the question and on the context, the student activates selected pieces of his or her prior knowledge to eventually create a specific mental representation of force on the basis of which he or she provides a response.

We believe that this work succeeds in capturing important aspects of the concept of force in young children, both in terms of how it is related to assumed underlying beliefs and presuppositions and in terms of its relationship to other concepts and categories.

A COGNITIVE MODEL OF ELEMENTARY SCHOOL STUDENTS' SOLUTION OF THREE PROBLEMS IN MECHANICS

(Daniel Kayser and Marc Champesme)

Modelling the knowledge state of students is an important objective for theoretical and practical (e.g. pedagogical) reasons. The model needs be validated and the best **validation** consists in implementing it and run the computer program on various questions in order to check whether the answers are identical - or at least, analogous - to those provided by the students.

In this section, we report on the experiment described in the previous section. The data have been analyzed by Cognitive Psychologists and the resulting models have been implemented jointly with Computer Scientists specialized in Knowledge Representation.

The Language

Recent work in Artificial Intelligence shows that the most difficult problem is to find appropriate trade-offs between the **efficiency** (of the inference procedures) and the **expressiveness** (of the representation language). Therefore, in this research, we attempt to **tailor the expressiveness to the exact needs** of the model, care being constantly taken that the algorithms using the knowledge represented remain tractable.

Early works in **Knowledge Representation**, such as KRL (Bobrow & Winograd, 1977; Lehnert and Wilks, 1979) also originated from a collaboration between AI and Cognitive Psychology. But their purpose was more ambitious, because they aimed at a general framework for knowledge, therefore requiring maximal expressiveness, while we aim here at the minimal expressiveness compatible with the data.

The main line of further research (e.g. the KL-ONE family of languages (Brachman & Schmolze, 1985) followed by terminological and description logics) has emphasized on formal limitations in expressiveness in order to remain compatible with polynomial-time inferential mechanisms (a synthesis can be found in (Gottlob, 1996)).

More recently, research concerned with biological plausibility, as e.g. (Shastri, 1993) — notice being taken that biologists contest the relevance of this work, see discussion following (Shastri & Ajjanagadde, 1993) — have opened other insights in the efficiency vs. expressiveness trade-off. Papers by Fahlman (1979) and Shastri might have inspired very indirectly the present study.

Terminological Component

Concepts are structured in an ordinary "is-a hierarchy", with multiple inheritance. Relations between concepts are represented by roles, which may have cardinalities. Less common, but still very important (e.g. in order to define at least a weak notion of consistency), is the ability to express the disjointness of concepts. Obviously, every statement of the language is translatable into first-order predicate logic, the reciprocal being false.

Assertional Component

The student is given a text from which (s)he is supposed to build a representation. For example, a sentence such as: "*a man pushes a stone*" is assumed to create an instance M of *man*, an instance S of *stone*, and an instance P of *push* having as arguments respectively M and S.

The assertion of an entity may be direct (entity supposed to be created while reading the text) or indirect (existence asserted as the consequent of an inference rule, see below).

Inference Rules ; Inference Engine

The students also entertain beliefs of the form, e.g. "when an agent xxx, then yyy"; this corresponds to the classical notion of inference rules.

Representing Several Models

Every piece of knowledge belongs to one or several of the mental models identified (see second section). A large part of the "is-a hierarchy" is model-independent (a stone is a physical object in every model), but some critical areas do depend on the model, e.g. the ontological nature of *force.*

We therefore associate to the internal representation of every concept, role, and rule, a **vector of boolean values**. The dimension of the vector is the number of models identified (currently, a dozen of them). Technically, we first provide the computer with the list of the names of all mental models. Each name is assigned an index in the vector. Then comes the description, in the language of the information (terminological component and inference rules) supposed to be valid in every model. It is compiled and stored with a vector filled with "true". We then repeat a sequence composed of a list of the names of the model(s), followed by statements considered valid only in the models named in the list; the boolean vector accordingly sets to "true" only the corresponding indices. Once all descriptions have been processed, we begin a "*session*" intended to simulate the behaviour of a student.

Implementing the Models in the Language

The implementation of the psychological models has been a long process with several feedbacks between Computer Scientists (CS) and Cognitive Psychologists (CP).

The first reason is that, from the CS point of view, a large part of the CP theories remains implicit, either because it constitutes the common knowledge assumed in the cultural background of CP, or because CP do not consider making it explicit as a valuable effort.

Another reason is that CS implementation resulted in making some aspects of the psychological theory more explicit, raising new important questions which needed be answered without ambiguity, and in some cases this led to some changes in the modelization (cf. subsubsection "Adding Psychologically Essential Features" in this section).

Refining the Ontology

Implementation first requires, once the representation formalism is designed, to translate the psychological theory into that formalism. Now the theory was initially expressed in very heterogeneous forms, ranging from rather general theoretical statements to concrete responses of students in natural language extracted from the experimental data.

The first proposed implementation was strongly guided by the most explicit parts of the theory. Therefore, it corresponded to a rather literal interpretation of the psychological data: many concepts were created in an attempt to capture all subtleties of the psychological models. In view of this preliminary modelisation, CP's feedback led to a pruning of the hierarchy of concepts, resulting in a clarification of which concepts were the most important, and what they meant.

After this clarification, all concepts were classified into three main categories: `physical objects`, which denote the concrete objects of the real world (e.g. human, stones...), `propositions`, which express statements about physical objects and `abstract entities` like measures which are neither concrete objects, nor statements about physical objects.

After these clarifications were completed, only minor changes revealed necessary in the ontology.

Adding Psychologically Essential Features

During the refinement of the first implementation, it turned out that some characteristics of the psychological models of the students were not represented accurately. As these features were considered as essential from the CP point of view, the CS had to modify their proposal.

This fact must be pointed out as an important result of this work, since these features were not explicitely stated in the initial model provided by CP, and would perhaps remain unnoticed otherwise. Another important point is that, although the representation issues constitute in general difficult problems for AI research, a careful analysis of the psychological model showed that several features could eventually be represented, at least in this case, in a rather simple way.

Validation

Internal Validation

A compiler transforms the language into an internal form, performs several syntactic verifications (e.g. checks the well-formedness of the chain of roles), and goes beyond: using **partition** and **exclusive** statements, it checks that an entity does not inherit from two entities declared as incompatible with each other. Such checks proved useful to point at problems that were overlooked when writing the models.

A student can shelter simultaneously inconsistent beliefs, but in a given situation, (s)he should not use beliefs generating inconsistencies. Therefore, during a session, care is taken that every newly created entity is compatible with the is-a hierarchy, and obeys the cardinality restrictions declared in the **relation** statements.

External Validation

The above controls are more debugging aids than genuine validation. It is by far more important to compare the output of the program with the behaviour of a student supposed to work under the model(s) selected for a given session.

Obviously, some differences are irrelevant, as are also some similarities between student and computer answers. What matters is whether, for every situation in the domain of investigation, the computer outputs a result considered as plausible from a student supposed to use the corresponding model(s). Of course, this judgment is theoretically biassed, since the models identified exist only in the theory. A better test, which has been used in (Chaignaud, 1996), consists in coding the student reactions in a way formally indiscernible from computer outputs, and to evaluate statistically whether experts are able to discriminate between human and computer protocols. Even this method is not completely immune to criticism. Anyhow, validating cognitive models raises deep epistemological issues, which we are not willing to develop further here.

Model Selection

After having declared which model(s) S the student is supposed to have access to, we describe the situation as a list of entities.

Introducing entities triggers inference rules. The information relevant to this process (both the existence of "is-a" links propagating the search for the rules, and the rules themselves) is indexed by the set M of models in which it is assumed to hold. Three cases are to be considered:

- M and S are disjoint: nothing happens;
- M contains S: the information is used;
- $\emptyset \subset M \cap S \subset S$; here, we need to know more about the influence of the context over the behaviour of the student. The only empirical data at hand being probabilistic, we select at random, obeying to the probabilities measured by CP, the (unique) model in which the student is assumed to reason in this case, and the decision of using or not the information is taken accordingly.

CONCLUSION

This research had two kinds of benefits:

- globally, models of the knowledge of students have been hypothesized, specified in a precise way, tested, and machine learning systems have been run in order to reproduce the acquisition of some concepts;
- locally, each team has taken advantage of the presence of the others in the following way:
    - the Psychologists have been forced to refine their models, and to resolve some inconsistencies which were not perceptible before the Computer Scientists had to implement them;
    - the analysis of the Psychologists has in turn influenced the design of a knowledge representation language having, per se, interesting features in terms of expressiveness and efficiency;
    - finally, the researchers in Machine Learning have been able to test their ideas on theory revision on real data.

Several other works concerning the change in understanding have been conducted in "taskforce 1". They are described in (Kayser & Vosniadou, in preparation).

Now that we have stronger tools to represent the knowledge state of a student, promising perspectives are opened to ask new questions about the evolution of this knowledge state under the influence of teaching, and the answer to these questions has obviously a great importance for Education.

ACKNOWLEDGMENTS

This work has been done in the framework of Project "Learning in Humans and Machines" funded by the European Science Foundation. Project leaders : Hans Spada, Univ. of Freiburg, Germany and Lorenza Saitta, Univ. of Turin, Italy.

REFERENCES

Bobrow, D. G., & Winograd, T. (1977). An Overview of KRL, a Knowledge Representation Language *Cognitive Science, 1,* 3-46.

Brachman, R. J., & Schmolze, J. G. (1985). An Overview of the KL-ONE Representation System *Cognitive Science, 9,* 171-216.

Chaignaud, N. (1996). *Étude cognitive et informatique de la résolution d'un problème : analyse, modélisation et implantation.* Thèse d'Informatique, Université Paris-Nord.

Fahlman, S. E. (1979). *NETL - A System for Representing and Using Real-World Knowledge.* M.I.T. Press.

Gottlob, G. (1996). Complexity and Expressive Power of KR Formalisms. In *Proceedings of KR'96* (pp. 647-649). Cambridge Mass..

Ioannides, C., & Vosniadou, S. (1993). *Mental models of force.* Paper presented at the Fifth European Conference for Research on Learning and Instruction. Aix-En-Provence, France.

Ioannides, C., & Vosniadou, S. (submitted). Aspects of the development of the concept of force. *Child Development.*

Johnson-Laird, P. N. (1983). *Mental Models.* Cambridge University Press, Cambridge.

Kayser, D., & Voșniadou, S. (Eds.) (in preparation). *Change in Understanding in Humans and Machines: Case Studies in Physical Reasoning.*

Lehnert, W. G., & Wilks, Y. (1979). A Critical Perspective on KRL, *Cognitive Science, 3,* 1-28.

Semeraro, G., Esposito, F., Fanizzi, N., and Malerba, D. (1995). Revision of Logical Theories. in Topics in Artificial Intelligence, Lecture Notes in Artificial Intelligence 992, M. Gori and G. Soda (Eds.), Springer, 365-376.

Shastri, L. (1993). A Computational Model of Tractable Reasoning — taking inspiration from cognition. In *Proceedings of the 13th IJCAI* (pp.202-207).

Shastri, L., & Ajjanagadde, V. (1993). From simple associations to systematic reasoning. *Behavioral and Brain Sciences, 16,* 417-494.

Vosniadou, S. (1994). Capturing and modelling the process of conceptual change. *Learning and Instruction, 4,* 45-69.

Vosniadou, S., & Brewer, W. F. (1992). Mental models of the earth: A study of conceptual change in childhood. *Cognitive Psychology, 24,* 535-585.

Vosniadou, S., & Brewer, W. F. (1994). Mental models of the day/night cycle. *Cognitive Science, 18,* 123-183.

Implicit and explicit learning in ACT-R

Christian Lebiere
Department of Psychology
Carnegie Mellon University
Pittsburgh, USA
cl+@andrew.cmu.edu

Dieter Wallach
Department of Psychology
University of the Saarland
Saarbrücken, Germany
dwallach@cops.uni-sb.de

Niels Taatgen
Department of cognitive
science and engineering
University of Groningen
Groningen, Netherlands
n.a.taatgen@bcn.rug.nl

ABSTRACT

A useful way to explain the notions of implicit and explicit learning in ACT-R is to define implicit learning as learning by ACT-R's learning mechanisms, and explicit learning as the results of learning goals. This idea complies with the usual notion of implicit learning as unconscious and always active and explicit learning as intentional and conscious. Two models will be discussed to illustrate this point. First a model of a classical implicit memory task, the SUGARFACTORY scenario by Berry & Broadbent (1984) will be discussed, to show how ACT-R can model implicit learning. The second model is of the so-called Fincham task (Anderson & Fincham, 1994), and exhibits both implicit and explicit learning.

Keywords

ACT-R, implicit learning, explicit learning, skill acquisition, instance theory.

INTRODUCTION TO ACT-R

Knowledge Representation

ACT-R (Anderson, 1993; Anderson & Lebiere, in press) is a hybrid production system architecture for cognitive modeling. It is a hybrid architecture because it works at two interdependent levels: a symbolic level and a subsymbolic level. Each level is divided into a procedural and declarative component.

Symbolic Level

Declarative knowledge consists of chunks. Chunk structures are composed of a number of labeled slots, each of which can hold a value which can also be another chunk. Each chunk is an instance of a particular chunk type, which defines the name and number of slots. Procedural knowledge consists of productions. A production is a condition-action pair, which specifies the action to be taken if a particular condition is satisfied.

ACT-R is a goal-directed architecture. At any time, a goal is selected as the current focus of attention. Goals are organized on the goal stack, on which a goal can be stored (pushed) and later restored (popped). ACT-R operates in discrete cycles. At the start of each cycle, each production is matched against the state of the current goal. The productions that match enter the conflict set. A production is selected from the conflict set. The rest of the production condition can specify a number of chunk retrievals from declarative memory. If the retrievals are not successful, then the next production in the conflict set is selected. If the retrievals are successful, then the production action is executed. The action can modify the current goal, push it on the stack or pop it and restore a previous goal.

Subsymbolic Level

At the symbolic level, ACT-R operates in discrete, deterministic steps, but the subsymbolic level provides a measure of continuity and randomness. The previous section left two points unspecified: how are productions ordered in the conflict set, and if several chunks match a particular declarative retrieval, which is selected?

The productions are selected in order of decreasing expected utility. The current goal is assigned a value, or gain, equal to the worth of successfully achieving it. To each production is associated the probability and cost of achieving the goal to which it applies. The expected utility of a production applied to a goal is equal to the gain of the goal times the probability of success of the production, minus its cost. Noise is also added to the expected utility of a production, making production selection stochastic.

If several chunks satisfy a declarative retrieval, then the most active one is retrieved. The activation of a chunk is the sum of a base-level activation and an associative activation. The associative activation is spread from the sources of activation, which are the components of the current goal, to all related chunks in memory. Noise is added to each activation, making the retrieval of chunks stochastic. If no chunk activation reaches a retrieval threshold, then the retrieval fails. Furthermore, chunks which only partially match the retrieval pattern can also be retrieved, but their activation level will be penalized by an amount proportional to the degree of mismatch between the retrieval pattern and the actual chunk values.

Finally, the time to retrieve a chunk from memory is an exponentially decreasing function of its activation level. Therefore, although ACT-R operates in discrete cycles, the latency of each cycle, which is equal to the sum of the time to perform all the chunk retrievals plus the action time of the successful production, is a continuous quantity. Whereas the specification of an ACT-R model at the symbolic level has a precise, algorithmic quality, its operation at the subsymbolic level matches the stochasticity and continuity of human performance.

Learning

The previous section describes the performance of ACT-R assuming a certain state of knowledge. However, to provide an adequate model of human cognition, it is also necessary to specify how that knowledge was acquired. In ACT-R, knowledge is learned to adapt the system to the structure of the environment (Anderson, 1990; Anderson & Schooler, 1991).

Symbolic Learning

When a goal is popped, it becomes a chunk in declarative memory. That (and the encoding of environmental stimuli) is the only source of declarative knowledge in ACT-R. The chunk resulting from a goal represents the statement of the task addressed by the goal and usually its solution. Therefore, the next time that task arises, its solution, depending upon the activation of the chunk, might be directly retrieved from declarative memory instead of being recomputed anew.

Productions are created from a special type of chunk called dependency. When a goal is solved through a complex process, a dependency goal can be created to understand how it was solved (e.g. which fact was retrieved or which subgoal was set). When that dependency goal is itself popped, a production is automatically compiled to embody the solution process. Thus the next time a similar goal arises, the production might be available to solve it in a single step instead of a complex process.

Symbolic knowledge is learned to represent in a single, discrete structure (chunk or production) the results of a complex process. Subsymbolic knowledge is adjusted according to Bayesian formulas to make more available those structures which prove most useful.

Subsymbolic Learning

When a production is used to solve a goal, its probability and cost parameters are updated to reflect that experience. If the goal was successfully solved, then the production probability is increased. Otherwise, it is decreased. Similarly, the production cost is updated to reflect the actual cost of solving that goal. Declarative parameters are adjusted in the same way. When a chunk is retrieved, its base-level activation is increased. The strength of association between the current sources and the chunk is also increased.

Subsymbolic knowledge does not result in new conscious knowledge, but instead makes the existing symbolic knowledge more available. Chunks which are often used become more active, and thus can be retrieved faster and more reliably. Productions which are more likely to lead to a solution and/or at a lower cost will have a higher expected utility, and thus are more likely to be selected during conflict resolution.

IMPLICIT LEARNING IN THE SUGARFACTORY TASK

Introduction

In contrast to rule-based approaches that conceptualize skill acquisition as learning of abstract rules, theories of instance-based learning argue that the formation of skills can be understood in terms of the storage and deployment of specific episodes or instances (Logan, 1988; 1990). According to this view, abstraction is not an active process that results in the acquisition of generalized rules, but that rule-like behaviour emerges from the way specific instances are encoded, retrieved and deployed in problem solving. While ACT-R has traditionally been associated with a view of learning as the acquisition of abstract production rules (Anderson, 1983; 1993), we present a simple ACT-R model that learns to operate a dynamic system based on the retrieval and deployment of specific instances (i.e. chunks) which encode episodes experienced during system control. It is demonstrated that the ACT-R approach can explain available data as well as an alternative model that is shown to be based on critical assumptions.

The Task

Berry & Broadbent (1984) used the computer-simulated scenario SUGARFACTORY to investigate how subjects learn to operate complex systems. SUGARFACTORY is a dynamic system in which participants are supposed to control the sugar production *sp* by determining the number of workers *w* employed in a ficticious factory. Unbeknown to the participants, the behavior of SUGARFACTORY is governed by the following equation:

$$sp_t = 2 * w_t - sp_{t-1}$$

The number entered for the workers *w* can be varied in 12 discrete steps $1 \le w \le 12$, while the sugar production changes discretely between $1 \le sp \le 12$. To allow for a more realistic interpretation of *w* as the number of workers and *sp* as tons of sugar, these values are multiplied in the actual computer simulation by 100 and 1000, respectively. If the result according to the equation is less than 1000, *sp* is simply set to 1000. Similarly, a result greater than 12000 leads to an output of 12000. Finally, a random component of ± 1000 is added in 2/3 of all trials to the result that follows from the equation stated above. Participants are given the goal to produce a target value of 9000 tons of sugar on each of a number of trials.

The models

Based on Logan's *instance theory* (1988; 1990) Dienes & Fahey (1995) developed a computational model to account for the data they gathered in an experiment using the SUGARFACTORY scenario. According to instance theory, encoding and retrieval are intimately linked through attention: encoding a stimulus is an unavoidable consequence of attention, and retrieving what is known about a stimulus is also an obligatory consequence of attention. Logan's theory postulates that each encounter of a stimulus is encoded, stored and retrieved using a separate memory trace. These separate memory traces accumulate with experience and lead to a „gradual transition from algorithmic processing to memory-based processing“ (Logan, 1988, p. 493). In the following, we contrast the Dienes & Fahey (1995) model (D&S model) with an alternative instance-based ACT-R model and discuss their theoretical and empirical adequacy.

Algorithmic Processing

Both models assume some algorithmic knowledge prior to the availability of instances that could be retrieved to solve a problem. Dienes & Fahey (1995, p. 862) observed that 86% of the first ten input values that subjects enter into SUGARFACTORY can be explained by the following rules:

(1) If the sugar production is below (above) target, then enter a workforce that is different from the previous input by an amount of 0, +100, +200 (0, -100, -200).

(2) For the very first trial, enter a work force of 700, 800 or 900.

(3) If the sugar production is on target, then respond with a workforce that is different from the previous one by an amount of -100, 0, or +100 with equal probability.

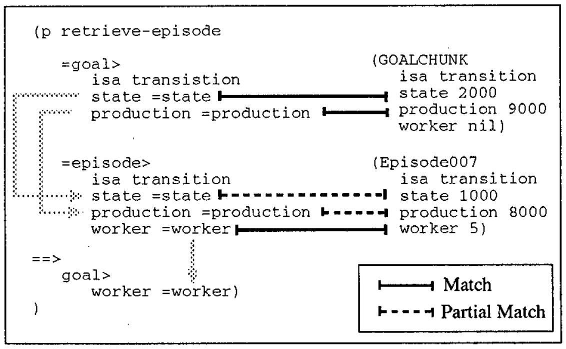
While this algorithmic knowledge is encoded in the D&F model by a constant number of prior instances that could be retrieved in any situation, ACT-R uses simple production rules to represent this rule-like knowledge. The number of prior instances encoded is a free parameter in the D&S model that was fixed to give a good fit to the data reported below. There is no equivalent parameter in the ACT-R model.

Storing Instances

Logan's instance theory predicts that every encounter of a stimulus is stored. The D&F model, however, does *only* store instances for those situations, in which an action successfully leads to the target; all other situations are postulated to be forgotten immediately by the model. Moreover, the D&S model uses a „loose" definition of the target that was unavailable to subjects: While subjects were supposed to produce 9000 tons of sugar as *the* target state in the experiment, a loose scoring scheme was used to determine the performance of the subjects. Because of the random component involved in the SUGARFACTORY, a trial was counted as being on target if it resulted in a sugar production of 9000 tons with a tolerance of ±1000. The D&M model stores only instances that are successful in this loose sense and thus uses information about a range of target states that subjects were not aware of. ACT-R, on the other hand, encodes *every* situation, irrespective of its result. The following chunk is an example for an instance acquired by the ACT-R model as a restored goal.

```
(transition1239
    ISA transition
    STATE 3000
    WORKER 8
    PRODUCTION 12000)
```

The chunk encodes a situation in which an input of 8 workers, given a current production of 3000 tons, led to subsequent sugar production of 12000 tons. While the model developed by Dienes & Fahey (1995) stores multiple copies of instances, ACT-R does not dublicate identical chunks.



```
(p retrieve-episode

   =goal>                                        (GOALCHUNK
      isa transistion                               isa transition
      state =state                                  state 2000
      production =production                        production 9000
                                                    worker nil)

   =episode>                                     (Episode007
      isa transition                                isa transition
      state =state                                  state 1000
      production =production                        production 8000
      worker =worker                                worker 5)
==>
   goal>
      worker =worker)
)
```

Figure 1. Matching process in the Sugar Factory model

Retrieving Instances

In the D&F model each stored instance „relevant" to a current situation races against others and against prior instances representing algorithmic knowledge; the first instance after a finishing post determines the action of the model. An instance encoding a situation is regarded to be „relevant", if it either matches the current situation exactly, or if it is within the loose range discussed above. As with the storage of instances, memory retrieval in the D&F model is again based on specific information not available to subjects. Retrieval in the ACT-R model, on the other hand, is governed by similarity matches between a situation currently present and encodings of others experienced in the past (see Buchner, Funke & Berry, 1995 for a similar position in explaining the performance of subjects operating SUGARFACTORY). On each trial, a memory search is initiated based on the current situation and the target state '9000 tons' as cues in order to retrieve an appropriate intervention or an intervention that belongs to a similar situation. The production rule `retrieve-episode` (figure 1) is used to model the memory retrieval of chunks based on their activation level. Instances which only partially match the retrieval pattern, i.e. which do not correspond exactly to the present situation, will be penalized by lowering their activation proportional to the degree of mismatch. As a parameter of the ACT-R model, normally distributed activation noise is introduced to allow for some stochasticity in memory retrieval.

As figure 2 shows, the use of instances over the initial algorithmic knowledge increases over time, resulting in the gradual transition from algorithmic to memory-based processing as postulated by Logan (1988, p. 493).

Theoretical Evaluation

While both models of instance-based learning share some striking similarities, the theoretical comparison has shown that the D&F-model makes stronger assumptions with respect to the storage and the retrieval of instances that can hardly be justified. Dienes & Fahey (1995) found out that these critical assumptions are essential to the performance of the D&F model(p. 856f):

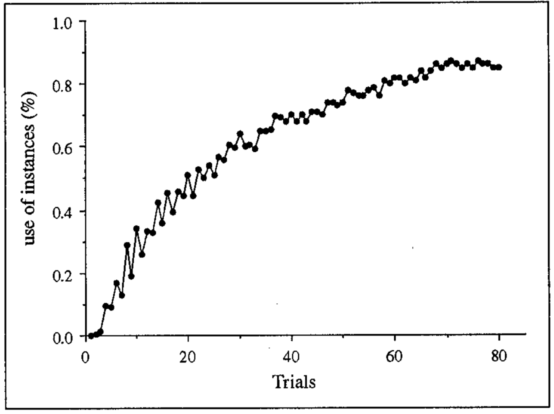


Figure 2. Relative use of instance retrieval per trial

> „The importance to the modeling of assuming that only correct situations were stored was tested by determining the performance of the model when it stored all instances. ... This model could not perform the task as well as participants: The irrelevant workforce situations provided too much noise by proscribing responses that were in fact appropriate ... If instances entered the race only if they exactly matched the current situation, then for the same level of learning as participants, concordances were significantly greater than those of participants".

Since the ACT-R model does not need to postulate those critical assumptions, this model can be regarded as the more parsimonious one, demonstrating how instance-based learning can be captured by the mechanisms provided by a unified theory of cognition.

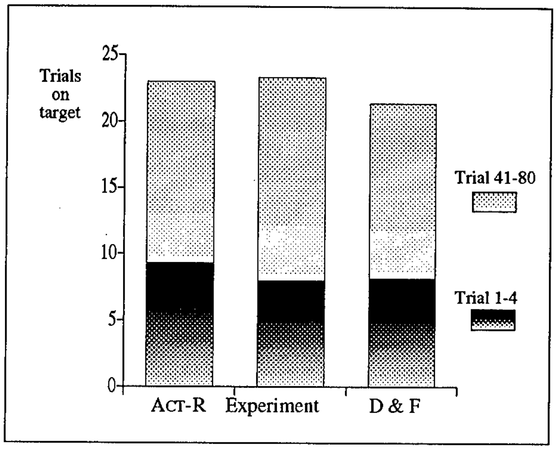


Figure 3. Results of the experiment, ACT-R model and D&F model

Empirical Evaluation

While the theoretical analysis of the assumptions underlying the two models has favoured the ACT-R approach, we will briefly discuss the empirical success of the models with respect to empirical data as reported by Dienes & Fahey (1995). Figure 3 shows the trials on target when controlling SUGARFACTORY over two phases, consisting of 40 trials each. ACT-R slightly overpredicts the performance found in the first phase, while the D&F model slightly underpredicts the performance of the subjects in the second phase. Since both models seem to explain the data equally well, we cannot favour one over the other.

Figure 4 shows the performance of the models in predicting the percentage of times („Concordance") that the subjects gave the same (correct or wrong) response in a questionaire as they did when controlling SUGARFACTORY. Again, both models seem to do a similar good job in explaining the data, with no model being clearly superior. Although space limitations do not allow for a detailed discussion, the picture illustrated by these two empirical comparisons remains the same after several additional model comparision tests. We are currently running an experiment, exploring different predictions of the models in more details.

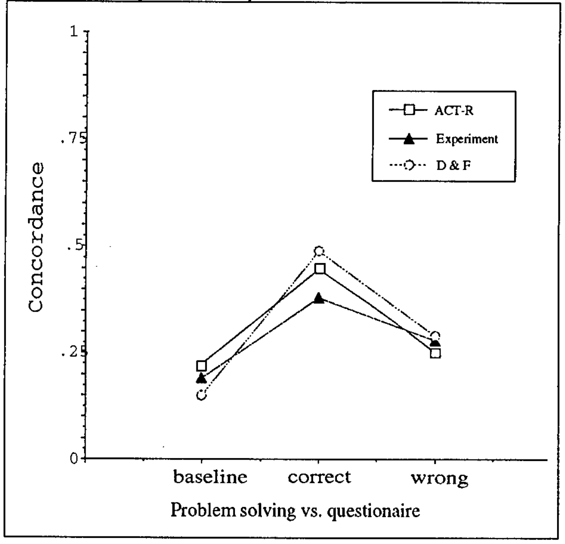


Figure 4. Concordances for the experiment and both models

Conclusion

We discussed and compared a simple ACT-R model to an approach based on Logan's instance theory with respect to their ability to modeling the control of a dynamic system. While both models were similar in their empirical predictions, the ACT-R model was found to require fewer assumptions and is thus preferred over the model proposed by Dienes & Fahey (1995). Generally, ACT-R's integration of an activation-based retrieval process with a partial matcher seems to be a very promising starting point for the development of an ACT-R theory of instance-based learning and problem solving.

IMPLICIT AND EXPLICIT LEARNING IN THE FINCHAM TASK

The learning mechanisms in ACT-R are all quite basic, and can be used in several different ways to achieve different results. The idea of a learning mechanism as an integral part of an architecture has properties in common with the psychological notion of implicit learning. Both types of learning are considered to be always at work and not susceptible to change due to development or great variation due to individual differences. One of the defining properties of implicit learning, the fact that it is not a conscious process, is harder to operationalize within the context of an architecture for cognition. The closest you can get in an architecture is the notion that implicit learning is not guided by learning intentions, but is rather a by-product of normal processing. The SUGARFACTORY model discussed in the previous section is an example of implicit learning, since ACT-R uses old goals that are stored unintentionally to improve its behavior.

Explicit learning, on the other hand, is tied to intentions, or goals in ACR-R terms. Since there are no learning mechanisms that operate on goals, explicit learning can best be explained by a set of learned learning strategies. An example of a learning strategy to improve memorization of facts is using rehearsal to improve base-level learning. Base-level learning increases the activation of a chunk each time it is retrieved. If this increase of activation through natural use is not enough for the current goals, rehearsal can be used to speed up the process. By repeating a fact a number of times, its base-level activation can be boosted intentionally.

In this section we will discuss a paradigm for skill learning that involves both an implicit and an explicit strategy. The implicit strategy corresponds to instance-based learning, and the explicit strategy to rule-learning. Figure 5 shows an overview of this paradigm. First we assume that a participant has some initial method or algorithm to solve the problem. Generally this method will be time-consuming or inaccurate. Each time an example of the problem is solved by this method, an instance is learned. In ACT-R terms an instance is just a goal that is popped from the goal stack and is stored in declarative memory. Since this by-product of performance is unintentional, it can be considered as implicit learning.

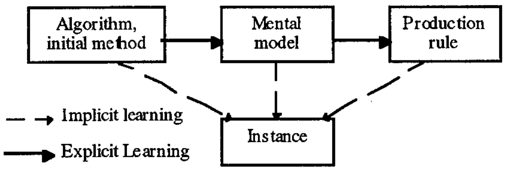


Figure 5. Diagram that illustrates the learning scheme used in the Fincham-task model

Other types of learning require a more active attitude from the participant. If the initial method is too time consuming, the participant may try to derive an re-representation of the information needed for the task to increase efficiency, which we will call, using Johnson-Laird's (1983) terminology, a mental model. If the initial method leads to a large number of errors, the participant may try to deduce or guess new relationships in the task in order to increase performance. The next step, from mental model to production rule, can only be made if the mental model is simple enough to convert to a production rule. Both the application of mental models and firing new production rules will create new instances. So regardless of what is going on due to explicit learning, implicit learning keeps accumulating knowledge.

So, if we have that many ways of learning, what type of learning will we witness in a particular experiment? To be able to answer this question we go back to the principle of rational analysis. According to this principle, we will principally witness that type of learning that will lead to the largest increase in performance. If we have task in which it is very hard to discover relationships or mental models, learning can probably be characterized primarily by implicit instance learning. Tasks in which there are too many instances too learn, but in which relationships are more obvious, will probably be better explainable by rule and abstraction learning. The SUGARFACTORY task is an example in which it is very hard to discover the rules that govern the system due to the influence of the previous sugar production and random factor in the output.

The Fincham Task

An example of a task in which both rule learning and instance learning are viable strategies is described by Anderson & Fincham (1994). In this task, participants first have to memorize a number of facts. These facts are in the form of

"Hockey was played on Saturday at 3 and then on Monday at 1."

We will refer to these facts as "sport-facts" to prevent confusion with facts and rules in the model. A sport-fact contains a unique sport and two events, each of which consists of a day of the week and a time. After having memorized these facts, participants were told the facts are really rules about the time relationships between the two events. So in this case "Hockey" means you have to add two to the day, and subtract two from the time. In the subsequent experiment, participants were asked to predict the second event, given a sport and a first event, or predict the first event, given the sport and the second event. So participants had to answer questions like: "If the first game of hockey was Wednesday at 8, when was the second game?" In this paradigm, it is possible to investigate evidence for both rule-based learning and instance-based learning. Directional asymmetry, evidence for rule-based learning, can be tested for by first training a sport-fact in one direction (by predicting the second event using the sport and the first event), and then reverse the direction (by predicting the first event using the sport and the second event) and look how performance in the reverse direction relates to performance on the trained direction. If the performance is worse in the reverse direction, this is evidence for the use of rules. Evidence for instance learning can be gained by presenting specific examples more often than other examples. Better performance on these specific examples would indicate instance learning. Anderson & Fincham (1994), and later Anderson, Fincham & Douglass (1997) performed five variations on this basic experiment. The basic findings we will focus on are as follows:

- In general, reactions times improve according to the power law of practice, starting at around 35 seconds for the first few trials and improving to around 7 seconds at the third session.
- There is evidence for rule learning as witnessed by directional asymmetry. However, the effect only starts at the third or fourth session, and is relatively small.
- There is evidence for instance learning, since problems that are repeated more often than others are solved faster.
- Although it can not be inferred directly from the data, participants report they use abstract versions of the rules, for example by memorizing "Hockey day +2" and "Hockey time -2".

On basis of this evidence, Anderson et al. conclude that participants use four strategies: analogy, abstraction, rule and instance. The interesting question is what learning processes play a role in changing strategies. Each of the four strategies can be related to one of the learning stages from figure 5.

The analogy strategy is the initial strategy: first the memorized example that has the same sport as the new trial is recalled, the relationship in this example is determined, and this relationship is mapped on the current trial. Analogy is not very efficient, since it consists of many steps.

The abstraction strategy assumes the participant has created and memorized a mental model of the sport that corresponds to the current trial, like "Hockey day +2". The strategy involves retrieving and applying the abstraction, which is easier and faster than the analogy strategy.

The rule strategy assumes a production rule has been learned that can fill in the answer directly. An example of this rule is (variables are indicated by italics):

```
IF   the goal is to find the day of the second event
     the sport is hockey
     and the day of the first event is day1
     AND day1 plus two days equals day2
THEN put day2 in the second event slot of the goal
```

The rule strategy is more efficient than the abstraction strategy, since it requires only a single step in stead of two.

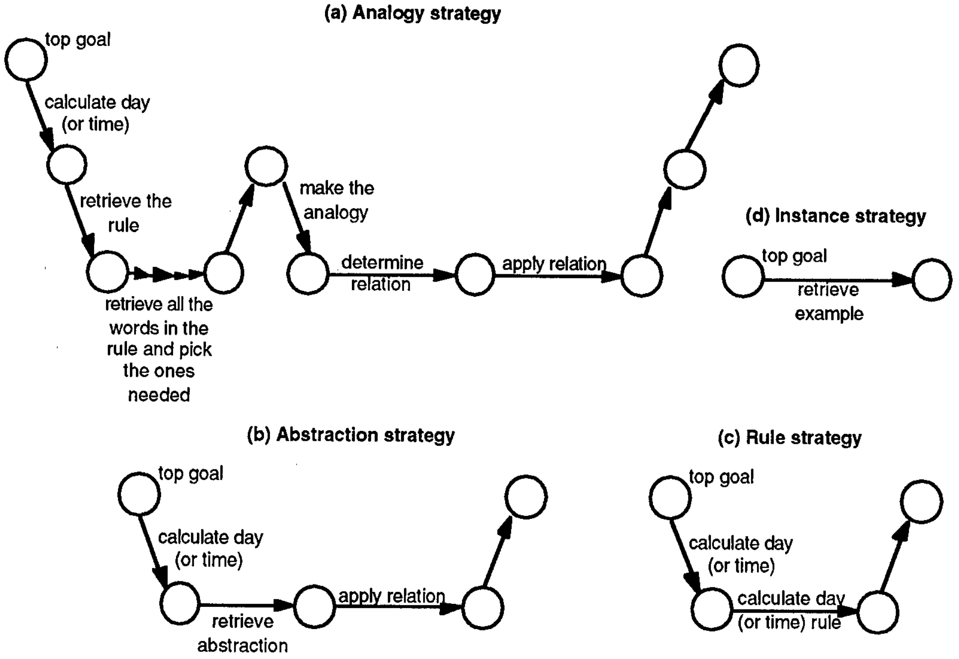


Figure 6. Overview of the four strategies in the Fincham task as modeled in ACT-R

The instance strategy assumes the answer can be given using a previous example. This previous example must be the same as the current trial. So an instance may contain the following information:

item1434
    isa instance
    sport hockey
    type day
    left sunday
    right tuesday

To use the instance strategy, it is sufficient to retrieve the right instance. This will of course only succeed if this instance is present in memory and is retrievable.

An ACT-R Model

We will now briefly discuss the ACT-R model of the task and its results. A more extensive discussion can be found in Taatgen & Wallach (in preparation). Figure 6 shows a schematic diagram of the implementation of the four strategies.

The analogy, abstraction and rule strategies are performed in a subgoal, that focuses on calculating either the day or the time. The instance strategy attempts to retrieve one of these subgoals, and fill in the answer directly in the topgoal. So learning instances is an implicit process in ACT-R, since past goals are always stored in declarative memory, an reoccurrence of the same goal just increases the activation of that goal. Knowledge for the other two strategies has to be acquired in an explicit fashion. An abstract mental model of a sport is no automatic by-product of the analogy strategy, so an explicit decision must be made to memorize an abstraction. To learn a new production rule in ACT-R, a special dependency structure must be created in declarative memory, which is also an explicit decision. In the current model, learning a new production rule is only successful if there is already an abstraction present in declarative memory, else it is too difficult to collect the necessary information.

Results of the Model

In this paper we will only discuss results of the model on the second experiment of Anderson & Fincham (1994). In this experiment, participants had to learn eight sport-facts. In the first three days of the experiment, four of these sport-facts were tested in a single direction: two from left to right and two from right to left. On each day 40 blocks of trials were presented, in which each of the four sport-facts was tested once. On the fourth day all eight sport-facts were tested in both directions. On this day 10 blocks of trials were presented, in which each of the eight sport-facts was tested twice, once for each direction. Figure 7 shows the latencies in the first three days of the experiment, both the data from the experiment and from the model. The fit between the model and the data is quite good (R^2=0.94).

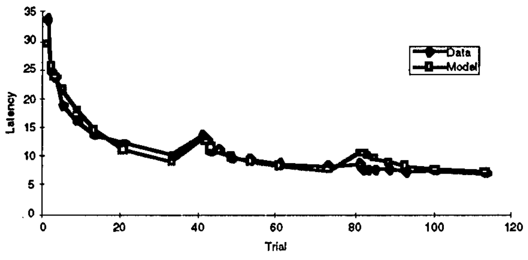


Figure 7. Latencies in experiment 1 for days 1-3

The results on day 4 can be summarized in the following table:

	Data	Model
Same direction, practiced	8.9 sec	8.4 sec
Reverse direction, practiced	10.9 sec	9.3 sec
Not practiced	13 sec	16 sec

Both in the data and in the model there is a clear directional asymmetry, since items in the practiced direction are solved faster than reversed items. The fact that unpracticed items are slower than the reversed items indicates that rule learning can not be a sufficient explanation for all of the learning in the first three days of the experiment.

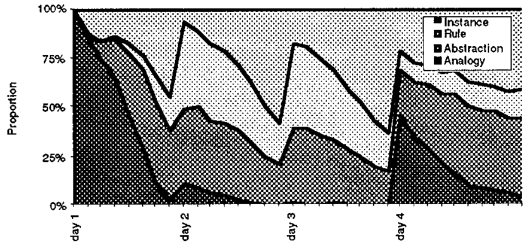


Figure 8. Strategy use in experiment 1 for days 1-4

Figure 8 shows how the model uses the four strategies in the course of the experiment. At the start of the experiment, analogy is used most of the time, but both the abstraction and the instance strategy gain in importance after a few blocks of trials. The rule strategy only appears later, and only plays a minor role during the first day. At the start of the second day, there is a large shift toward using rules at the expense of instances. This can be explained by the fact that the activation of a large portion of the instances has decayed between the two days, so that they can not be retrieved anymore. Since only few rules are needed for successful performance, they receive more training on average and are less susceptible to decay. Note that the abstraction strategy remains relatively stable between the days since it also less susceptible to decay than the instance strategy. This pattern is repeated at the start of the third day, although the instance strategy looses less ground due to more extended training of the examples. At the start of the fourth day, the frequency of use of the analogy strategy goes up again, since there are no production rules for the new four sport-facts. The abstraction strategy can take care of the reversed items though, so in that case the expensive analogy strategy is not needed. This explains the fact that reversed items are still faster than completely new items.

Except for a model of this experiment, the model has successfully modeled two other experiments as well, using the same parameters. The following additional phenomena could successfully be explained:

- The reaction time for examples that are repeated more often is shorten, since instance learning is more successful and the facts it represents have a higher activation.
- Directional asymmetry increases between day 2 to 4, but decreases again on day 5. The model can explain this by the fact that by day 5 the instance strategy starts dominating the rule strategy.
- The results of the model concur with participant's reports on whether they use a rule or an example to solve a particular trial.

Conclusions

The ACT-R architecture is an ideal platform to study implicit and explicit learning. It not only allows insights in both types of learning separately, but, more importantly, also in the interaction between them.

REFERENCES

Anderson, J. R. (1990). *The adaptive character of thought.* Hillsdale, NJ: Erlbaum.

Anderson, J. R. (1993). *Rules of the mind.* Hillsdale, NJ: Erlbaum.

Anderson, J.R. & Fincham, J.M. (1994). Acquisition of Procedural Skills From Examples. *Journal of experimental psychology: Learning, Memory, and Cognition*, vol. 20, no. 6, 1322-1340.

Anderson, J.R. , Fincham, J.M. & Douglas, S. (1997). The role of Examples and Rules in the Acquisition of a Cognitive Skill. Journal of experimental psychology: Learning, Memory, and Cognition, vol. 23, no. 4, 932-945.

Anderson, J. R. & Lebiere, C.. (in press). *The atomic components of thought.* Mahwah, NJ: Erlbaum.

Anderson, J. R., & Schooler, L. J. (1991). Reflections of the environment in memory. *Psychological Science, 2,* 396-408.

Berry, D. & Broadbent, D.A. (1984). On the relationship between task performance and associated verbalizable knowledge. *The Quarterly Journal of Experimental Psychology*, 36A, 209-231.

Buchner, A., Funke, J. & Berry, D. C. (1995). Negative correlations between control performance and verbalizable knowledge: Indicators for implicit learning in process control tasks? *The Quarterly Journal of Experimental Psychology*, 48A, 166-187.

Dienes, Z. & Fahey, R. (1995). Role of specific instances in controlling a dynamic system. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 21 (4), 848-862.

Johnson-Laird, P.N. (1983). *Mental Models: Towards a Cognitive Science of Language, Inference, and Consciousness.* Cambridge, MA: Harvard University Press.

Logan, G.D. (1988). Toward an instance theory of automatization. *Psychological Review*, 95, 492-528.

Logan, G.D. (1990). Repetition priming and automaticity: Common underlying mechanisms? *Cognitive Psychology*, 22, 1-35.

Taatgen, N.A. & Wallach, D. (in preparation). Models of rule and instance-based skill learning.

Poster Abstracts

Separation of logical and calculation capabilities in a problem solving task

Jean-Bernard Auriol - Jean-Louis Dessalles
Département Informatique - ENST
46, rue Barrault, 75013, Paris - France
auriol@inf.enst.fr - dessalles@enst.fr

ABSTRACT

We present herein a model based on a strict separation between logical and calculation capabilities, designed to mimic aspects of human problem solving behaviour. Our model has been designed to be simple and psychologically plausible. We have tested our approach on the Tower of Hanoi task by comparing the results provided by our model with the performance of novice subjects. We also compared these results with the performance of a few other computational models. These comparisons are quite promising.

Keywords

problem solving, logical knowledge, procedural knowledge, calculation.

INTRODUCTION

In (Johnson-Laird and Byrne, 1991), convincing evidence is presented that seems to undermine the existence of human logical capabilities. Mental models (Johnson-Laird, 83) would explain experimental results on logical problem solving tasks much better than logical models do. Evidence from the observation of natural conversations (Dessalles, 1993) suggest however that the ability we have to argue with each other in everyday verbal interactions relies on genuine logical capabilities. Our hypothesis is that the same logical capabilities are involved in problem solving. We propose that the problem solving behaviour of subjects can be partly explained by the joint operation of two separate sets of capabilities: logical and calculation capabilities.

THE MODEL

Calculation Capabilities

Calculation Knowledge Representation: Operators

The representation of calculation knowledge is based on operators. An operator takes the following form:

(State 1, Operation, State 2)

where **State 2** results from the application of **Operation** to **State 1**. Operators are able to propose in sequence all existing legal steps from a given situation. An operator can be applied recursively, up to a given search depth, by taking one of the resulting states it has proposed as a new starting state.

Preference

We postulate a contextual preference for operators; in a given context, the operator will propose legal steps in a given order.

Reversibility

Operators are reversible in two ways. Given a resulting state, an operator can propose legal steps leading to this state and the associated starting state. Given a step, an operator can propose a starting state in which this step would be legal.

Logical Capabilities

The role of the logical part of the model is to evaluate situations and design goals. Its specific form is motivated by independent studies, particularly conversation modelling (Dessalles, 1993).

Logical Knowledge Representation

Logical knowledge is represented by first-order logical rules, in an extension of the negative conjunctive normal-form:

List of terms ⇒ Mod

Each term in the list is in conjunction with the rest of the list, and **Modality** (noted '**Mod**') is either **Undesirable** or **False**. Facts are stored in memory with no specific order, in the following basic form:

(Fact, Truth Value)

where **Truth Value** can be either '**true**' or '**false**'. Facts with an unknown truth value are not stored in memory.

Saturation Detection

The first capability that we put forward for the logical part of our model is the systematic detection of rule saturation. A rule is said to be saturated when all the terms of the rule are known to have the truth-value with which they appear in the rule. Depending on the modality, an undesirable or paradoxical situation will be detected in this case. We call such a situation a **problematic situation.**

Counter-Factual Production

To get out of a problematic situation, the subject has to change the truth-value of one term of the saturated rule. This is done by producing a counterfactual. A counterfactual is a term with a truth-value that is known to be false but that cancels the problematic aspect of the current situation. This counter-factual generation can be done repeatedly until the situation is no longer problematic.

Coupling logical and calculation capabilities

Problem representation

The problem representation is split into two parts. In the logical part, the situation is represented by facts. In the calculation part, the situation is represented by states. Goals are represented by undesirability rules in the logical part, and are not represented in the calculation part.

Goal-Oriented, Preference-Oriented Exploration

The strategy used to solve the problem is to explore the search space until reaching a state where the current undesirability is no longer saturated. It can be written in the following form:

```
OPERATORS: EXPLORE PROBLEM SPACE WITHIN SEARCH
        DEPTH
IF CURRENT UNDESIRABILITY SATURATED
    CONTINUE EXPLORATION
ELSE
    PLAY PROPOSED MOVE(S)
    IF NEW UNDESIRABILITY NEW_UND
        CURRENT UNDESIRABILITY = NEW_UND
    ELSE
        STOP
```

With a restricted search depth, the set of reachable states is limited. It often happens that all of them are uninteresting. In this case, the preferred move of the operator will be played, and the search process will start again from the new state reached. After a few steps made along according to mere preference, and if no interesting state is reached, the search stops: this is a dead end.

Getting Out of Dead End: 'Counter-factual' and Operator Reversibility

A dead-end situation is characterised by the fact that the current undesirability is out of reach of the operator. The strategy used to get out of dead ends can be sketched this way:

```
SELECT A TERM OF THE CURRENT SATURATED RULE
INVERT THE TRUTH VALUE OF THIS TERM
IF A NEW RULE BECOMES SATURATED
    REPEAT THE PROCESS
ELSE
    CALL OPERATOR WITH
        CURRENT STATE AS STARTING STATE
        DESIRED STATE AS ENDING STATE
TURN SITUATION RETURNED BY OPERATOR
    INTO UNDESIRABILITY RULE
RE-START SEARCH PROCESS
```

EXPERIMENTS

Our experiment is based on the comparison between solutions given by our model and by human subjects. We performed a step by step comparison between both solutions. In order to be able to compare the solutions after the first difference in move, our solution is bound to follow the subject's solution. At each step, our model computes its next move, which we compare to the human move. The human step is always the one played. Differences are counted, and whenever the erroneous move was chosen due to operator preferences, the involved preference is inverted.

We tested the system on 40 protocols, produced by seven novice subjects, and totalling 1462 steps. We also tried different others models. Besides random strategies (pure random, random without moving the same disk twice, preferences replaced by random[1]), we experimented with a model inspired by (VanLehn, 1991).

In this latter model, three steps out of four are forced steps. Each time the model moves Disk 1, the next allowed move is to take the only other legally moveable disk and to put it on the only legal peg. Each time the model moves Disk 2, the next allowed step is to put Disk 1 back on it. The model chooses the optimal move for each unresolved move. Without correction, this algorithm always gives the optimal solution.

RESULTS AND DISCUSSION

For each model, we computed the percentage of correctly predicted moves out of the total number of moves. The results obtained after these trials are:

Random:	33.68%
Random without backtrack:	68.88%
Our model without preferences:	73.76%
Inspired by VanLehn:	78.66%
Our model:	80.71%

The results given by models involving random may vary by 1%. Results given by our model also vary by 0.7% around the value we give, because initial preferences are fixed at random. The results of the VanLehn inspired model do not vary.

The differences between the three first models and ours are significant (chi^2 = 15.49, $p < 0.0005$, for the comparison between our model and the random and logic model). The difference between our model and the VanLehn inspired model is not significant (chi^2 = 1.51, $p < 0.25$). The VanLehn inspired model gives good result principaly because it takes avantage of task specific constraints. Yet, the VanLehn inspired model generates by itself only the optimum solution, and cannot be, as such, a good model of human behaviour.

The comparison with the three random models is interesting. Our model without preferences is much better that random alone and is significantly better than random without backtrack. This confers an independent validation to the logical part of our model. Also, the results given by the complete model are better than those obtained by replacing preference by random, which indicates that, on the calculation side, preferences better account for human behaviour than random choices do.

ACKNOWLEDGEMENTS

We thank Dr. Bernard Auriol for his help.

REFERENCES

Dessalles, J-L. (1993). *Modèle cognitif de la communication spontanée, appliqué à l'apprentissage des concepts - Thèse de doctorat.* Paris: ENST - 93E022.

Johnson-Laird, P. N. (1983). *Mental Models.* London: Cambridge University Press.

Johnson-Laird, P. N. & Byrne, R. M. J. (1991). *Deduction.* Lawrence Erlbaum Associates.

VanLehn, K. (1991). "Rule acquisition events in the discovery of problem-solving strategies". *Cognitive Science, 15.*

[1] That is, our model where the operator preferences were replaced by random choice.

Simulating chess players' recall: How many chunks and what kind they can be?

Heikki Hyötyniemi
Control Engineering Laboratory
Helsinki University of Technology
Otakaari 5 A, FIN-02150 Espoo, Finland
+358 9 451 3327

Pertti Saariluoma
Cognitive Science
P.O. Box 13 (Meritullinkatu 1 B)
FIN-00014 University of Helsinki, Finland
+358 9 19123458

ABSTRACT

This paper presents a numeric rather than symbolic approach to the chunking problem. The application area is the expert recall of chess board configurations. It is shown that a relatively low number of 'skilled' chunks is enough to explain the chess players recall of chess positions.

Keywords
Chunking, mental images; neural networks

INTRODUCTION

Chess players' recall of chess positions has been one of the major experimental paradigms in basic cognitive skills research (Chase and Simon 1973, Djakov, Petrovski and Rudik, 1926, de Groot 1965, 1966). In this research it was shown that expert chess players are superior to novices in recalling real game positions but not essentially better in recalling their randomized versions. The finding has been generalized over a large number of cognitive skills and it has proven to be very stable.

Perhaps the only issue of real concern has been the number of chunks experts have to learn to achieve their skill. Simon and Gilmartin (1973) argued that they must have learned, at least, 50,000 to 100,000 chunks. The evidence was based on simulation. However, Holding (1985) noticed that in these models the locations of the pieces were absolutely coded. Consequently, it was possible to assume that much lesser a number of chunks could explain the performance of the subjects. Saariluoma (1994) and Gobet and Simon (1996) have met the criticism by showing that chess players recall is impaired by transposition of the chunks on a chess board, which is critical to Holding's (1985) argumentation.

Another, theoretical presupposition in the original Simon and Gilmartin (1973) argumentation is the reliance on symbolic modeling. It might be possible that the whole philosophy of symbolic modeling is not adequate approach to the problems of human memory. As is well known various types of neural networks have challenged very deeply the idea of symbolic modeling. The evidence is today vast and it should be discussed in the context of chess players' memory recall as well.

In this paper one specific type of neural network model is used to simulate chess players' recall. The outcome of simulation shows that if neural networks are used the number of chunks could be reduced substantially. Thinking of the large support neural network models have in modeling human memory processes, the neural simulation makes it necessary to rethink the explanatory validity of Simon and Gilmartin (1973) argumentation and all models of the same type.

In this experiment, the framework differs very much from the traditional symbolic setting. For example, the chunks are now numeric and real-valued; and rather than expanding, they become more and more specialized as the training goes on. This view of chunks is in contrast with the original chunk idea.

ADAPTATION ALGORITHM

There are various neural network algorithms for pattern classification and feature extraction tasks available (see Bishop, 1995). The following approach[1] is specially tailored for self-organizing search of correlation structures. In statistical terms, it is a special combination of *cluster analysis* and *principal component analysis*; the resulting set of features can also be interpreted as sparsely coded, non-orthogonal *factors*.

The memory structure is a derivation of the Kohonen self-organizing map (Kohonen, 1984). There are *N nodes*, each of which is characterized by a *prototype vector* θ_i, where $1 \le i \le N$. The dimension of the vectors is *n*. The prototype vectors should represent the observed input vectors as accurately as possible – to reach this goal, the standard self-organization algorithm has been modified: rather than constructing only a set of *N* cluster centers characterized by the prototype vectors, the prototype vectors are interpreted now as 'coordinate axes' in the input data space, spanning a rather low-dimensional subspace. The algorithm can be implemented as follows.

1. Take the next input vector sample f.
2. Select the node with the best correlation with the input vector *f*, that is, determine the 'winner' index *c* such that the absolute value $|\varphi_c|$, where $\varphi_c = \theta_i^T f$, reaches its maximum value.
3. Calculate the 'neighborhood' parameter $h_{c,i}$ between the network nodes *i* and the winning node *c*. This parameter has value near 1 if the nodes are 'near' each other in the net, and lower value otherwise, as presented in (Kohonen, 1984).

[1] The analysis and other applications of the algorithm are presented in Hyötyniemi (1997) and (1998).

4. Apply the Kohonen type adaptation (Kohonen, 1984) of the network using the vector $\phi_c \cdot f$ as input. That means, for each network node i update the vector θ_i as $\theta_i \leftarrow \theta_i + \gamma h_{c,i} \cdot (\phi_c f - \theta_i)$. The parameter γ is a decaying function of time to assure that the network finally converges.

5. Normalize the feature vectors: $\theta_i \leftarrow \theta_i / \sqrt{\theta_i^T \theta_i}$ for all $1 \le i \le N$.

6. Eliminate the contribution of the feature number c by setting $f \leftarrow f - \phi_c \cdot \theta_c$.

7. If m features have not yet been extracted, go back to Step 2, otherwise, go to Step 1.

After the network has converged, the prototype vectors represent *features* that can be used to construct the input patterns. That means, given an input vector f, find the sequence of φ_i values as presented in Steps 2 – 7 above (ignoring the updating steps 3 – 5), so that the estimate for f can be constructed as a weighted sum of the features

$$\hat{f} = \theta_1 \varphi_1(f) + \cdots + \theta_N \varphi_N(f).$$

In this context, it is assumed that the extracted features are the *chunks*, conveying the dependency relations between the input elements. The number N stands for the capacity of the long-term memory, while the parameter m is the size of the short-term memory. It is also assumed that at any instant only the references to the static memory structures and the respective weights are operated on.

SIMULATION EXPERIMENTS

To apply the presented algorithm, the input data is first coded appropriately. This means that one must present the chess piece configuration as a vector of real numbers. The coding is now location-based and rather trivial.

It is assumed that the lower-level processing has produced the *component level* constructs, that means, the visual image has been analyzed and atomic information about the board has been extracted – these *information atoms* are now something like 'white king in g1', etc. For simplicity and for generality, it is assumed that each of these information atoms spans a dimension of its own in the input data space – this means that the input vector is 768 dimensional (six pieces of two colors, together 12 alternatives, for each of the 64 board locations). Naturally, this coding is far from optimal - the complexity of different modalities is changed to the high-dimensionality of the input vector space.

In the experiments, 5000 samples were iteratively used for training the network model. These samples were successive piece configurations during real chess games, given in random order. The simulation was implemented in a Matlab environment. The huge size of the data structures made the simulations rather capacity-demanding.

Three chunk models were extracted: the first with only 9, the second with 25, and the third with 100 chunk prototypes available, so that $N = 9$, $N = 25$, and $N = 100$, respectively. Five chunks were used to reconstruct the observed configuration, that means, $m = 5$. There were 500 additional game positions for testing purposes. To visualize the high-dimensional vectors representing the board and the chunks, the numerical values of the vector elements were thresholded – that means, if the value of the element exceeded 0.5, it was assumed that the corresponding piece was there; otherwise its contribution was ignored. No rules of chess were incorporated – in principle, it is possible that, say, two white kings will be displayed simultaneously, but because of the 'skilled' chunk prototypes, this seldom happens[2].

CONCLUSIONS

In the presented approach, the chunks are not 'crisp' - rather, their constituents have continuous (or fuzzy) values. This is one reason why *scalability* seems to apply, so that allocating more resources results in better reconstruction of the piece locations. For the 868 chunks, the average recall rate was about 75%.

Because of the numerical nature of the chunks, they are flexible and they can be added together in a natural way. Due to the possibility of combining chunk prototypes, a rather low number of 'skilled' chunks seems to be enough to reach relatively high level of accuracy.

REFERENCES

Bishop, C.M. (1995). *Neural Networks for Pattern Recognition.* Clarendon Press, Oxford.

Chase, W. G. and Simon H. A. (1973). The mind's eye in chess. In W. Chase (ed.), *Visual information processing.* Academic Press, New York.

de Groot, A. D. (1965). *Thought and Choice in Chess.* Mouton, The Hague.

de Groot, A. D. (1966). Perception and memory versus thought: Some old ideas and recent findings. In B. Kleinmuntz (ed.), *Problem Solving.* New York.

Djakov, I. N, Petrovsky, N. B., and Rudik, P. A. (1926). *Psihologia Shahmatnoi Igry (Chess psychology).* Avtorov, Moskow.

Gobet, F. and Simon, H.A. (1996): Templates in Chess Memory: A Mechanism for Recalling Several Boards. *Cognitive Psychology,* 31, pp. 1 – 40.

Holding, D. H. (1985). *The Psychology of Chess Skill.* Erlbaum, Hillsdale, N.J.

Hyötyniemi, H. (1997). On the Statistical Nature of Complex Data. In *SCAI'97 – Sixth Scandinavian Conference on Artificial Intelligence: Research Announcements* (ed. Grahne, G.), Helsinki University, Department of Computer Science, Report C-1997-49, Helsinki, Finland, pp. 13 – 27.
Available from http://Saato014.hut.fi/Hyotyniemi/publications/.

Hyötyniemi, H. (1998). Automatic Structuring of Unknown Dynamic Systems. In *Soft Computing in Engineering Design and Manufacturing* (eds. Chawdhry, P.K., Roy, R., and Pant, R.K.), Springer-Verlag, London, pp. 410 – 419. Available from http://Saato014.hut.fi/Hyotyniemi/publications/.

Kohonen, T. (1984). *Self-Organization and Associative Memory.* Springer-Verlag, Berlin.

Saariluoma, P. and Hohlfeld, M. (1994). Chess players' long range planning. *European Journal of Cognitive Psychology,* 6, pp. 1–12.

Saariluoma, P. (1994). Location Coding in Chess. *The Quarterly Journal of Experimental Psychology,* 47A (3), pp. 607–630.

Simon, H. A. and Gilmartin, K. (1973). A simulation of memory for chess positions. *Cognitive Psychology,* 5, pp. 29–46.

[2] A simulation environment, implemented in Java, can be found at http://Saato014.hut.fi/Hyotyniemi/publications/97_scai.htm

Is Mental Imagery Symbolic? Exploratory Simulations in an Interactive Activation Model

Rita Kovordányi
Dept. of Computer and Information Science
Linköpings Universitet
S-581 83 Linköping, Sweden
+46 13 28 14 30
ritko@ida.liu.se

ABSTRACT

In this article we present an interactive activation simulation framework for mental image reinterpretation. By varying central parameters in this framework, two qualitatively different models have been emulated: One in which reinterpretation is obtained via a series of symbolic inference steps, and one in which reinterpretetation is driven by parallel operations on a depictive mental image. The simulations are run with the following objectives: 1. To minimally verify that the models can produce reinterpretations. 2. To verify that the parametric relationships predicted by the models hold in the face of empirical constraints on the simulation outcome. 3. To expose unforeseen parametric constraints which are entailed by the two models.

INTRODUCTION

When we close our eyes and mentally image a capital 'X', superimpose a capital 'H' on it, and recognize a "bow tie" in the resulting image, we generate, manipulate, and reinterpret mental images. Psychological experiments on human performance reveal interesting anomalies in how easily mental images are reinterpreted.

Are mental images *reinterpretable* because they supplement symbolic structures with new affordances? These and related matters lie at the heart of 'the imagery debate' (e.g., Kosslyn 1994; Pylyshyn, 1981). This article investigates the role of visual versus symbolic representations as a mediating factor in mental reinterpretation tasks.

Two models of mental image reinterpretation have been

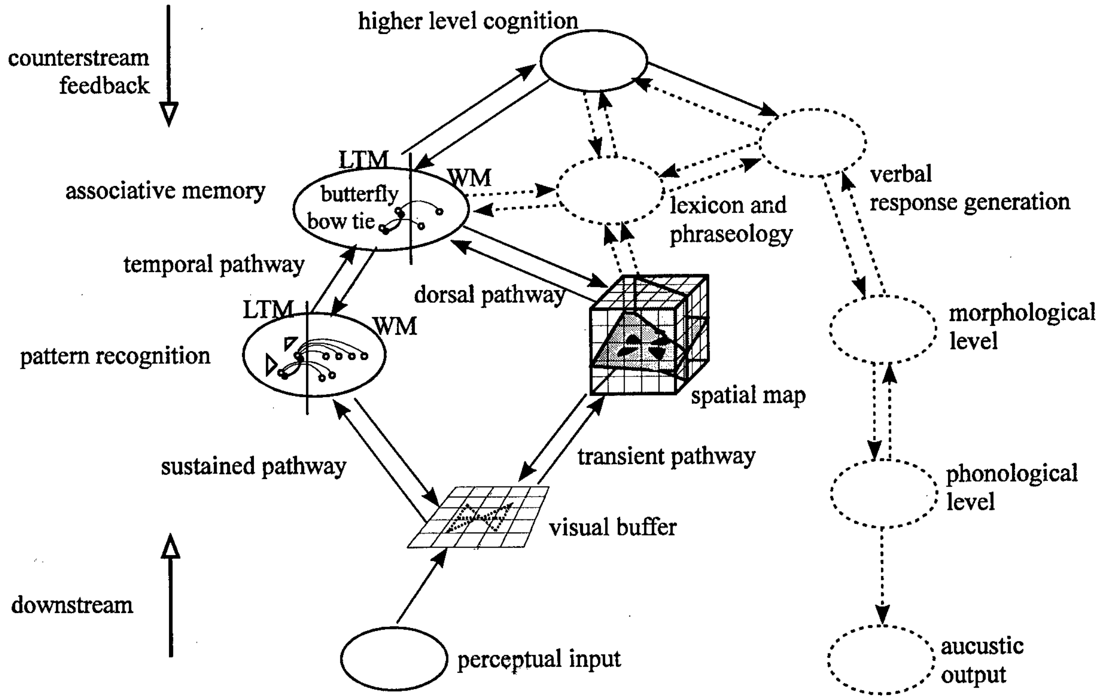


Figure 1. The visual system with its major subsystems at different levels of processing.

compared using McClelland & Rumelhart's interactive activation model (1981; 1994/1988) as a parametric framework. Our ambition has been to keep the set of working hypothesis concerning the neural architecture and processing style employed to a minimum, and instead explore the gaps which are left unspecified by empirical data.

Both models are discrete and deterministic, and can be conceived of as a set of constraints on inter-parameter dependencies within the envelope of the simulation framework. The two models stipulate that different parametric relations should hold in order for the simulation outcome to conform to the empirical constraints. *Simulation outcome* is measured as the relative conformance with empirically based constraints on how reinterpretation performance should change *when simulation is switched from perceptual to mental mode.*

Experimental data on mental image reinterpretation

Contrary to the classical findings on mental image reinterpretation difficulties, Finke, Pinker and Farah (1989) demonstrated that mental images can be as easy to reinterpret as perceptual images when the interpretations generated comprise verbal descriptions of *geometric patterns* contained in the image. Two types of reinterpretations seem to be involved: *Geometric reinterpretations*, when the composite image is described in simple geometric terms, for example, "two adjacent triangles pointing towards each other", and *symbolic reinterpretations*, in which the image is freely associated with an object or concept, for example, "a bow tie". In experiment 1 (Finke et al. 1989) relative performance rate for symbolic reinterpretations was on the average 30-50% of the possible total produced during imagery *and* perception. As opposed to this, up to 80-90% of the geometric reinterpretations were detected in the mental images proper.

VISUAL VERSUS SYMBOLIC REPRESENTATIONS

In a very general sense, qualitatively different styles of computation is afforded by symbolic representations as opposed to visual representations, with the main difference being that of accessibility in a linked versus a directly addressable data structure.

We operationalize these different assumptions, and would like to examine whether visual representations are needed as a *mediating link* between old and new interpretations in an interactive activation model. We have two possible hypothesis:

1. The subjective experience of "seeing mental images" is a non-functional side-effect of symbolic knowledge being activated in associative long term memory. No "*real*" image is formed in the visual buffer during imagery, so mental reinterpretations have to be based on inferences using "lateral" associations between symbolic representations.
2. A mental image is recreated in the visual buffer, and this image plays a pivotal role in mental reinterpretation. In this case, it is the image that drives processing towards a new interpretation, while the image's symbolic content acts as a source for indexing and sustaining, and thereby locking, the current interpretation.

Methodology

We evaluate the two representational hypothesis by freely exploring parametric variants of a simulation framework (Fig. 1) and by evaluating these variants against the empirical constraints of Finke, Pinker and Farah (1989). Simulation outcomes depend at the outset on the parameter constraints imposed by the individual models *plus* the following minimal assumptions about the neural architecture and processing style of the human visual system:

- Visual subsystems are hierarchically organized into processing levels.
- Adjacent processing levels in the visual system communicate with each other reciprocally.
- Visual processes operate in cascade.
- Mental imagery reuses parts of the visual system. In particular, images formed during mental imagery are assumed to reside in the visual buffer.

What is measured?

Keeping exploration of the parameter space within the envelope of the simulation framework and within the parametric constraints imposed by a particular model, simulation of the models should substantiate that whenever the system's transition behavior between perceptual and mental modes conforms to the empirical constraints, the parametric relations predicted by the models hold. Based on the interdependencies which can be detected when simulation results are systematically plotted against parameter combinations, the soundness of the two models can be evaluated and additional properties which necessarily follow from the two models can be exposed.

For a preliminary analysis of our simulation results, see www.ida.liu.se/~ritko

REFERENCES

Finke, R. A., Pinker, S. & Farah, M. J. (1989). Reinterpreting visual patterns in mental imagery. *Cognitive Science, 13*, 51-78

Kosslyn, S. M. (1994). *Image and Brain: The resolution of the imagery debate.* Cambridge, MA: MIT Press

McClelland, J. L. & Rumelhart. D. E. (1994/1988). *Explorations in parallel distributed processing: A handbook of models, programs and exercises.* Cambridge, MA: MIT Press

Pylyshyn, Z. W. (1981). The imagery debate: Analogue media versus tacit knowledge. *Psychological Review, 87*, 16-45

Modelling Individual Differences in Reasoning

Padraic Monaghan
Centre for Cognitive Science
University of Edinburgh
2 Buccleuch Place
Edinburgh, EH8 9LW, UK
+44 131 650 4415
pmon@cogsci.ed.ac.uk

ABSTRACT

The serialist-holist learning style distinction has received renewed interest due to its predictive power with regard to students' responses to new learning situations. In particular, individual differences in students' computer use indicate an area where knowing about style differences is of theoretical interest and practical import. This study concerns the differing responses of students to a computer-based logic program – Hyperproof – where serialist-holist style differences emerge spontaneously in the proofs produced by students. Proof style and strategy change are found to relate to independent measures of reasoning ability. These different strategies are analysed in terms of working memory load, and this points towards potential methods of modelling the serialist-holist learning style.

Keywords
Serialist-holist, reasoning, working memory, learning.

INTRODUCTION

Students use different strategies when they solve problems. Certain patterns of behaviour in new learning situations have been expressed in terms of the serialist-holist distinction. However, the environments where these differences have been diagnosed and observed have been complex, subjective, and lengthy, hence assessing contributing factors that influence performance has been difficult.

Currently, we have been studying a computer-based environment for problem-solving called Hyperproof (Barwise & Etchemendy, 1994) where serialist and holist strategies emerge spontaneously. This environment has the advantage over previous studies of learning strategies in that it is a constrained context within which variables can be manipulated, and detailed data on performance can be collected as students' interactions with the problem are logged by the computer.

This paper presents the background necessary for modelling serialist-holist learning styles, and offers a preliminary model of the interaction between changing problem requirements and strategy selection. Modelling differences in strategies within the restricted domain of Hyperproof will help to define what the serialist-holist distinction means from a cognitive perspective.

THE SERIALIST-HOLIST DISTINCTION

Pask (1976) used the serialist-holist distinction to describe the different strategies used by students in new learning situations. Serialists concentrate on concrete instances within the learning framework, building up an overall understanding of the situation by forming links between low-level features. In contrast, holists prefer to focus on the global structure of the learning situation, filling out the details once the structure has been explored. Roughly speaking, the serialist is a 'bottom-up' learner, whereas the holist's approach is 'top-down'.

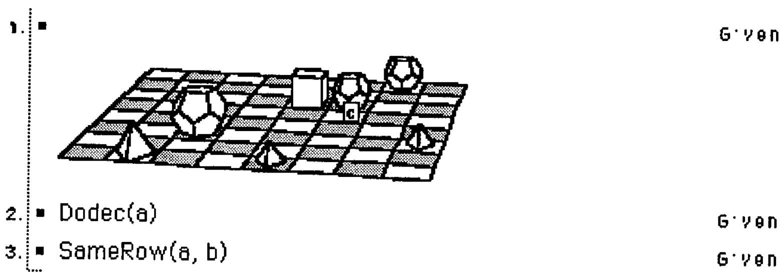
Versatile students will select the strategy that is most appropriate to the task, and this requires a combination of awareness of the task constraints and of the individual's own resource limitations and aptitudes. Pask has found that most students are inflexible in their approach to problems – a student that always uses one particular strategy when solving problems is said to have a learning 'pathology'.

These differences have proved to be ubiquitous and pervasive in a variety of different learning situations. In research on human-computer interaction, for example, the distinction does much to classify and predict the different responses of students to alternate interfaces (for a review see Helander, 1990, pp.541-580). Though important to learning, little computational or cognitive research has been directed towards defining or describing the different processes that underly each learning strategy.

HYPERPROOF

Hyperproof is a multimodal computer-based tool designed to teach first order logic through the dual presentation of a graphical situation and sentential descriptions of elements of the situation. The graphical situation is made up of objects of varying size and shape taking up positions on a chess-board. One particular type of problem requires the student to concretise an abstract situation: in order to solve the problem, the student must express graphically information that is given in a sentential (propositional calculus) form. A simple example of this type of task is illustrated in Figure 1.

In this problem the student is required to display all situations that are consistent with the given information. In short, the several ways that the labels 'a' and 'b' and the predicate information 'object a is a dodecahedron' and 'a and b are in the same row' have to be illustrated one after another in the graphical part of the window. There are two distinct strategies by which all the situations can be constructed. One method will apply all pieces of sentential information simultaneously in each situation, thus the strategy is a



1. ▪ Given

2. ▪ Dodec(a) Given

3. ▪ SameRow(a, b) Given

Figure 1: The Hyperproof problem.

case-by-case method and therefore serialist. The alternative method applies one piece of information at a time, with the second piece of information being superimposed onto the situation formed from applying the first sentential expression. As this proof is less concrete, it is interpreted as reflecting a holist strategy. This latter method is akin to constructing nested assumptions in a logical proof.

TOWARDS A COMPUTATIONAL MODEL

Serialist and holist strategies, as described with respect to the problem in Figure 1, have been observed in the proofs of students on a Hyperproof course (Cox, Stenning & Oberlander, 1994; Monaghan, 1998). These different uses of strategy have been related to an independent measure of reasoning ability (derived from the analytic reasoning section of the USA graduate recruitment exam (GRE)). Two Hyperproof problems solved under exam conditions were analysed. These questions contained as a main subtask the above type of problem, one question requiring the construction of three situations, the other requiring nine situations to be indicated. Students using a serialist strategy on the simpler problem and a holist strategy on the complex problem were better GRE reasoners than other groups, including the 'pathological' students who rigidly used only one strategy on the Hyperproof problems ($F(3, 18) = 5.69$, $p<0.01$). This suggests that there are general strategic approaches to complex problem solving situations that are more successful than others.

A preliminary model of the Hyperproof problem assessed the working memory load at each step in the proofs as a result of applying the different strategies. The holist strategy minimises working memory load, but more steps in the proof are required: seven to the serialist's five for the Figure 1 example. For students that are good at solving problems, strategy choice seems to be a pay-off between working memory load and the effort required to structure the solution. For simple problems, like the one illustrated, a serialist method may be more efficient. For more complex problems, a holist proof will reduce the working memory load.

The Hyperproof environment provides a suitable domain for studying serialist-holist strategies from a computational perspective. It also allows for a study of learning pathologies and strategy change under different conditions. A cognitive model of serialist-holist strategy use will have implications for several areas of cognitive science research. Principally, it will provide a formalism of what the different strategies mean from a computational perspective allowing better provision of resources in areas such as human-computer interaction. Also, insight into the cognitive properties of substeps in problem-solving procedures would result (Catrambone, 1996). Finally, the cognitive properties of external representations during problem-solving can be assessed (Scaife & Rogers, 1997).

ACKNOWLEDGMENTS

This work was supported by ESRC research studentship R00429634206.

REFERENCES

Barwise, J. & Etchemendy, J. (1994). *Hyperproof.* CSLI Lecture Notes. Chicago: Chicago University Press.

Catrambone, R. (1996). Generalizing solution procedures learned from examples. *Journal of Experimental Psychology: Learning, Memory, and Cognition 22*, 1020-1031.

Cox, R., Stenning, K., and Oberlander, J. (1994). Graphical effects in learning logic: reasoning, representation and individual differences. *Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society*. Pittsburgh: Lawrence Erlbaum Associates, 237-242.

Helander, M. (Ed.) (1990*). Handbook of Human Computer Interaction* (2nd edition). North Holland: Elsevier Science Publishers.

Monaghan, P. (1998). *Holist and Serialist Strategies in Complex Reasoning Tasks: Cognitive Style and Strategy Change.* (Research Paper EUCCS/RP-73). University of Edinburgh, Centre for Cognitive Science.

Pask, G. (1976). Styles and strategies of learning. *British Journal of Educational Psychology*, 46, 128-148.

Scaife, M. & Rogers, Y. (1997). *External Cognition: How do Graphical Representations Work?* (Cognitive Science Research Papers 335). University of Sussex, School of Cognitive and Computing Sciences.

A Feedforward Connectionist Account of Causal Discounting and Augmentation

Frank Van Overwalle & Dirk Van Rooy
Department of Psychology
Vrije Universiteit Brussel
Pleinlaan 2, B–1050 Brussel, Belgium
+32 2 629 25 18
Frank.VanOverwalle@vub.ac.be

ABSTRACT

We investigated the degree of discounting and augmentation of a target cause given varying frequencies of a competing cause. Several experiments showed that greater frequencies by which the competing cause covaried with the effect resulted in greater discounting or augmentation of a target cause. These competition size effects cannot be explained by current attribution theories in social psychology, but can be accounted for by a feedforward connectionist framework (Van Overwalle, 1998).

Keywords

Connectionism, Causal Judgments, Blocking.

INTRODUCTION

According to Kelley (1971), perceivers take into account not only how a possible factor covaries with the event, but also how this factor competes with rival factors that serve as alternative explanations. Despite the central place accorded to the covariation principle in attribution theory, Kelley (1971) argued that this principle in itself is insufficient to explain how perceivers select between competing causes. To account for such competition, Kelley (1971) proposed two complementary principles of discounting and augmentation.

The discounting principle specifies that if the influence of a cause is clearly established, perceivers will disregard other possible causes as irrelevant. The opposite tendency is described in the augmentation principle which specifies that if the inhibitory influence of a cause is firmly established, perceivers will overestimate the strength of a facilitatory cause to compensate for the inhibitory effect.

Our major question was whether discounting and augmentation of a target cause would be influenced by the frequency (or size) by which the competing cause covaried with the outcome. Based on a novel feedforward connectionist approach of causality (Van Overwalle, 1998), we predicted that greater frequencies would result in greater discounting or augmentation. Such competition size effect is not anticipated by current attribution theories in social psychology.

METHOD

In three experiments, the strength of competition was manipulated by varying how often the competing cause covaried alone with its outcome : Either one time (small size) or five times (large size). In contrast, the frequency of the target cause remained constant throughout all conditions. Type of competition was manipulated by pairing the competing cause with an outcome that was either similar to the target outcome (discounting) or opposite (augmentation). In addition, we manipulated the order in which the target information was presented (backwards or forwards) and the format of presentation (sequential trial-after-trial or summarized in short sentences).

RESULTS

Our results confirmed the feedforward connectionist account. First, in all experiments, we found that a higher frequency of covariation of a competing cause reliably increased the amount of discounting and augmentation of a target cause. These results are problematic for statistical models based on the notion of probability (e.g., Cheng & Holyoak, 1995) or of constraint satisfaction (Read & Marcus-Newhall, 1993). Second, the size effects were stronger when the information was presented in a sequential format, which is consistent with the feedforward connectionist view that the most natural way of processing causal information occurs on a trial-by-trial incremental basis. Third, there were no differences between forward and backward competition, supporting the notion that missing factors must be coded as absent as proposed by Van Hamme and Wasserman (1994).

REFERENCES

Cheng, P. W., & Novick, L. R. (1990). A probabilistic contrast model of causal induction. *Journal of Personality and Social Psychology, 58,* 545—567.

Kelley, H. H. (1971). Attribution in social interaction. In E. E. Jones, D. E. Kanouse, H. H. Kelley, R. E. Nisbett, S. Valins & B. Weiner (Eds.) *Attribution : Perceiving the causes of behavior* (pp. 1—26). Morristown, NJ : General Learning Press.

Read, S. J. & Marcus-Newhall, A. (1993). Explanatory coherence in social explanations : A parallel distributed processing account. *Journal of Personality and Social Psychology, 65*, 429—447.

Van Hamme, L. J. & Wasserman, E. A. (1994). Cue competition in causality judgments : The role of nonpresentation of compound stimulus elements. *Learning and Motivation, 25*, 127—151.

Van Overwalle, F. (1998) Causal Explanation as Constraint Satisfaction : A Critique and a Feedforward Connectionist Alternative. *Journal of Personality and Social Psychology*, in press.

Models of Two-person Games in ACT-R and SOAR

Frank E. Ritter
University of Nottingham
Nottingham NG7 2RD, UK
frank.ritter@nottingham.ac.uk

Dieter P. Wallach
Saarland University
66041 Saarbrücken, Germany
dwallach@cops.uni-sb.de

We were interested in understanding and comparing how ACT-R (Anderson & Lebière, in prep.) and SOAR (Newell, 1990) could each model a given dataset. We analyze and compare two models in their ability to account for a classical 2 person game, including the effort necessary to create and run them. In comparing the models and their results we provide two sample models and start to explore the potential role of abstract models and different types of data.

Game description. In two player, 2x2 games each player can choose one of two alternatives in each round. The players are rewarded according to a payoff matrix. The prisoner's dilemma is an example of such a 2 person game.

We used data from a classical experiment (Suppes & Atkinson, 1960) of how people learn when they play a normal form, two player 2x2 game with a nontrivial unique mixed strategy equilibrium. Table 1 shows the payoff matrix used in the experiment that we model here. This matrix has a unique mixed strategy equilibrium point, that is, a stable set of strategies, when Player 1 chooses option A1 with probability 1/3 and player 2 chooses option A2 with probability 5/6. Figure 1 shows the empirical choice frequencies of option A for player 1 (A1) and player 2 (A2) aggregated in 5 blocks with 40 rounds each, of 20 pairs of participants playing the game for 200 rounds (Erev & Roth, 1998).

ACT-R model. Figure 2 shows the structure of the ACT-R model used to account for this data. For a full description of the ACT-R model see Bracht, Wallach and Lebière (1998). The model consists of two simple productions for each player representing the options available:

Rule1: If Player 1 chooses => choose Option A.

Rule2: If Player 1 chooses => choose Option B.

In every round, both of these productions are applicable for each player modeled. ACT-R's subsymbolic cost learning mechanism learns the relative payoff of each production rule and updates their expected gain based on the outcome of the round. In general, ACT-R selects the production rule with the highest expected gain. Two architectural parameters were used to fit the model to the data (*expected gain noise* and *number of previous production applications*). The model with the same parameter settings has also been applied successfully to data from three other experiments taken from Erev and Roth (1998).

		Player 2	
		Option A	Option B
Player 1	Option A	2, 4	6, 0
	Option B	3, 3	1, 5

Table 1. Payoff matrix used by the models here.

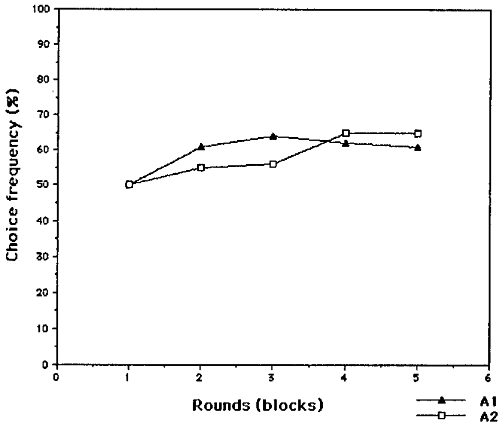


Figure 1. The evolution of strategies in the subjects on the Table 1 payoff matrix

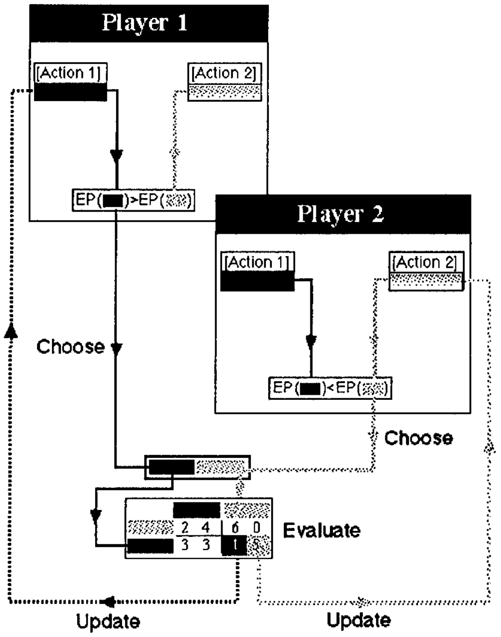


Figure 2. Description of the ACT-R model.

SOAR model. The easiest way to explore a SOAR model of this task is to create an abstract model. An abstract model is based on an information processing model or architecture. It predicts what a running model would do, without implementing the internal behaviors (e.g. Langley, 1996; Ohlsson & Jewett, 1994).

An abstract model of the simplest SOAR model could start with a single operator representing each choice. Each round, an operator is randomly chosen to apply. After each round, the expected values of each of the four payoffs occurring can be computed for each player. Operators that do better than the average payoff can be duplicated through a reflection-like process (not specified, but similar to the process in Bass et al., 1995). Various other ways of duplicating operators are possible (e.g. duplicate operators as many times as their payoff). In SOAR these processes are determined not by the architecture but by knowledge. It is fairly straightforward to implemented a program to compute the expected population of operators on each round. The results of this program are shown in Figure 3. While this model is not currently based on a running Soar model, creating such a model should be straightforward. Deriving its predictions is much simpler as an abstract model, for programming an interface to record multiple rounds and games would be less straightforward.

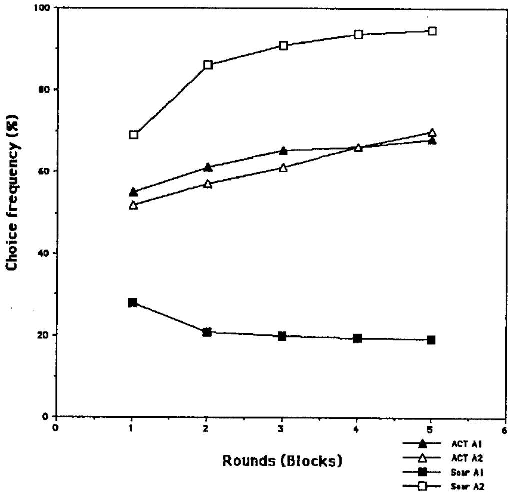


Figure 3. The evolution of strategies in the two models on the Table 1 matrix.

Comparisons

Model fit. As Figure 1 shows, the ACT-R model captures the general tendencies in the empirical data quite nicely. In addition to this *short term* prediction, the model converges asymptotically to the equilibrium of classical game theory in the *long term* (after >1500 rounds). The initial Soar model, on the other hand, does not match the subject data (short term) nearly as well, but instead appears to quickly converge to near the equilibrium.

Effort. Both models took about the same time to implement (4-5 hours), including the ability to automatically run and trace the model. Both models can run 200 rounds of 20 subject pairs in under 30s.

Abstract models. The Soar model would not be as easy to run if it was implemented in Soar productions. It would not be straightforward to implement an abstract version of the ACT-R model based on its current mechanism, but it is easy to create an abstract model of the operator population model in ACT-R (as a rule population), or an ACT-R model directly based on this principle. The difficulty of creating abstract models within each architecture varies by task, but appears to be generally easier in SOAR. Creating full models appears, however, to be more difficult. In this task, the SOAR architecture appears to have less to say than ACT-R because it lacks architectural mechanisms to account for the learning observed here. While the Soar model does not match nearly as well (yet), it allows the space of possible models to be explored quite quickly (about 5 min. per model).

Conclusions

These results are very interesting, for they start to suggest possible trade-offs in modeling; between abstract and information processing models, and between architectures. This work also emphasizes the role of usability as a necessary precondition for explorations of this kind.

REFERENCES

Anderson, J. R. & Lebière, C. (in prep.). *The atomic components of thought.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Bass, E. J., Baxter, G. D., & Ritter, F. E. (1995). Using cognitive models to control simulations of complex systems. *AISB Quarterly*, 93, 18-25.

Bracht, J., Wallach, D. & Lebière, C. (1998). On the need and performance of cognitive game theory: ACT-R in experimental games with unique mixed strategy equilibria. To appear in *The Economic Science Association (ESA) Annual Conference.*

Erev, I. & Roth, A. (1998, in press). Predicting how people play games: Reinforcement learning in experimental games with unique, mixed strategy equilibria. *American Economic Review.*

Langley, P. (1996). An abstract computational model of learning selective sensing skills. In *Proceedings of the 18th Annual Conference of the Cognitive Science Society.* 385-390. Hillsdale, NJ: Lawrence Earlbaum Associates.

Newell, A. (1990). *Unified theories of cognition.* Cambridge, MA: Harvard University Press.

Ohlsson, S. (1995). Abstract computer models: Towards a new method for theorizing about adaptive agents. In *Machine Learning: ECML-95.* Berlin: Springer-Verlag.

Suppes, P. & Atkinson, R. C. (1960). *Markov learning models for multiperson interactions.* Stanford, CA: Stanford University Press.

A connectionist account of Illusory Correlation

Dirk Van Rooy
Department of Psychology
Vrije Universiteit Brussel
Pleinlaan 2, B–1050 Brussel, Belgium
+32 2 629 26 03
dvrooy@vub.ac.be

ABSTRACT

Illusory correlation occurs when perceivers make an erroneous judgment of a relation between two or more unrelated categories. In this study, subjects read information about members of 4 groups, which differed in size : Group A contained twice as much behaviors as group B, group B twice as much as C and so on. The behavioral information about these groups was identical, in that 33% of the behaviors engaged in by the members were undesirable and 67% desirable. Preliminary results show that a greater amount of members in each category leads to a decrease of the illusory correlation effect. These results can be readily accounted for by a feedforward connectionist framework (Van Overwalle, 1998).

Keywords

Illusory Correlation, connectionism.

INTRODUCTION

Illusory correlation occurs when perceivers make an erroneous judgment of a relation between two or more unrelated categories. The original demonstration by Chapman (1967) showed how subjects overestimated the co-occurrence of long words in the context of a list of relatively short words. Presumably, the distinctiveness of the long word pairs led to a more thorough processing, which led to the illusory correlation effect.

Hamilton and Gifford (1976) applied this mechanism to the formation of group stereotypes. In their study subjects read statements about members of a majority group, labeled A, and a minority group, labeled B. Both groups revealed the same ratio of desirable to undesirable behaviors. After reading the statements, subjects overestimated the frequency of negative behaviors by group B members and also had a more negative impression of group B. According to Hamilton and Gifford, the less frequent and therefore more distinct undesirable group behaviors apparently received more extensive encoding. This probably led to greater accessibility in memory, leading to errors in frequency estimation and impression formation.

Recently several studies challenged the distinctiveness-paradigm (Smith, 1991, Fiedler, 1991). These studies claim that the phenomenon is not so much the consequence of mere distinctiveness of the stimuli, but simply reflects the general working of the human memory. Although this critique is well elaborated, it leaves certain question unanswered. The aim of the present research is to answer these questions by approaching the illusory correlation phenomenon form a connectionist angle.

The aim of the present research is to approach the illusory correlation phenomenon from a connectionist angle. Our connectionist approach depicts learning as a gradual process, during which associations between group membership and desirability are formed instantaneously. Every time a member of a certain group performs a (un)desirable behavior, the association between that group and (un)desirable behavior in general becomes stronger. As more learning takes place, these associations become stronger and are easy to discriminate, so the perceiver can form a relatively correct impression of a group based on these associations. However when these associations between group membership and desirability are weak, they are hard to discriminate and judgments will be prone to illusory correlation effects. Therefore, the main prediction of our connectionist model is that an increase in the amount of behaviors will lead to a decrease in the illusory correlation effect. Although apparently trivial, this effect is not a straightforward prediction of the distinctiveness hypotheses or any other recent model.

METHOD

Methodology and instructions followed the Hamilton and Gifford (1976) paradigm. Table 1 summarizes the distribution of the behavioral information for the 4 groups.

Table 1

Number of desirable and undesirable behaviors assigned to each group

Group :	A	B	C	D
Desirable behaviors	16	8	4	2
Undesirable behaviors	8	4	2	1

Subjects sat at individual computers and were told that the experiment concerned "the way people process and retain information". Furthermore they were told that they would receive information concerning four groups (A, B, C and D), these groups represented groups in the real world and that group A was bigger than group B, group B bigger than group C and so on. Finally they were told to read each statement carefully. Each statement remained on the screen until the subject pushed the space bar. After reading all statements, subjects completed a filler task, a free recall task, a group assignment task, a frequency estimation task and a group evaluation task.

RESULTS

Overall, the results confirmed our hypotheses. We expected that groups with more members (or behaviors) would be less subject to illusory correlation. Specifically this means that as the groups became smaller, group evaluations would become less favorable and relatively more undesirable behaviors would be attributed to these groups.

Likability Ratings. The main effect of group was significant, F(3, 72) = 4,62, p < ,005, revealing as expected that groups were rated less favorable as they became smaller.

Frequency Estimation. There was no significant main effect of group (p > ,1). However, contrast analyses show that subjects tended to attribute less undesirable behaviors to Group A than to other groups, F(1, 24) = 3.93, p < .06. This might indicate that only for group A the association between group membership and desirability was well established, enabling subjects to make a fairly accurate judgment.

Group Assignment. Analyses showed that subjects were more likely to assign desirable as opposed to undesirable behaviors to group A, F(1,24) = 4.419, p < .05. This confirms our prediction that for group A the associations between group membership and desirability are strong and therefore easy to discriminate. As subjects experienced more desirable group A behaviors then undesirable, the association between group A and desirable behavior is stronger than the association with undesirable behavior, leading to a tendency to assign more desirable behaviors to group A. The contrast analyses show the reverse effect for group D, in that more undesirable as opposed to desirable behaviors were assigned to this group, although this was only marginally significant, F(1,24) = 3.841, p = .06. This is probably due to the fact that there was only 1 undesirable behavior in group D, which would have made it very distinctive.

Free Recall. Two separate proportions were used : General free recall reflects the recalled behaviors regardless of whether they were correctly associated with a group. Correct free recall reflects only those behaviors correctly assigned to a group.

With respect to general free recall we see as predicted that relatively more undesirable behaviors are attributed to groups B,C and D in comparison with group A, F(1, 57) = 5.11, p < .03. This can be due to the overall response bias to attribute negative behaviors to smaller groups in parallel with the likability of those groups. As stated before, this confirms our prediction that the associations between group membership and desirability are weak for the smaller groups, leading to illusory correlations.

There is however another possible explanation. According to our connectionist model, during learning strong associations tend to suppress weaker associations (competition effect). For instance, that would mean that the strong association between group membership and desirability for group A would suppress the associations of the unique behaviors with that group. This would be less the case for the smaller groups, where the associations between group membership and desirability are weaker. As a result, more unique behaviors should be recalled by the subjects as the groups become smaller. In fact this is partly confirmed by the data for correct free recall : undesirable behaviors were recalled better than desirable for group D, F(1, 57) = 5. 82, p < . 03. However, the data for the other groups show no sign of this competition effect as recall is weak in all these cells. This is nonetheless an important aspect of connectionist learning models, as it easily explains distinctiveness effects. Hence further research into this matter is required.

REFERENCES

Chapman, L.J. (1967). Illusory correlation in observational report. *Journal of Verbal Learning and Verbal Behavior, 6,* 151—155.

Fiedler, K. (1991). The tricky nature of skewed frequency tables : An information loss account of distinctiveness-based illusory correlations. *Journal of Personality and Social Psychology, 60,* 24—36.

Hamilton, D. L., & Gifford, R.K. (1976). Illusory correlation in interpersonal perception : A cognitive basis of stereotypic judgment. *Journal of Experimental Social Psychology, 12,* 392—407.

Smith, E. R. (1991). Illusory correlation in an simulated exemplar-based memory. *Journal of Experimental Social Psychology, 27,* 107—123.

Van Overwalle, F. (1998) Causal Explanation as Constraint Satisfaction : A Critique and a Feedforward Connectionist Alternative. *Journal of Personality and Social Psychology*, in press.

Analogical Problem Solving by Adaptation of Schemes

Ute Schmid

Department of Applied Computer Science, Technical University Berlin
FR 5-8, Franklinstrasse 28, 10587 Berlin, Germany
+49 30/314-23938, schmid@cs.tu-berlin.de

Abstract. We present a computational approach to the acquisition of problem schemes by learning by doing and to their application in analogical problem solving. Our work has its background in automatic program construction and relies on the concept of recursive program schemes. In contrast to the usual approach to cognitive modelling where computational models are designed to fit specific data we propose a framework to describe certain empirically established characteristics of human problem solving and learning in a uniform and formally sound way.

1 Introduction

The use of analogies is a powerful and ubiquitous strategy in human reasoning and problem solving. A lot of (symbolic, connectionist & hybrid) computational models have been proposed with the aim of getting more precise insights in the underlying processes (Anderson & Thompson, 1989; Falkenhainer, Forbus, & Gentner, 1989; Hummel & Holyoak, 1997) and with the aim of exploiting this strategy in AI applications (cf. case based reasoning).

Most of the computational models are focusing on analogical access and mapping thereby neglecting two crucial aspects of analogical problem solving: (1) generation of problem representations which are suitable for analogical problem solving (i.e. problem schemes), and (2) solving a target problem by adapting a - not necessarily isomorphical - source problem.

The model proposed by Anderson and Thompson (1989), for example, relies on schemes for representing the structure of problems and solutions which are available to the system from the beginning. Thereby the authors suppose that the system has already knowledge about the structure of the problem domain. But the crucial deficit of novices is that they have *no* knowledge about the structural characteristics relevant for problem solving (Novick, 1988; Schmid & Kaup, 1995). Otherwise, there would be no need for analogical problem solving. The problem could be solved by applying already acquired automatisms (production rules) or abstract schemes.

The examples Anderson and Thompson (1989) give for analogical transfer are restricted to generalized problem isomorphs, i.e. identical structures where predicate and operation symbols can be substituted in a unique way. There is no statement whether the model could be extended to adaptation of non-isomorphical structures. In everyday reasoning, availability of isomorphical source problems is the exception. Empirical studies demonstrate that people also *can* use partially isomorphical source problems (Pirolli & Anderson, 1985; Schmid & Kaup, 1995).

We are proposing a framework for analogical problem solving which overcomes the limitations described above: First we present our concept of problem schemes and a method for inferring such schemes from problem solving experiences. Than we describe our approach to analogical transfer which works for both isomorphical and non-isomorphical source problems.

2 Induction of Problem Schemes

The central concept of our approach is the notion of recursive program schemes (RPSs; see Schmid & Wysotzki, 1998 for the formal definitions). An RPS represents the structure of a problem as (recursive) equation. On the left side the name of the RPS and its parameters are given. The right side represents a operations together with their conditions for application. An RPS representing the knowledge of clearing a block is

clear-one-block(x, s) = if cleartop(topof(x)) then puttable(topof(x)) else s.

The variable *s* ("situation variable") represents the current problem state (for example *on(A, B), on(B, C), cleartop(A)*). This RPS can only be applied if *one* block is lying on block *x*. For the problem state given above it can be applied to block *B* only. An RPS representing the knowledge of clearing an arbitrary block in a tower is

clearblock(x, s) = if cleartop(x) then s else puttable(topof(x), clearblock(topof(x, s))).

The representation format of an RPS simultaneously catches the *structure* of a problem and its executable *solution strategy* (cf. Rumelhart & Norman, 1981).

In our program IPAL (Schmid & Wysotzki, 1998) we are modelling the acquisition of RPSs by a two-step process: In a first step some initial states of a problem are solved by applying predefined production rules using heuristic search. That is, without experience in a problem domain the system has to use a general purpose strategy which can be inefficient because search may lead to dead ends and there is need for backtracking. The solution sequences found for the initial states are composed into a so called initial program generalizing over the application conditions of the predefined production rules and over the constants occuring in the initial states.

An initial program which represents the experience with towers up to three blocks is

```
if cleartop(x) then s else
 if cleartop(topof(x)) then puttable(topof(x),s)
  else if cleartop(topof(topof(x)))
   then puttable(topof(x),
         puttable(topof(topof(x)),s)).
```

Building initial program corresponds roughly to chunking and refinement of production rules.

By using a method for inductive program synthesis initial programs can be generalized to RPSs (Schmid & Wysotzki, 1998). The general idea of our algorithm is to identify a pattern and a substitution in the initial program which makes it possible to reproduce the whole structure. For the initial program given above the pattern is *if cleartop(x) then s else puttable(topof(x), m)* with the substitution $x \leftarrow topof(x)$. If found, the pattern and substitution are extrapolated to an RPS. This process describes a fundamental aspect of human intelligence: the ability of induction as for example described by (Holland, Holyoak, Nisbett, & Thagard, 1986).

3 Transformation Based Adaptation

RPSs formally are elements of a term algebra. That means, they represent syntactical structures only. The semantics of an RPS is gained by interpretation of the symbols in accordance to some domain model. Thereby an RPS represents the class of all structurally identical problems. This is a characteristic extremely suitable for analogical reasoning.

In IPAL already inferred RPSs are stored in memory. An RPS can be "unfolded" to an initial program again. If a new initial program is gained by investigating some problem states, the memory is checked for an RPS whose corresponding initial program is similar to the new one. In that case inference of a general solution strategy for the problem (represented by an RPS) can be omitted and the known RPS can be adapted to the new problem instead.

In our approach mapping and adaptation is performed by means of tree transformation. An initial program of an already known RPS is transformed to a new initial program by substitution, insertion and deletion of symbols. The set of transformations can than be applied to adapt the known RPS. Two initial programs are isomorphical if one can be transformed into the other by a set of unique substitutions only. We give an example of adaptation in the non-isomorphic case (see fig. 1). To transform "clearblock" into "factorial" we have to perform the unique substitutions *cleartop/equal0*, *s/1*, *puttable/mult*. Additionally we have two transformations for the "topof" symbol: substitute *topof/pred* and delete *topof*. By using contextual information, we can decide at which position in the RPS *topof* has to be deleted (in the first argument of *puttable* rsp. *mult*).

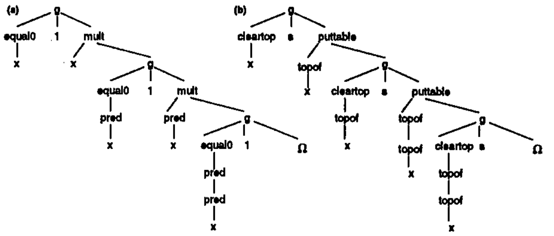


Fig. 1. Initial programs for (a) factorial and (b) clearblock ("g" represents the conditional "if-then-else"; *Ω* stands for "undefined")

4 Discussion and Further Work

We believe that our framework can be of use for the cognitive modeling community for two reasons: (1) it addresses the problem of scheme acquisition often neglected in computational approaches, (2) the notion of RPSs as representation format makes it possible to describe the structural characteristics of problems in a way which makes it possible to perform precise analyses of the structural similarity of problems. This can be useful to construct source-target relations for empirical studies of the conditions of successful adaptation in analogical problem solving.

References

Anderson, J., & Thompson, R. (1989). Use of analogy in a production system architecture. In S. Vosniadou & A. Ortony (Eds.), *Similarity and analogical reasoning* (p. 267-297). Cambridge University Press.

Falkenhainer, B., Forbus, K., & Gentner, D. (1989). The structure mapping engine: Algorithm and example. *Artificial Intelligence*, *41*, 1-63.

Holland, J., Holyoak, K., Nisbett, R., & Thagard, P. (1986). *Induction - processes of inference, learning, and discovery.* Cambridge, MA: MIT Press.

Hummel, J., & Holyoak, K. (1997). Distributed representation of structure: A theory of analogical access and mapping. *Psychological Review*, *104*(3), 427-466.

Novick, L. R. (1988). Analogical transfer, problem similarity, and expertise. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *14*, 510-520.

Pirolli, P., & Anderson, J. (1985). The role of learning from examples in the acquisition of recursive programming skills. *Canadian Journal of Psychology*, *39*, 240-272.

Rumelhart, D. E., & Norman, D. A. (1981). Analogical processes in learning. In J. R. Anderson (Ed.), *Cognitive skills and their acquisition* (p. 335-360). Hillsdale, NJ: Lawrence Erlbaum.

Schmid, U., & Kaup, B. (1995). Analoges Lernen beim rekursiven Programmieren (Analogical learning in recursive programming). *Kognitionswissenschaft*, *5*, 31-41.

Schmid, U., & Wysotzki, F. (1998). Induction of recursive program schemes. In *Proceedings of the 10th European Conference on Machine Learning (ECML-98).* Springer.

Toward a theory of the control of dynamic systems

Wolfgang Schoppek
Department of Psychology
University of Bayreuth
D-95440 Bayreuth,Germany
+49 921 555003
wolfgang.schoppek@uni-bayreuth.de

ABSTRACT

A categorization of three types of knowledge which can be relevant for the control of dynamic systems is suggested. These are (1) input-output knowledge, (2) structural knowledge which is subdivided in knowledge about effects and knowledge about dependencies, and (3) strategic knowledge. The assumptions are embedded in the theoretical framework of the ACT-R theory. An ACT-R model of the early stages of knowledge acquisition, and its implications for future research are described.

Keywords

knowledge acquisition, causal relations, ACT-R, dynamic system

INTRODUCTION

This contribution deals with the control of dynamic systems of the following type: There are about 2-4 input-variables which are exclusively controlled by the problem solver, and about the same number of output-variables whose values depend on the values of input- and output-variables. The systems are modelled by simultaneous linear equations. In order to minimize the variability of domain specific knowledge, the variables have phantasy names. As a consequence, only general prior knowledge, e.g. knowledge about causal relations, can be brought to bear in the problem solving process. Fig. 1 shows a simple example of such a system.

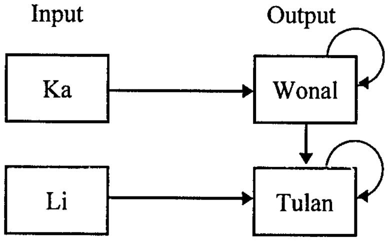


Fig. 1: A simple dynamic system

The control of dynamic systems is a form of complex problem solving. Unlike many other problem solving tasks, the effects of the operators are not explained in the instructions. The problem solver has to induce them by analyzing self generated state-action-state sequences.

Many authors assume, that controlling systems effectively requires structural knowledge. The notion of structural knowledge comprises knowledge about the variables and their causal relationships. But the results concerning the relation between structural knowledge and control performance are inconsistent. In some studies subjects report considerable structural knowledge, but fail to attain the goals for system control (Schoppek, in prep.). In other studies subjects are successful in controlling the system but can hardly report anything about its structure (Berry & Broadbent, 1984). There is, however, also evidence for a convergence of structural knowledge and control performance (Funke, 1992). It is obvious, that the construct of structural knowledge is too undifferentiated to account for the diversity of the results.

TYPES OF KNOWLEDGE FOR SYSTEM CONTROL

As a step towards an integrative explanation of these results I want to suggest a theoretical distinction of three different types of knowledge which can be relevant for the control of dynamic systems.

(1) Input-output knowledge (I-O-knowledge) represents interventions and their effects. These may be stored either external or in declarative memory. In early exploration phases I-O-knowledge is the material from which structural knowledge is induced. With extended practice, successful I-O-sequences can be recalled directly from declarative memory. A third possibility of using I-O-knowledge is the successive adjustment of an input-pattern without any induction of general rules.

(2) Structural knowledge is subdivided in two types: knowledge about effects (E-knowledge) and knowledge about dependencies (D-knowledge). E-knowledge is supposed to be acquired from an early stage of practice with the system. It can be induced quite easily from state-action-state sequences, provided that an appropriate input-strategy is applied. E-knowledge can be represented by solitary chunks. It is sufficient to answer most of the questionnaires that have been used to assess structural knowledge.

But the exact control of a dynamic variable requires knowledge about its dependencies. It is possible to search memory for all E-chunks containing the goal-variable in its output slot, but this is an error-prone procedure. In this situation an output-centerd integration of E-knowledge would be more effective. This is the hypothetical D-knowledge. Successful problem solvers seem to have access to this type of knowledge since they have no difficulties in quickly considering all dependencies of an output variable. D-knowledge can be deducted from E-knowledge, but this is an additional process. Thus deduction and use of D-knowledge takes more effort than induction of E-knowledge.

(3) Strategic knowledge comprises knowledge about how to acquire structural knowledge, (e.g. the strategy of isolated variation of conditions), and knowledge about certain input-strategies (e.g. the compensation of side-effects).

The three types of knowledge are differing in their generalizability. I-O-knowledge is only applicable for a single system and is goal specific. Structural knowledge refers to a single system, too, but is unspecific with respect to the goal states. Finally, strategic knowledge can be applied in the exploration and the control of many different systems.

THEORETICAL INTEGRATION

The assumptions are embedded in the theoretical framework of the ACT-R theory (Anderson, 1993). All the types of knowledge are supposed to consist of both declarative and procedural elements, whose parameters change with use according to ACT-R. Thus the theoretical distinction could serve as a link between the content-independent assumptions of the ACT-R theory and more specified models of system control.

EMPIRICAL SUPPORT

The assumptions are largely consistent with the data. Dissociations between verbalizable knowledge and control performance can be explained by the notion that most tasks for assessing structural knowledge can be solved with E-knowledge whereas successful system control requires more than access to single E-chunks. Findings that initial dissociations disappear with extended practice (Sanderson, 1989) are also in line with this explanation. Seemingly inconsistent results of tutoring structural knowledge, which were found in experiments of our workgroup are interpretable in terms of different focuses of the training procedures. A training which focused on D-knowledge (Preußler, 1997) lead to improved control performance whereas a training which focused on E-knowledge did not (Schoppek, in prep.).

ACT-R MODEL

I started to put these deliberations into practice in form of an ACT-R model which handles the static system depicted in fig. 2. At present the model is able to explore the system. It induces positive effects on the base of self generated data and creates single E-chunks for every detected effect. With this knowledge the model can produce judgements about effects in a fact-retrieval paradigm. Finally the model can use its E-knowledge to obtain simple goal states.

The main problem in this early stage of model construction is to find an appropriate representation of new causal knowledge. As indicated above, the model creates a new chunk for every detected effect. The chunk-type has three slots: „input“, „output“, and „factor“. This takes into account that judgements about causal relations cannot be explained by the assumption of simple associations between cause and effect (Waldmann, 1996).

In the fact-retrieval task the model exhibits no effect of the number of outputs that are affected by an input (e.g. judgements of „Eltan-Ordal“ and „Bulmin-Fontil“ take the same time, although Bulmin affects only one output whereas Eltan affects three).

In a preliminary experiment five subjects explored the static system shown in fig. 2 and then processed the fact-retrieval task with pairs of variable-names. In contradiction to the model, there seems to be a fan effect: The judgements for the effects of input „Eltan“ (fan 3) take longer than the judgments for the effects of „Bulmin“ and „Dulan“ (fan 1). This might, however, be due to the fact, that the judgements were based on a secondary verbal representation of a rather sensorimotor primary representation of the effect. Indeed, four of the five subjects reported that they memorized the effects in terms of locations and that memorizing the names was an additional demand. In the main experiment it will be tested if there are different effects depending on the presentation of spatial cues.

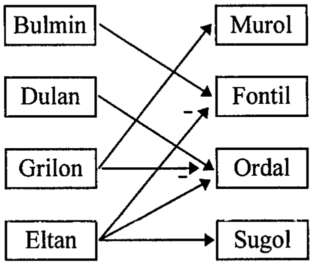


Fig.2: Static system controlled by the model

OPEN QUESTIONS

Thus even the initial representation of single causal relations can be regarded as an open question. A more serious problem is posed by the question, how the hypothetical D-knowledge is transformed into productions. Experienced problem solvers obviously dispose of such fairly complex productions.

Despite all those open questions I hope to have pointed out that there is a long way between the acquisition of single effect-chunks, including their application in fact-retrieval tasks, and the integrated use of this knowledge for the determination of input-values in order to obtain specific goal states.

REFERENCES

Anderson, J. R. (1993). *Rules of the mind.* Hillsdale, NJ: Lawrence Erlbaum Associates

Berry, D. C., & Broadbent, D. E. (1984). On the relationship between task performance and associated verbalisable knowledge. *The Quarterly Journal of Psychology, 34A,* 209 -231.

Funke, J. (1992). Dealing with dynamic systems: research strategy, diagnostic approach and experimental results. *The German Journal of Psychology, 16,* 24-43.

Preußler, W. (1997). Zur Rolle expliziten und impliziten Wissens bei der Steuerung dynamischer Systeme. *Zeitschrift für Experimentelle und Angewandte Psychologie, 43,* 399-434.

Sanderson, P.M. (1989). Verbalizable knowledge and skilled task performance: association, dissociation, and mental models. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 15,* 729-747.

Waldmann, M. R. (1996). Knowledge-based causal induction. *The Psychology of Learning and Motivation, 34,* 47-88.

A semi-symbolic cognitive model of usage polysemy

Sylvain Surcin
E.C.Art. / L.R.I.A. – University of Paris VIII
21, rue Baudelique
75018 Paris, France
tel.: +33 1 44 92 83 24
surcin@winimage.com

INTRODUCTION

The status of polysemy as a source of lexical ambiguities is still not clear neither among linguists, nor among computational semanticists. Here, we take a step outside of the classical debate among homonymy and vagueness and we postulate that polysemy is at the origin of most of the cases of ambiguity.

In this paper, we focus on a particular kind of polysemy that we call *usage polysemy*. Unlike other kinds of polysemy, usage polysemy cannot be reduced to operations of sense composition and selection. Usage polysemy takes place when a polysemous lexical unit has several closely related interpretations, corresponding to different uses and none can be said to be 'the right one'. It will be further defined in section THEORETICAL AND EMPIRICAL APPROACHES. Our aim is to design a cognitive model for the computation of usage polysemy, and to implement it as an expert agent cooperating with other agents in a Natural Language Processing (NLP) architecture.

We present our model in the PELEAS MODEL section. We designed it on the main postulate that interpreting usage polysemy is a process similar to translating an ambiguous expression. We also present the set of software pieces we developed around our model, along with a qualitative evaluation we have conducted at the time being. At last, in the CONCLUSION section, we give our temporary conclusion about our model.

THEORETICAL AND EMPIRICAL APPROACHES

A widely spread opinion among computational linguists is that polysemy is a false problem, and the ambiguities it generates are but artefacts produced by our models. The argument is that we, human beings, never fail when interpreting polysemy. But what to think about sentences like "The mother cell splits into two new identical cells"? Which is the right interpretation for "mother": generating, antecedent, prior, ruling or causal source? As a matter of fact, a human reader does not feel annoyed when reading such a sentence, because he/she unconsciously handles all the different interpretations simultaneously. We will show thereafter that this example falls in a particular category of polysemy we call *usage polysemy*, which, indeed, is not a problem once we do not require *the* right interpretation for a polysemous word.

Different kinds of polysemy

Lexical ambiguity has been abundantly studied and modelled by computational linguists. But what is usually referred to as 'polysemy' is described as *functional polysemy* by Prince and Bally-Ipsas (1991). It involves semantic features as much as syntactic ones in order to resolve the lexical ambiguities it generates, by restricting the selection of the concept which is compatible with the context. An other category of polysemy is described by Rastier (1996) as *sense polysemy*[1] and involves linguistic devices known as *isotopy* and *isosemy* in differential semantics. The last kind of polysemy we can distinguish involves also sociolinguistics data, as conventional uses of words, tropes and topoï. This is precisely this category we study here.

Usage polysemy

Our framework is composed of polysemous word occurrences for which there are no syntactic / semantic necessary and sufficient conditions, nor intralinguistic isotopy relationships allowing us to discriminate between the different possible interpretations. This means that all interpretations are closely related conceptual *points of view* on the word's meaning. They differ only by slight shades of meaning for the word's *usage*. These shades can be established in discourse, on a cultural basis.

Such phenomena have been observed by Tanaka and Umemura (1994) to occur frequently for *common words* (representing approximatively 30% of the lexicon for any given language). Common words are not terms: they are not used as items of a nomenclature but rather in the everyday discourse.

Usage polysemy of common words may arise in a various set of situations: (i) *usage transfers*: when a word is used outside of its most usual application field, mostly in order to illustrate a technical concept; (ii) *deliberate sense overlapping*: when an author play with the lexical ambiguity due to polysemy in order to describe a complex situation in a limited textual space; (iii) *joker words*: when a common word is so much used inside a linguistic community that its semantic contents becomes too generic; and (iv) *plays on words* referring to cultural references shared by the locutors.

Interpretation and translation

The most adapted linguistic theory for studying usage polysemy seems to be the differential semantics theory. However, it is too fuzzy to be implemented straight away, and does not account for the influence of sociolinguistic data on the behaviour of lexical units. Sticking to the interpretation paradigm of the differential theory, we

[1] The original 'polysémie d'acception' could be better translated into 'polysemy of linguistic aspects of the senses'

postulated that interpretation is similar to translation in a certain way. That is why we observed a team of technical translators resolving problems raised by cases of usage polysemy. Their procedure seems to be incremental and hierarchical: (i) to find a general semantic direction by probing the global context, (ii) to restrict the set of possible interpretations by finding textual markers in the local context, (iii) to list valid and plausible interpretations by using inhibition and reinforcement, and (iv) to produce a synthesised translation.

THE PELEAS MODEL

The model we designed is called PELEAS (Pyramids and Ellipses as Lexical Entries in Ambiguous Sentences). It is a lexicon driven by lexical entries, but each entry owns semasiological substructures.

Description of the model

The model was designed as a *dynamic lexicon*: it does not contain all possible interpretations of a word, but rather computes them from a minimalist static representation of well acknowledged uses. That is why it is constituted of a static part (this representation) and a dynamic part, which handles the salience attribution process. This model is in the same trend of representations as the Generative Lexicon of Pustekovsky (1991) and Edgar of Prince (1994) Our model differs from the Generative Lexicon because it does not try to specify the relationships between a word and its description further than "the descriptors of a word lexically co-occur in the close context of its occurrences". It also differs from Edgar by taking the sociolinguistics context into account, and by allowing a kind of variable depth reasoning.

Each entry is stored as a hierarchical graph where each level corresponds to a particular kind of description: (i) *notions* are 'general semantic directions', (ii) *domains* mark the influence of the activity fields on the discourse, (iii) *conceptual views* are partial concepts, and (iv) *features* are pertinent properties of these concepts. Included in the static representation are *contextual conditions* and *semantic constraints*. Contextual conditions are a set of rules for initial salience attribution corresponding to very particular and well-known influences of some morpho-syntactic markers for *this* entry interpretation.

The edges between a parent node and its children nodes correspond to an *is-described-by* or *is-specialised-by* relationship. The semantic constraints are the edges between sibling nodes. They can be either neutral (co-validity of descriptions), reinforcement connectors (implication / increase of salience between two nodes), or inhibition connectors (opposition / decrease of salience).

Interpreting a polysemous word becomes, in our model, attributing salience rates to each node of the lexical structure. We use four symbolic rates: (i) *ignored*, meaning "not pertinent in this context", (ii) *valid*, meaning "possible but not very important", (iii) *salient*, meaning "important" and (iv) *negated*, which means "important but in a negative way". We use a salience propagation algorithm, initiated by the triggering of the contextual conditions. This algorithm is similar to the resolution of a system of non-linear recurring equations of k variables, which converges in k steps, if k is the number of descriptor nodes in the descriptive structure of a lexical entry. It terminates, in the worst case, in as many steps as there are nodes in the descriptive structure.

Implementation of the model

We have implemented this model in a pack of three software pieces: first, an engine, LightPeleas, managing the descriptive structures of a lexicon and applying the propagation algorithm on request. Then, a graphical editor, Melisande, to build and modify descriptive structures. And finally, Bard, a corpus parser that helps up to gather raw material for building the descriptive structures.

We implemented the engine LightPeleas as an ActiveX control. It publishes in the operating system 29 classes allowing the manipulation of any item from an entry to a single node or edge. Entries are stored on disk in a format we called PDL (Peleas Description Language). In order to help us use the interpretation given by LightPeleas for an entry, the output is a set of salient or negated conceptual views pondered by a "hint" between 0 and 1. It evaluates the plausibility of each interpretation (0 means 'perhaps', and 1 means 'rather sure')

So far, we used our system to build five descriptive structures (for 'mother', 'father', 'to devour', 'life' and 'little') and conducted a test with twenty-two sentences. The results we obtained were all sets of propositions with meaningful interpretations for the first or two first 'guesses'.

CONCLUSION

So far, we have delimited a kind of lexical ambiguities and their sociolinguistic cause: usage polysemy. We designed a model for its processing based on the observation of some translators' behaviour. This model is implemented and presents encouraging results so far

REFERENCES

Prince, V., and Bally-Ipsas, R. (1991). *Un algorithme pour le transfert de règles pragmatiques dans le processus complexe GLACE*. Rapport interne du LIMSI n° 91-17, Université Paris XI, Orsay.

Prince, V. (1994). A discrete approach based on logic simulating continuity in lexical semantics. In *Continuity in Linguistic Semantics*, Fuchs, C. and Victorri, B. (eds). *Linguisticae Investigationes Supplementa*, vol. 19. John Benjamins Publishing Company, Amsterdam/Philadelphia.

Pustejovsky, J. (1991). The Generative Lexicon. In *Computational Linguistics*, vol. 17(4).

Rastier, F. (1996). *Sémantique interprétative*, 2nd ed. Presses Universitaires de France, Paris.

Tanaka, K., and Umemura K. (1994). Construction of a Bilingual Dictionary Intermediated by a third Language. In *Proceedings of COLING-94*, Kyoto, Japan.

ESQIMO : Modeling Analogy with Topology

Erika Valencia, Jean-Louis Giavitto, Jean-Paul Sansonnet
LRI ura 410 CNRS, Bâtiment 490 Université Paris-Sud,
91405 Orsay Cedex, France
+33 (0)1 69 15 42 25
{erika,giavitto,jps}@lri.fr

ABSTRACT

ESQIMO is a computational model for analogy solving based on a *topological formalism of knowledge representation.* The source and the target analogs are represented as *simplicial complexes* and the analogy solving is modeled as a topological *deformation* of these complexes along a polygonal chain.

Key Words: Analogy solving, Algebraic topology, Simplicial complexes, IQ-tests.

TOPOLOGY FOR KNOWLEDGE REPRESENTATION

A representational formalism for analogy must allow the explicit expression of the features involved in similarity. M. Johnson (Johnson, 1987) argues that mental images are too close to perception and that logic approaches are too syntactic and arbitrary for representational purposes. He proposes to use a topological structure to represent and solve metaphors (which he considers to be the generalization of analogies (Lakoff and Johnson, 1980)).

Simplicial Complexes

Cognitive models use different models of space (Freska, 1997; Johnson, 1987) and the central question is in the choice of the basic spatial entities in a spatial representation of knowledge. We take here the elementary spatial entities to be *simplicial complexes.*

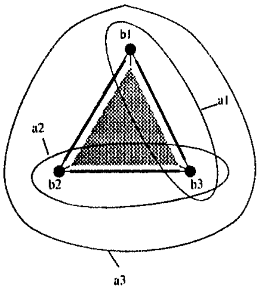
A simplicial complex is a couple (V, K) where V is a set of elements called vertices and K is a set of finite parts of V such that if $s \in K$, then all the parts $s' \subseteq s$ belongs also to K. The elements of K are called simplexes. The dimension of a simplex s is equal to $Card(s) - 1$. *All complexes with dimension* < 2 *are graphs.* Thus, simplicial complexes generalize semantic networks and allow the expression of hierarchies like in a relational graph.

The Q-Analysis

Atkin proposed the **Q-Analysis** (Atkin, 1981) to represent a binary relation λ between two sets with a simplicial complex. Let Λ be the incidence matrix of a binary relation $\lambda \subset A \times B$. Let $a \in A$, the set S_A of b_i such that $(a, b_i) \in \lambda$. All the elements b_i can be taken as vertices to represent the element a as a simplex. The whole matrix Λ can then be represented as a simplicial complex containing all the simplexes representing each element $a_i \in A$, we note it $K_A(B, \lambda)$ (see figure 1). Likewise, we can represent Λ^{-1} with the dual simplicial complex.

λ	a_1	a_2	a_3
b_1	1	0	1
b_2	0	1	1
b_3	1	1	1

(a) Incidence matrix of the binary relation λ



(b) Simplicial representation of λ

Figure 1: Representation of a binary relation

Extension of Q-Analysis

We extend the Q-Analysis to allow the representation of sets of predicates as a simplicial complex too. We can take a set of predicates $P = \{p_1, ..., p_n\}$ and represent the binary relation $\lambda \subset A \times P$ such that $(a_i, p_j) \in \lambda$ if $p_j(a_i)$ holds.

In this representational formalism, the same simplex is associated to elements of A that cannot be distinguished with the predicates of P available in the system. Moreover, two simplexes that have a smaller k-simplex in common are said to share a k-face. In terms of representation, it means that they have k features in common.

THE ESQIMO SYSTEM

A representational system is composed of a data structure and programs operating on it corresponding to reasoning tasks. We try now to model a simple analogy solving task using the representational structure proposed before : we chose the typical IQ-test problem. The system has to find an element D such that it completes a four-term analogy with three other given elements A, B and C.

The analogy is solved in 3 steps: find a relation R_{AB} between A and B, find the domain of C to apply R_{AB}, build $D = R_{AB}(C)$.

Representing the Problem

IQ-tests are given in terms of geometrical elements so that they can express many properties and stay simples. We took the following properties of *shape*: round, square, triangular; *color*: white, black; and *size*: big, small. According to our formalism, we build the complex $\mathcal{C}(\Omega)$ representing all the properties. The figures of the test are seen as relations between the set of properties and the elements of the figures, so we represent them also as sub-complexes $\mathcal{C}(A), \mathcal{C}(B)$ and $\mathcal{C}(C)$ of $\mathcal{C}(\Omega)$. Note that this formalization does *not* depend on the geometrical nature of the elements.

ESQIMO's Algorithm

To find R_{AB} we look for a transformation T_{AB} between $\mathcal{C}(A)$ and $\mathcal{C}(B)$ along a polygonal chain from $\mathcal{C}(A)$ to $\mathcal{C}(B)$ in $\mathcal{C}(\Omega)$. A *Polygonal chain* is a sequence of simplexes belonging to the same complex and where two successive simplexes have a non empty intersection. An elementary step linking σ_i to σ_{i+1} in a chain is then viewed as an elementary transformation $T_{\sigma_i,\sigma_{i+1}}$.

If there are several such chains, then there are several possible relations between A and B. To minimize the number of solutions, we give a higher priority to chains that are short and of higher dimension. Indeed, they correspond to transformations with less steps, and with more properties conserved at each step.

When T_{AB} is found, we use the same algorithm to determine T_{AC}. This second transformation is used to determine the domain of $\mathcal{C}(C)$ on which we can apply T_{AB}. Several strategies have been implemented (Valencia, 1997) considering only the things that changed between $\mathcal{C}(A)$ and $\mathcal{C}(C)$, or considering only the invariants between them, or some other hybrid methods. Finally, we can apply T_{AB} to this domain and build $\mathcal{C}(D)$. The translation of $\mathcal{C}(D)$ into a geometrical element of the universe is then easy.

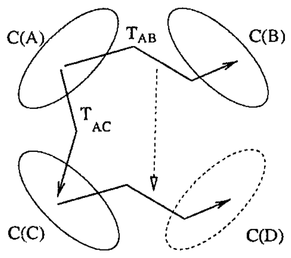


Figure 2: Analogy solving with ESQIMO

Remarks

The description of the properties of each figure in terms of predicates can be a problem for properties such as position. In that case, we can take only relative positions into account. Moreover, our transformations could be called 0−degree since they preserve the minimum of topological properties along a chain. The next step of this modelization would be to pair higher-order structures.

CONCLUSION

Different computational models have been developed to model analogy solving and are based on different representational structures. Among them, the ANALOGY system proposed by Evans (Evans, 1968) uses rules, the SME system proposed by Falkenhainer to illustrate Gentner's theory for analogy (Falkenhainer et al., 1989; Gentner, 1983) uses propositional structures, the ARCS system developed by Thagard and Holyoak to simultaneously satisfy the structural, semantic and pragmatic constraints uses neural networks and COPYCAT uses semantic networks with asynchronous parallelism. Like in SME, we focused on the structural constraint introduced by Gentner (Holyoak and Thagard, 1989) and we modeled the steps of analogy solving like in the ANALOGY system.

Our contribution lies in the search for a new representational structure (Valencia, 1997) that can be justified in terms of the naturality of a diagrammatic representation (Glasgow et al., 1995). Like in the COPYCAT project, we are concerned with the mechanisms of enrichment of a representation through analogy and our formalism can be seen as an intermediate structure between a symbolic and an analogical approach.

ESQIMO has been implemented in the ML programming language, the various strategies experimented and some solving examples are given in details at `http://www.lri.fr/~erika`.

REFERENCES

Atkin, R. H. (1981). *Multidimensional Man*. Penguin.

Evans, T. G. (1968). A program for the solution of a class of geometric analogy intelligence-test questions. In *Semantic Information Processing*, chapter 5, pages 271–353. The MIT Press.

Falkenhainer, B., Forbus, K. D., and Gentner, D. (1989). The structure-mapping engine. *Artificial Intelligence*, 41(1):1–63.

Freska, C. (1997). Spatial and temporal structures in cognitive processes. In Freska, C., Jantzen, M., and Valk, R., editors, *Foundations of Computer Science*, volume 1337 of *LNCS*. Springer-Verlag.

Gentner, D. (1983). Structure-mapping a theoretical framework for analogy. *Cognitive Science*, 7(2):155–170.

Glasgow, J., Narayanan, N. H., and Chandrasekaran, B. (1995). *Diagrammatic reasoning : Cognitive and Computational Perspectives*. AAAI Press/MIT Press.

Holyoak, K. J. and Thagard, P. (1989). Analogical mapping by constraint satisfaction. *Cognitive Science*, 13(3):295–355.

Johnson, M. (1987). *The Body in the Mind*. The University of Chicago Press.

Lakoff, G. and Johnson, M. (1980). *Metaphors we live by*. The University of Chicago Press.

Valencia, E. (1997). Un modèle topologique pour le raisonnement diagrammatique. Rapport pour le DEA Sciences Cognitives, LIMSI. See also `http://www.lri.fr/~erika`.

Author Index