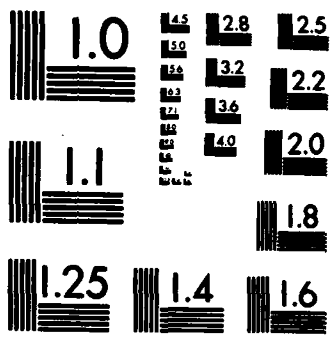
AD-A150 867 RESEARCH NEEDS FOR ARTIFICIAL INTELLIGENCE APPLICATIONS 1/1
IN SUPPORT OF C3 (..(U) NAVAL OCEAN SYSTEMS CENTER SAN
DIEGO CA R A DILLARD DEC 84 NOSC/TR-1009

UNCLASSIFIED F/G 17/2 NL

END
FILMED
DTIC


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0

TR 1009

AD-A150 867

(2)

TR 1009

Technical Report 1009

December 1984

RESEARCH NEEDS FOR ARTIFICIAL INTELLIGENCE APPLICATIONS IN SUPPORT OF C^3

R. A. Dillard
Artificial Intelligence Branch
Code 444

Prepared for
Naval Electronic Systems Command
Code 613

Naval Ocean Systems Center
San Diego, California 92152-5000

Approved for public release; distribution unlimited.

NAVAL OCEAN SYSTEMS CENTER SAN DIEGO, CA 92152

AN ACTIVITY OF THE NAVAL MATERIAL COMMAND

F. M. PESTORIUS, CAPT, USN
Commander

R.M. HILLYER
Technical Director

ADMINISTRATIVE INFORMATION

This task was performed for Naval Electronic Systems Command, Code 613, Washington, D.C. 20363, under program element 61153N, subproject XR0140801. Work was performed by the Artificial Intelligence Branch, NAVOCEANSYSCEN, San Diego, CA 92152-5000.

Released by
G. R. Allgaier, Head
Artificial Intelligence
Branch

Under authority of
W. T. Rasmussen, Head
Advanced C^2 Technologies
Division

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY 2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited
4. PERFORMING ORGANIZATION REPORT NUMBER(S) NOSC TR 1009	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Artificial Intelligence Branch 6b. OFFICE SYMBOL (if applicable) Code 444	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State and ZIP Code) Naval Ocean Systems Center San Diego, CA 92152-5000	7b. ADDRESS (City, State and ZIP Code)
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Naval Electronic Systems Command 8b. OFFICE SYMBOL (if applicable) Code 613	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State and ZIP Code) Washington, DC 20363	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. 61153N PROJECT NO. XR0140801 TASK NO. WORK UNIT NO. 444-CC25

11. TITLE (include Security Classification)

RESEARCH NEEDS FOR ARTIFICIAL INTELLIGENCE APPLICATIONS IN SUPPORT OF C^3

12. PERSONAL AUTHOR(S)

Robin A. Dillard

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Interim	FROM ______ TO ______	December 1984	24

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Artificial intelligence Command control Data fusion

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Earlier investigations of Navy data fusion problems and techniques are extended into three other areas. The first investigations, from a single site and platform, are extended to a global network of systems. The problem of sharing information among subsystems of a command control system is expanded to that of sharing information of mutual interest with other units and battlegroups. The next investigation concerns a reconstruction and post-analysis system. The reconstruction process is simply data fusion after all data are in. After reconstruction, artificial intelligence (AI) techniques may be used to interpret and help analyze the event records. The last topic concerns information storage and retrieval in novel mediums. Throughout these discussions, the main emphasis is on the application of AI tools. Additional keywords: maps, pictures, geometry, information processing, communications, networks.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT [X] UNCLASSIFIED/UNLIMITED [ ] SAME AS RPT [ ] DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION	Agency Accession DN213 029
22a. NAME OF RESPONSIBLE INDIVIDUAL Robin Dillard	22b. TELEPHONE (include Area Code) 619-225-7778	22c. OFFICE SYMBOL Code 444

DD FORM 1473, 84 JAN

83 APR EDITION MAY BE USED UNTIL EXHAUSTED
ALL OTHER EDITIONS ARE OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

DD FORM 1473, 84 JAN

CONTENTS

ILLUSTRATIONS

Figure

INTRODUCTION

Part of our work for several years has been to identify Navy data fusion problems that cannot be handled by present techniques, and to experiment with new and postulated techniques. The techniques have included some for natural language processing, tactical inferencing, problem solving, and database updating. Our primary interest has been complementary interaction of the various techniques in a data fusion system and, more recently, automated sharing of knowledge with other subsystems in a command control system. This year, we have extended our earlier investigations of data fusion into three other areas.

The first investigations from a single site and platform are extended to a global network of systems. The problem of sharing information among subsystems of a command control system is expanded to that of sharing information of mutual interest with other units and battlegroups. The next investigation concerns a reconstruction and post-analysis system. The reconstruction process is simply data fusion after all data are in. After reconstruction, artificial intelligence (AI) techniques may be used to interpret, and help analyze the event records. The last topic concerns information storage and retrieval in novel mediums. We previously dealt only with information in computer memory, but much of the needed information derives from photographs, maps, and other mediums. Throughout these discussions, the main emphasis is on the application of AI tools.

This brief survey of problems and the approaches to their solution is far from exhaustive, but it points to many needs that should be addressed in research and exploratory development projects. We encourage other researchers working in AI, and new technology areas, to consider the problems discussed herein, and we invite comments and ideas.

NETWORK CONSIDERATIONS

Before discussing techniques for efficiently sharing knowledge and maintaining consistency among cooperating information processing systems, we need to outline a network in which these techniques could be employed. The terminology used to describe the components was chosen for convenience and is arbitrary.

Figure 1 shows clusters of command control (C^2) information processing systems tied together via a regional network, e.g., the HP intratask-force communication network proposed by Baker (Reference 1). (Communication with other clusters is shown occurring through a global communication network, most likely using satellites.) Each such information processing system, which we shall refer to as a "unit," consists of subsystems for data fusion, planning, natural language processing, updating, comparing, communicating, simulating, etc. Each subsystem has a private memory and shares the common database.

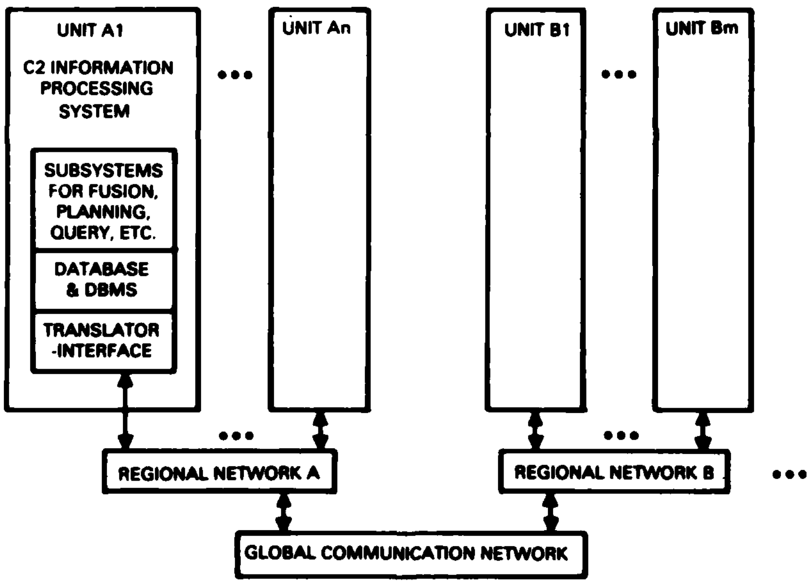


Figure 1. Command control information processing systems communicate regionally and globally via interfaces with communication networks.

Any unit able to transmit to, and receive from, all units can be designated the regional processing unit (RPU), although this requirement can be loosened if communication failures are extensive. (The RPU of a carrier group would most conveniently be aboard the carrier, provided its transmissions do not make it more vulnerable to detection and homing.) Each unit sends new reports to the RPU on contact descriptions, and behavior, and on activities, and plans of possible interest or concern to other units. The RPU may filter the reports and address copies to units whose area of interest overlaps with the location in the report or will overlap based on projections. Optionally, some units might directly exchange information on their areas of overlap or concern. Similarly, the RPU can send filtered reports from its own sensors, patrol aircraft, and RPUs in other regions.

A COMMON LANGUAGE

We assume that the subsystems of a unit can communicate conceptual information of mutual interest among themselves in a generic form and, similarly, that a generic form common to the cluster can be used in communicating data among units (Figure 1). A generic form of communication is, in addition to interchanging NTDS tracks, administrative messages, Rainform messages, etc. This form of communication would be for reports of events, activities, plans, and other conceptual information. Ultimately, however, this form of communication could replace much of the Rainform message traffic. It would also include the results of the unit's correlations and its attempts to recognize platforms and infer intent.

In Reference 2, the generic form proposed was that of object-attribute-value tuples, where the value can be a number, a word (e.g., destroyer, CVN-70), a string (e.g., "This is a string."), or a vector. Essentially, every expert system is able to use this structure, and various representative events, complex concepts, and missions were satisfactorily represented with these structures. In experimenting with these representations, we found it useful to distinguish between "actual" events (those which occurred or possibly occurred) and "virtual" events (those which were planned or were expected to occur) by using the prefix $ for an actual event, and the prefix V$ for a virtual event. Sometimes, it is also useful to prefix an ongoing event with O$. The character $ is reserved in some systems, in which case & or some other symbol should be used.

Even though this object-attribute-value structure is common to the various expert systems, each has its own particular version. The following examples are equivalent input statements in several AI languages.

ROSIE Representation

```
Assert $ATTACK5 is an $ATTACK
and let SQUADRON-VS22 be the attacker of $ATTACK5
and SUBCONTACT-SA5 be the victim of $ATTACK5
and MK-46 be the weapon of $ATTACK5
and 291320 be the time of $ATTACK5
and <27.923, 50.035> be the lat-lon of $ATTACK5
and DESTROYED be the result of $ATTACK5.
```

ROSIE (Rule-Oriented System for Implementing Expertise) was developed by the Rand Corporation (Reference 3). The first statement is what is known as an ISA, or "is a," relation or attribute. Different systems express ISA in different ways, and many have varying degrees of inheritance mechanisms attached. The remaining statements bind values to the various attributes of the object $ATTACK5.

STAMMER2 Representation

```
(MESSAGE
 ($ATTACK $ATTACK5)
 (ATTACKER $ATTACK5 SQUADRON-VS22)
 (VICTIM $ATTACK5 SUBCONTACT-SA5)
 (WEAPON $ATTACK5 MK-46)
 (TIME $ATTACK5 291320)
 (LAT-LON $ATTACK5 (27.923 50.035))
 (RESULT $ATTACK5 DESTROYED))
```

STAMMER2 (Version 2 of System for Tactical Assessment of Multisource Messages, Even Radar) is a small, experimental rule-based system developed at NOSC for the purpose of performing tactical situation assessment (Reference 4). STAMMER's main input is formatted data, which it converts to its own system syntax, but it also accepts a message such as the one shown below. The first assertion is the ISA statement, and the rest are in the form "(attribute object value)."

PROLOG Representation

```
&attack(&attack5).
attacker(&attack5, squadron_VS22).
victim(&attack5, subcontact_SA5).
weapon(&attack5, mk_46).
time(&attack5,291320).
lat_lon(&attack5, [27.923, 50.035]).
result(&attack5, destroyed).
```

PROLOG (PROgramming LOGic) is a popular language first developed in France (Reference 5). The style is that of first-order, predicate-calculus terms. The first term is the ISA statement. Since $ is a reserved character in PROLOG, we use the prefix &.

FRL Representation

```
(FASSERT &ATTACK5 (AIO ($VALUE (&ATTACK)))
                  (ATTACKER($VALUE (SQUADRON-VS22)))
                  (VICTIM ($VALUE (SUBCONTACT-SA5)))
                  (WEAPON ($VALUE (MK-46)))
                  (TIME ($VALUE (291320)))
                  (LAT-LON ($VALUE (27.923) (50.035)))
                  (RESULT ($VALUE (DESTROYED))))

(FASSERT &ATTACK (INSTANCES ($VALUE (&ATTACK5))))
```

FRL (Frame Representation Language) was developed at the AI Laboratory of MIT (Reference 6). Several systems have been built in FRL, including some military planning systems. AIO stands for "An Instance Of," an ISA relation. The ISA relation is expressed again to facilitate the reasoning process.

ROSS Representation

```
(ASK $ATTACK CREATE INSTANCE $ATTACK5
   WITH ATTACKER SQUADRON-VS22
        VICTIM SUBCONTACT-SA5
        WEAPON MK-46
        TIME 291320
        LAT-LON (27.923 50.035)
        RESULT DESTROYED)
```

ROSS is an object-based, message-passing language developed at Rand for constructing simulation and, in particular, battle simulations (Reference 7). Creating an instance gives it an ISA relationship.

OPS5 Representation

```
(make $attack  label $attack5
               attacker squadron-VS22
               victim subcontact-SA5
               weapon MK-46
               time 291320
               lat-lon 27.923 50.035
               result destroyed)
```

OPS5 is a member of the OPS family of rule-based systems developed at Carnegie-Mellon University (Reference 8). (Historically, the name OPS derives from Official Production System.) This representation is effectively equivalent to the others, since the first statement implies an ISA relationship. (The word "name" frequently is used where we have used the word "label." We could have used "isa" here.) OPS5 has the limitation that an instance can have only one vector attribute; so while it can talk to other systems, it cannot always understand them.

OPS83 Representation

```
make ($attack label=$attack5;
              attacker=squadron_VS22;
              victim=subcontact_SA5;
              weapon=MK_46;
              time=291320;
              lat_lon[1]=27.923;
              lat_lon[2]=50.035;
              result=destroyed;
```

OPS83 is the most recent member of the OPS family of production system languages (Reference 9). It was developed for use aboard the carrier USS Carl Vinson (CVN-70), in a testbed environment. Position is shown as an "array" above, analogous to the other representations, but latitude and longitude probably will be separate attributes in OPS83 tactical reports. Also, attribute values will probably be pointers to the actual values. OPS83 is written in language C and will run on a VAX-11/780 on the Vinson.

ROSIE Alternative:

Assert SQUADRON VS22 did destroy SUBCONTACT-SA5
about 291320 at <27.923, 50.035> with MK-46.

ROSIE is capable of more complex representations, such as the one above, but since translation into a common syntax would be difficult, we should avoid these English-like forms for data that should be shared. Also, retrieval of values would be more difficult using these English-like forms. In ROSIE, retrieval is sometimes facilitated by expressing an assertion in two ways: "trackA is a track of platformA" and "platformA is a platform of trackA."

The representation would include a "free-form assertion" capability which links a natural language comment to an event, track, or other object. The latter should not duplicate information in the constrained representations. (Message composers have tended to repeat the formatted information with narrative comments, but future systems will be able to generate natural language descriptions from formatted data.) Since the free-form assertions would be usable only by query systems and would not be machine understandable for fusion or other processes, they should be confined to concepts not expressible in the system's vocabulary.

A convention would have to be agreed upon for how to represent each kind of event, mission, etc. Also, each system would either have to number its own instances (and relate them to the source), or use a common label, such as $ATTACK-SA5, where SA is, in this example, the two-letter code for Saratoga, the originator of the report. (This is a method currently used for labeling submarines.) When two sources report the same event, these two reports would be reduced to one event record, so a simple rule of using the label provided is unsatisfactory. Most systems have the capability of numbering instances (ROSIE would use the label $ATTACK #5); for historical reasons, these systems would also need to record the reported label or link the report to its source in some other way.

To minimize the number of bits transmitted, no unnecessary information should be sent. The problem of deciding what information to transmit is discussed later, but we note here that information that can be generated with inheritance mechanisms need not be sent. For example, if we transmit the name of a surface ship, we should not transmit its type, class, or the fact that it is a surface ship. Unless there are exceptions, we should not transmit data common to its class, such as the sensors and weapons it carries. Every unit should have the same inheritance rules, although their implementation among the different subsystems may vary greatly. For efficiency, the information should be transmitted in "frames" rather than as lists of assertions, although some frames will consist of a single assertion. The representation

```
//ATTACK/LABEL $ATTACK-SA5/ATTACKER SQUADRON-VS22/VICTIM
SUB-CONTACT-SA5/WEAPON MK-46/TIME 291320/LAT-LON 27.923
50.035/RESULT DESTROYED//
```

can be transmitted via standard links; for instance, in a manner similar to that mentioned on page 8 for transmitting "display summaries." Most of the routine tactical data can be communicated in existing formats, but a form such as this could be used for data which cannot be communicated in existing formats, or which is requested by a remote unit.

Another simple way to reduce the number of bits is to code the more common objects, attributes, and values before transmission and decode them at the receiver. For example, SQUADRON-VS22 could be sent as "SQ-VS22" and the attribute LAT-LON could be coded as "LL."

HOW INCONSISTENCIES OCCUR

Currently, many inconsistencies occur from gridlock errors, although this should be much less of a problem when NAVSTAR is installed on all platforms. Other inconsistencies occur from sensor inaccuracies, time-late messages, enemy deceptions, human error, etc. Correlation of bearing data may result in dual designations of a contact for some units and not for others.

Each unit may generate numerous conclusions based on its own and others' reports, and some of these conclusions, especially those concerning a description of a contact, may be incorporated into a new report of the contact. As a result, errors may be propagated. Also, units using different inference rules may combine similar event reports into the report of a single event, while others may conclude they are distinct events. Since rulesets can be modified by the user, variations in conclusions can be drawn from the same data. Figure 2 gives an example of how different conclusions might occur.

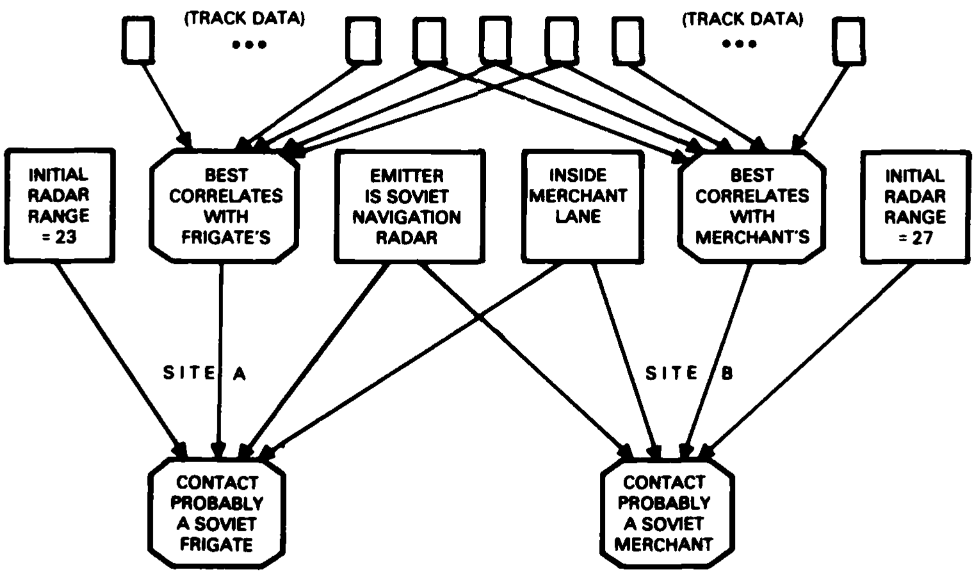


Figure 2. An example of different units reaching different conclusions. (Rectangles represent observed data; other blocks represent conclusions.)

DETECTING AND RESOLVING DATA INCONSISTENCIES

In this section, we will address problems that result from reasoning with different data and from differences in individual reasoning processes. A method of transferring essential information using compression and abbreviated representation is proposed by Grant (Reference 10). The information would be transferred in the form of "command summary displays," and would include both graphic and alphanumeric information. An example of how this information can be transmitted in a RAINFORM GOLD message is given in Reference 10. The displaying of compound threats may require, in addition to geometrical representation, a conceptual representation that expresses the various possible combinations and their threat, subject to the unknown facts. Generally, the summary should include merchant tracks. For example, in the problem in Reference 11, an attack unit is receiving target data from the RPU, based on reports from units that they have overlapped surveillance areas: the final targeting phase may call for direct transmissions to the attack unit from a unit holding the target's track, but it is important that pictures of merchant traffic in the area be consistent among the units providing such data.

The basic approach we suggest is to detect and resolve the inconsistencies among the databases of different units by comparing summary data and exchanging observed data pertinent to the inconsistencies. The conflicts should probably be detected at the RPU level and the units concerned informed. However, detection of conflicts at the unit level (by examining overlapping areas of summary displays from other units) should also be satisfactory. This same approach should apply to exchanging information among the regional networks via the global communication network, although data inconsistencies are much less likely because of the minimal overlapped area. Much of the information exchanged will be used for coordinating operations in overlapping regions. Some kinds of information should be communicated to all units, as in an attack when it is the first strike. Also, information about an attack where the target is destroyed is of interest to all units needing a current platform file.

One conclusion is that each unit should have mechanisms (probably at the subsystem level) for distinguishing between inferred and observed facts. The display summary would show some of each kind of facts. Again, however, exchanges to resolve conflicts should involve observed data.

Some types of inconsistencies can be minimized by automatically sharing observed data of mutual interest in addition to summary information. This approach is probably much less efficient than exchanging pertinent observed data only after conflicts in summary displays are detected. Which approach is better would have to be determined experimentally and would depend on EMCON state and other factors. We are addressing primarily the latter approach, where the conflict or inconsistency determines what data should be exchanged. (Again, exchange of NTDS and certain other data is not affected.) This data generally should not include anyting so detailed as buoy patterns and raw signal data, but should reveal (in this example) whether the class confidence was based on sonar data or on a correlation with an earlier track. Similarly, resolving a conflict in surface ship class will often require knowledge of emissions and lines of bearing, but sending the emitter name is obviously more practical than sending a description of the emission.

In general, only some of the attributes of an event need to be communicated. In the originator's database, attributes of $attack5 not shown in the examples in the section, A Common Language, may include the actor, "patrol15," which, itself, has many local-only attributes (the attacker "squadron-VS22" is of more general interest) and may include links to other events, such as the plan, "V$attack6," and the support, "$locate&ID32." Links to supporting events and plans are useful locally for determining when to retire event reports into the "history file." Later, these links are good for use by a reconstruction and post-analysis system.

Each unit should resolve or live with its own conflicts among reports and may communicate with other units to investigate. Priorities should be decided as to what conflicts to resolve and in what order to resolve them. For example, a friend/foe or combatant/noncombatant conflict should be investigated immediately, while a destroyer/frigate conflict might not matter. Similarly, a target-destroyed/target-active conflict in an event description is more important than having different times or positions.

RECONSTRUCTION AND POST-ANALYSIS

THE NEED FOR GOOD RECORDS

Good records are needed for assessing fleet performance and readiness. Also, in determining the most probable enemy reaction to an operation under consideration, historical records of the enemy's strategies and their reactions during earlier operations of a similar kind can be valuable. The records can also be useful in determining enemy capabilities and estimating the probable outcome and losses of an operation.

Human documentation is an extremely slow process, and the result is not subject to query. Sophisticated knowledge-based techniques are needed to select pertinent data, reconstruct events from them, and organize the fused event records in a representation suitable for querying by other systems and humans. Knowledge-based system techniques are also needed to analyze the data and provide useful interpretations. A system for reconstructing and analyzing the flow of events of naval exercises and operations should be developed in conjunction with other systems. One early benefit of such a system would be its use in evaluating other systems as they are developed.

RECONSTRUCTION FROM HISTORICAL FILES

Much of the available data needed for good records will at some time be stored in the database of the C^2 information processing system described on page 2. The "history file" created by the data fusion and planning subsystems will record events of tactical exercises and engagements in a manner useful in event reconstruction and evaluation. Reconstruction of events is a natural extension of the database updating features of the data fusion system, although here we have the advantage of having all data available at once, thus eliminating the need for backup and correction mechanisms. Also, data which could not be communicated from remote units during times of EMCON can be incorporated for overlapping geographical regions, using the history files of the respective units. The reconstruction process can use many of the computational algorithms of the data fusion system.

The reconstruction process should have access to:

- all files, position/movement data, and tactical messages available to the planning system, the data fusion system, and the decision maker;
- pertinent plans in a machine readable form, attainable via a planning support system;
- machine-fused assessments and conclusions;
- decision-maker actions;
- post exercise/operation information (e.g., damage assessments and the commanders' corrections of, and additions to, the machine reconstructed documentation).

Part of the reconstruction process requires detecting enemy deceptions not identified by the data fusion system and correcting event records accordingly. The reconstruction system can be used to test the data fusion system by employing a playback feature: time sequential fused "pictures" based on all data can synchronously be presented with the fused pictures which were available during the (simulated) exercise or operation.

INTERPRETATION TECHNIQUES

Techniques are needed which will exploit the historical records to provide automated analytical assistance in, for example:

- evaluating data fusion system conclusions and assessments, e.g., determining inconsistencies and inadequacies of data fusion rules;
- evaluating sensor, emission, and weapon control strategies (both by individual ships and by coordinated task force);
- evaluating human decision processes (operators, coordinators, commanders);
- determining combat system reaction times and errors;
- evaluating the effectiveness of battlegroup positions against enemy threats and predicting battle outcomes;
- determining the state of enemy knowledge from their actions;
- finding indicators of enemy intention and using records of their subsequent actions;
- updating a priori intelligence libraries and tactical inference and doctrine rulesets;
- developing and refining mathematical models and simulations of a C^3 system, e.g., developing analytical forms for representing decision processes and their impacts.

A first step to automation is a query system interfaced with a database. At this stage, statistical processes would probably not be directly interfaced. The next step should be to automate the users' repetitive processes. (Users generally would be military historians and planners.) Control would remain with the user, who would call the appropriate ruleset or other embodiment of the procedure to select, retrieve, and operate on particular data. The automated analyses would largely be long-term statistical analyses and a comparison of the most recent exercise/operation outcomes, with the statistical outcomes of similar earlier activities. As the users become more familiar with the programming process, they can implement more difficult procedures, such as evaluating human decision processes and finding indicators of enemy intention. Gradually, the experts' knowledge would be built into the system and the analysis would become increasingly automated.

In Reference 12, long, but simple formulas are used to model historical conflict and to project future outcomes. While these models deal primarily with armies, similar models could be developed for navy operations. The development of the formulas could be largely automated by integrating techniques such as those of BACON (References 13 and 5). BACON is a data-driven method of discovering simple laws relating to real-valued variables.

We plan to begin experiments with reconstruction and post-analysis techniques. These will likely be performed in ROSIE, since its architecture is well suited to this process. While no current implementation of ROSIE has the memory and speed required for operational use, ROSIE can be used for experimental work in this area. The statistical and geometrical computations can be implemented as function rulesets; the retrieval and manipulation processes can be implemented as procedural rulesets.

MAPS, PICTURES, AND GEOMETRY

THE PROBLEM

The assumption, so far, has been that all tactical data, whether NTDS tracks, conceptual data, or other kinds, are stored as bits in computer memory, and that the data fusion system evaluates its rule conditions, and the query system answers questions based on this data. In this section we will look at situations where this is impractical and investigate the alternative method of finding answers in an "external" subsystem (Figure 3) which is able to access data in other mediums. These other mediums include photographs, geographic maps, ocean flow maps, wind-field maps, radar scope "frames," synthetic-aperture radar maps, and optical discs. (See References 14-16 for discussions of remote sensing and radar imaging.) Examples of conditions/questions are:

- Is the contact in shallow waters?
- Is the contact following coastal cliffs?
- Is the contact in territorial waters of some nation?
- Is the contact near land?
- Is an island in the way of the hypothesized path?
- Is the motion that of a carrier turning into the wind?
- Is the contact in a storm?
- Is the contact avoiding a storm? An oil spill? Drift ice?
- Are the motions those of fishing boats? (e.g., pairs pulling nets)
- What, if any, is the formation of the group?

Also, problems to be solved in planning systems include avoidance of patrols, storms, waters too shallow, etc.

RELATED EFFORTS

Davis presents a computational model of memory for spatial relations, called MERCATOR (References 17 and 18). Figure 4 is a more general version of MERCATOR, since we are considering a variety of inputs. In Davis's example, the "scene description" is assumed to be derived from a robot's vision, and MERCATOR's objective is to collect, through wandering and looking around, all the information from a "world model" into its cognitive map. The assimilator finds correspondences between a scene description and the knowledge base (the cognitive map) and adds the new information from the scene description into the knowledge base. Knowledge of the robot's motion (and, in our applications, the camera's location and direction) is used in learning a large scale area from a sequence of small scale views of the area. MERCATOR's representation scheme, which is in two-dimensional space, primarily uses straight line segments, but also uses a number of other elements. Its spatial reasoning is

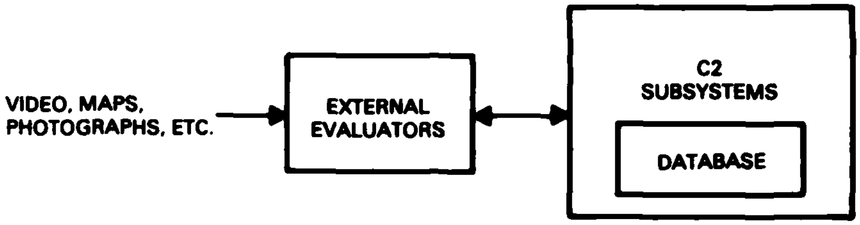


Figure 3. External evaluation of conditions and questions.

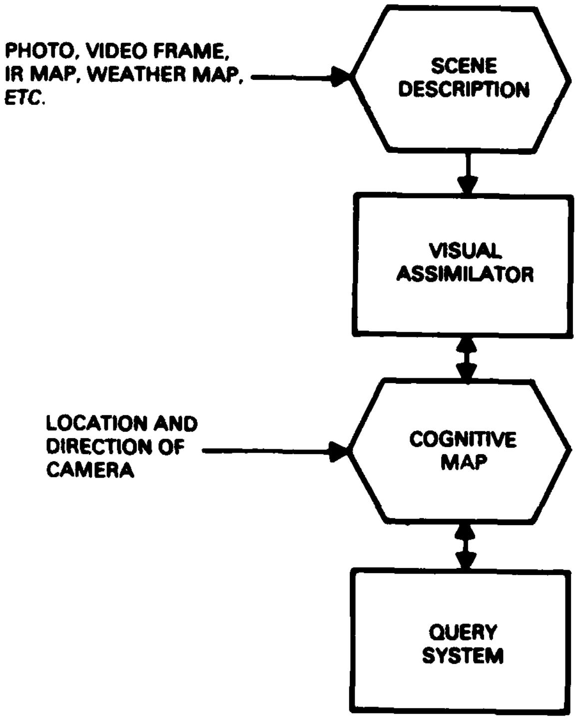


Figure 4. Generalized MERCATOR structure.

able to operate on incomplete and inexact information represented in the knowledge base.

MERCATOR adopted many of the basic features of another system developed at Yale — SPAM (SPAtial Module) — by Drew McDermott (Reference 19). (This SPAM should not be confused with the SPAM discussed below, developed by David McKeown and John McDermott.) The input to SPAM is a sequence of geographic and physical facts, while the input to MERCATOR is vision. SPAM has a number of features not incorporated into MERCATOR, while MERCATOR has some original representation techniques which avoid some of the problems of SPAM.

Expert systems can be involved in the use of photographs and maps in several ways. One way is to use rule-based systems to control the image processing and the interpretation of results. A system which does this is SPAM, a system for semi-automatic photo-interpretation of high resolution aerial photographs (Reference 20). (This system is not the SPAM described above.) A major component of SPAM is MAPS (Map Assisted Photo-interpretation System), a database of about fifty high-resolution aerial photographs, digitized maps and other cartographic products. The function of MAPS is to tie database feature descriptions to a geodetic coordinate system and use image-to-map correspondence to predict their location and appearance in the aerial photography.

ACRONYM is a goal-directed, model-based image understanding system intended to deal with key problems of interpreting scenes (Reference 21). It was developed at Stanford University under the DARPA Image Understanding Project. It has a three-dimensional model representation and a rule-based planning system. Some of the results of this project are used in a photo interpretation system described in Reference 22. This system attempts to identify interesting objects by matching shapes extracted from digitized images to shapes generated by geometric analysis of three-dimensional object models, and from information describing illumination conditions, etc.

While the subject of temporal activities is addressed by Bullock (Reference 22), none of the systems described above deals with temporal changes. Some researchers are addressing the problem of reasoning both in space and time (e.g., References 23 and 24), but their work deals mainly with events rather than scenes with moving objects or boundaries. Some work which does involve moving objects is described by Tsuji (Reference 25). A dynamic scene analyzer operates on motion picture film to separate moving objects from the background and analyze their motion patterns in dynamic line images.

Reference 26 describes a representation for image curves. The representation could be adapted to the problem of representing tracks in a minimum number of bits. The representation basically consists of a list of points in the plane with tangent direction and signed curvature specified at each point. The representation may also be used to describe coastlines. At a three-dimensional level, geographic "objects" such as landmarks and ocean sea bottom might be described using a decomposition technique discussed in Reference 27. An object is decomposed into symmetrical components which are meaningful to a human being.

GEOGRAPHIC COMPUTATIONS

First, consider the example on page 13, "Is the contact near land?" (This is a condition which helps recognize certain kinds of ships in a ship classification problem.) In practice, "near" would have different distance values for different ship types. However, all the distances would be great enough that land boundaries could be crudely described with long-line segments, and even take into account where the ports are located. The active memory of the data fusion and query systems needs to contain only the nearest land boundaries, and not even these if far out to sea. There is probably no need to call on a specialized external system to answer this kind of question. Similarly, the question "Is an island in the way of the hypothesized path?" is easily evaluated by modeling the islands as polygons (as done in STAMMER2). We can conceive, though, that an external system can be designated to do this more efficiently, perhaps in parallel.

Next, consider "Is the contact following coastal cliffs?" and "Is the contact in territorial waters of some nation?" Again, we can evaluate these conditions in the data fusion system (or query system), but at the expense of massive bit maps and extensive calculations. Alternatively, we might use a more efficient representation (e.g., an image-curve representation as shown in Reference 26) of the track and coastline, but computational evaluation of the conditions could be much more difficult or even impossible. Either way, we would never want to evaluate such conditions unless there was good reason to do so, and usually a human operator could observe this condition and enter it into the database (when asked by the system) much more easily. Still, if a condition in this category is found to be occasionally important, the capability of automatically evaluating it should be *implemented*. Perhaps the best approach at this time is to externally do the same thing that would be done inside the data fusion or query system; that is, store the map (probably as line segments) and perform the calculations, while the data fusion system continues other processing until the answer for that suspended rule is returned.

So far we have only considered evaluation involving fixed geographic features. Another relatively fixed evaluation feature is depth: "Is the contact in shallow waters?" The sea bottom varies with time in coastal areas, but it is still possible to keep, for the current region of concern, a fairly updated record in terms of boundaries at different depths. We envision that "shallow" is given a value in any particular application; the boundaries for that depth are the only ones needed in computations. Even then, a tremendous amount of data is needed, and we would not want to do this within the data fusion or query system. Also, marginally, we might judge this problem as better handled with some other medium and read the depth (perhaps as an intensity) at that location only.

The difficult question is how to deal with pictures and charts of scenes with moving objects and changing patterns. We would like to evaluate conditions through direct interaction with the medium of storage, which may be possible in certain cases some day. It appears, however, that the information must be digitized or converted to computer bits before reasoning can be performed. The best policy is probably to process only on demand, since much of the information may never be needed by an automated system. Also, the picture often needs to be processed only to determine a single feature.

At this point, some of the existing techniques show promise for handling a few of the problems (e.g., pictures showing wakes of ships might be processed with systems having knowledge of shadow effects). Clearly, though, most of the problems will require at least several years of research.

CONCLUSIONS

A number of diverse ideas have been presented here. They have in common the fact that they are artificial intelligence techniques employed in C^2 systems, with primary emphasis on representation schemes. The investigations are only partially completed, but will continue in FY 85. Plans for experimental implementations include (a) techniques for selecting data to be communicated within and among battle groups, and (b) techniques for reconstructing event sequences from a history file and using these reconstructions to evaluate earlier data fusion system conclusions. A new issue to be addressed is how C^2 systems, and subsystems can "grow together," gradually extending their capabilities of representing the complex concepts they must communicate among themselves.

REFERENCES

1. Baker, DJ, Wieselthier, J, and Ephremides, A, The HF Intratask Force Communications Network Design Study, Proc. 4th MIT/ONR Workshop, Vol III, 7-29, 1981.

2. Dillard, RA, Representation of Tactical Knowledge Shared by Expert Systems, NOSC TD 632, 1 October 1983.

3. Waterman, DA, Anderson, RH, Hayes-Roth, F, Klahr, P, and Martins, G, Design of a Rule-Oriented System for Implementing Expertise, Report N-1158-1-ARPA, The Rand Corporation, May 1979.

4. McCall, DC, Morris, PH, Kibler, DF, and Bechtel, RJ, STAMMER2: A Production System for Tactical Situation Assessment, NOSC TD 298, October 1979.

5. Cohen, PR, and Feigenbaum, EA, The Handbook of Artificial Intelligence, Vol. III, HeurisTech Press, Stanford, CA, 1982.

6. Roberts, RB, and Goldstein, IP, The FRL Manual, Memo 409, MIT, September 1977.

7. McArthur, D, and Klahr, P, The ROSS Language Manual, Note N-1854-AF, The Rand Corporation, September 1982.

8. Forgy, CL, OPS5 User's Manual, Department of Computer Science, Carnegie-Mellon University, Pittsburgh, 1981.

9. Forgy, CL, Overview of OPS83, Department of Computer Science, Carnegie-Mellon University, Pittsburgh, 1983.

10. Grant, CJ, Coordination of Battle Group Warfare Commanders through Summary Display Transfers, Proc. 6th MIT/ONR Workshop on C^3 Systems, 129-135, 1983.

11. Corman, DE, Measurement of Inter-Nodal Database Commonality, Proc. 4th MIT/ONR Workshop, Vol. III, 73-91, 1981.

12. Dupuy, TN, Numbers, Predictions and War: Using History to Evaluate Combat Factors and Predict the Outcomes of Battles, Bobbs-Merrill Co., Inc., Indianapolis, 1979.

13. Langley, PW, Rediscovering Physics with BACON.3, IJCAI-6, Vol. 1, 505-507, 1977.

14. Hibbs, AR, and Wilson, WS, Satellites Map the Oceans, IEEE Spectrum, 46-53, October 1983.

15. Holter, MR, Remote Sensing: The Next 50 Years, IEEE Trans. Aerospace and Electronic Systems, Centennial Issue, Volume AES-20, Number 4, July 1984.

16. Ausherman, DA, et al, Developments in Radar Imaging, IEEE Trans. Aerospace and Electronic Systems, Centennial Issue, Volume AES-20, Number 4, July 1984.

17. Davis, E, The MERCATOR Representation of Spatial Knowledge, IJCAI--8, vol. 1, 295-301, 1983.

18. Davis, E, Representing and Acquiring Geographic Knowledge, Yale/UCSD/RR#292, Yale University, January 1984.

19. McDermott, D, and Davis, E, Planning and Executing Routes through Uncertain Territory, Artificial Intelligence, 1983.

20. McKeown Jr., DM, and McDermott, J, Toward Expert Systems for Photo Interpretation, Proc. Trends & Applications, 33-39, IEEE Computer Society Press, 1983.

21. Brooks, RA, Greiner, R, and Binford, TO, The ACRONYM Model-Based Vision System, IJCAI-6, Vol. 1, 105-113, 1979.

22. Bullock, BL, Edwards, GR, Keirsey, DM, Tseng, DY, and Vilnrotter, FM, Image Understanding Application Project: Implementation Progress Report, Proc. Trends & Applications, 50-56, IEEE Computer Society Press, 1983.

23. Malik, J, and Binford, TO, Reasoning in Time and Space, IJCAI-8, Vol. 1, 343-345, 1983.

24. Dean, T, Time Map Maintenance, Yale/UCSD/RR#289, Yale University, October 1983.

25. Tsuji, S, Osada, M, and Yachida, M, Three Dimensional Movement Analysis of Dynamic Line Images, IJCAI-6, Vol. 2, 896-901, 1979.

26. Marimont, DH, A Representation for Image Curves, AAAI-84, 237-242, August 1984.

27. Levitt, TS, Domain Independent Object Description and Decomposition, AAAI-84, 207-211, August 1984.

GLOSSARY

AI	Artificial Intelligence
AIO	An Instance Of
FRL	Frame Representation Language
MAPS	Map Assisted Photo-interpretation System
OPS	Official Production System
ROSIE	Rule-Oriented System for Implementing Expertise
RPU	Regional Processing Unit
SPAM	SPAtial Module
SPAM	Semi-automatic photo-interpretation of high-resolution aerial photographs
STAMMER2	Version 2 of System for Tactical Assessment of Multisource Messages, Even Radar

END

FILMED

4-85

DTIC