DESIGNER: A KNOWLEDGE-BASED
GRAPHIC DESIGN ASSISTANT

Louis Weitzman

July 1986

ICS Report 8609
DESIGNER: A KNOWLEDGE-BASED GRAPHIC DESIGN ASSISTANT

Louis Weitzman

July 1986

ICS Report 8609

NPRDC-UCSD Intelligent Systems Group
Institute for Cognitive Science
University of California, San Diego
La Jolla, California 92093

This research was sponsored by the Personnel and Training Research Programs, Psychological Sciences Division, Office of Naval Research, under Contract No. N00014-85-C-0133, Contract Authority Identification Number, NR 667-541. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the sponsoring agencies. Approved for public release; distribution unlimited. Reproduction in whole or in part is permitted for any purpose of the United States Government. Requests for reprints should be sent to Louis Weitzman, Institute for Cognitive Science, C-015; University of California, San Diego; La Jolla, CA 92093. Copyright © 1986 by Louis Weitzman.
**Title:** Designer: A Knowledge-Based Graphic Design Assistant

**Abstract:** Designer is an interactive tool for assisting with the design of two-dimensional graphic interfaces for instructional systems. Graphic domain knowledge, stored in a frame-based representational facility, is coupled to a domain independent mechanism which analyzes and critiques the user's original design, and then synthesizes design alternatives. These alternative solutions are generated within a design context, or style, and are based upon graphic constraints. The underlying motivation is to improve the quality of the interfaces by making them more consistent and visually more effective.
Contents

OVERVIEW ................................................................. 2
DOMAIN-DEPENDENT KNOWLEDGE REPRESENTATION .......... 4
  Elements ...................................................................... 5
  Relationships ................................................................. 10
  Constraints ............................................................... 10
  Design Context ............................................................. 12
DESIGN PROCESSES ................................................ 13
  Analysis ......................................................................... 13
  Critique ...................................................................... 13
  Synthesis ..................................................................... 14
CONCLUSION ............................................................ 15
ACKNOWLEDGMENTS ................................................ 17
BIBLIOGRAPHY .......................................................... 17
Designer: A Knowledge-Based Graphic Design Assistant

Applying technologies from artificial intelligence and cognitive science to the development of computer-based training and computer aided design systems can provide support in areas where developers and users lack expertise. In addition, intelligent tools can substantially enhance the process of design. Designer is a tool to aid users of Steamer's Graphics Editor. Steamer is a computer-based training system to aid instruction in the domain of propulsion engineering (Hollan, Hutchins, & Weitzman, 1984). It is used to help students develop an understanding of the complex domain of steam propulsion. The system consists of a color graphics interface to a mathematical simulation. One can view and manipulate this simulation at a number of different hierarchical levels through the color interface. The current system contains over one hundred color views which range from abstract, high-level representations of the plant (Figure 1) to detailed views of gauge panels quite like the actual gauge panels in a ship (Figure 2). It was apparent that an editor for creating and maintaining this set of views was essential. The Graphics Editor allows nonprogrammers to graphically create interactive, dynamic views of the simulation. Figure 3 depicts the black-and-white interface of the Editor. This facility has allowed propulsion engineering instructors to create substantial portions of the student interface to this advanced training system. Even though the Editor was built for the construction of Steamer views, the tool is domain independent and has been used to build interfaces monitoring the real-time performance of a computer operating system.

Views are constructed out of graphic components called icons which represent elements in the steam domain. Icons perform two tasks. First, they graphically depict the state of the simulation. For example, pumps are red when stopped and green when operating, dials display their value by positioning an indicator, and pipes show their value by animating their fluid. Second, the user can affect the simulation via the icons. When the user positions a cursor over the icon and clicks the mouse, the state of the icon and its associated value in the simulation are modified. For example, a pump's state toggles from off to on and a dial’s value is set by positioning the indicator. Figure 4 shows a sampling of the types of icons available to users of this Editor. In creating a view, the user selects the icons to be added to the view from a menu on the black-and-white screen. The user then positions and sizes the new icon on the color display. This icon has its parameters defaulted according to the type of icon chosen. Then, through a process of incremental refinement, the user modifies only those attributes that differ in this particular application.

It is unrealistic to assume that instructional designers are facile with graphic design. Facilities were built into the Graphics Editor to support the construction of good views. These facilities include various types of grid latching and the maintenance of aspect ratio for specific icons. However, these constraints were often overridden by the designer. Even working within these constraints, users of the Graphics Editor often violate important graphic design principles and have difficulty maintaining stylistic conventions across sets of views. Designer is a tool to enhance the Graphics Editor by supplementing the designer's domain knowledge with the necessary graphic expertise.
OVERVIEW

Designer provides visual expertise to users of the Graphics Editor interactively constructing new Steamer views or modifying existing ones. It consists of three interrelated processes, an Analyzer, a Critiquer, and a Synthesizer, coupled to a domain-dependent knowledge base. This knowledge base consists of design elements, design relationships, techniques for their identification, sets of constraints for establishing a context, or style, for critiquing a design, and generative techniques for creating design alternatives. The Analyzer first parses a design based on the elements and relationships of the domain, and records this information in the knowledge base. The Critiquer uses this information along with domain-based design constraints to indicate where the current design succeeds or fails. Using knowledge representing the current state of the design, the Synthesizer then generates design alternatives within a design style. The separation of these three processes from the knowledge base provides independence and modularity to the system. This flexibility creates a technology that will be extensible to other design domains.

In order to support the internal mechanisms of Designer, a series of generic subsystems have been incorporated into the system architecture. These tools include Steamer’s frame-based knowledge representation facility (MSG) for storing all domain knowledge, and an Assumption-Based Truth Maintenance System (ATMS) for maintaining alternative design decisions which define the design space.
The system is being developed in the object-oriented programming environment of Flavors on a Symbolics 3600 family processor. The use of object-oriented programming techniques of Flavors has greatly facilitated the implementation and is used throughout the system. A preliminary interface used in the development of the system is shown in Figure 5. The multipaned interface provides access to existing Graphics Editor functions and new Designer functions through scrolling command panes (upper right collection of panes) while access to the domain knowledge is provided in a mouse sensitive graphing pane (upper left pane). A lisp interaction pane is provided (lower left pane) along with a scrolling pane for Designer information (e.g., constraint violations; lower right pane). Designer is just one application, or activity, in a larger instructional simulation environment (Hollan, Hutchins, McCandless, Rosenstein, & Weitzman, in press). Other activities in this environment include a model control facility, Steamer; a view construction facility, the Graphics Editor; a facility to create new icons with new behaviors, the Icon Editor; and a facility to create lessons for students based on particular views and simulation models, the Lesson Editor. A status line, which is consistent throughout all simulation activities, displays information relevant to the current activity. In Designer, the status line (near the bottom of the screen) displays the current values for the system, subsystem, view, and design style. The labels and their values are all mouse sensitive, providing access to functions on the class of item (clicking on the label) or operations on the item itself (clicking on the specific value).
FIGURE 3. Graphics Editor Interface. The Editor is a domain-independent tool allowing nonprogrammers to create graphic interfaces for monitoring and controlling underlying simulations or real-time processes.

DOMAIN-DEPENDENT KNOWLEDGE REPRESENTATION

Much has been written about the knowledge required for graphic design (Bertin, 1983; Cheatham, Cheatham, & Haler, 1983; Ching, 1979; Dondis, 1973; Hurlbut, 1977; Marcus, 1986; Reilly & Roach, 1984; Sherwood, 1981; Taylor, 1960; Wong, 1972). Unfortunately, the literature does not suggest a consistent representation for this knowledge. Designer incorporates much of this knowledge and maintains it in the frame-based representational system, MSG. Designer concentrates on the knowledge describing the domain elements, their relationships, graphic constraints imposed on both the elements and their relationships, and graphic techniques for their modification. For general graphic design the elements refer to points, lines, planes, etc. (Wong, 1972). In Designer, however, these domain elements represent the icons contained within a Steamer view which are characterized by their graphic properties.

MSG, a flavor enhancer, provides a class structure on top of the Flavors object-oriented programming facility. It provides the ability to define classes of objects and create instances of those classes. Each class provides a set of attributes, or slots, that define the characteristics of the class. Slots are grouped together in roles. Slots inherited from the class' abstractions, or parent classes, are included with the locally defined slots to completely describe the class. When a new class is defined, an instance of a meta-class is created that will maintain all pertinent information about the new class. This includes how to create new class instances, where to store the new instances, how to manipulate them, etc. In addition, when a new class is defined, a new flavor of the same name is also defined. Instances of the class are actually instances of this new flavor with the instance variables corresponding to all slot
FIGURE 4. Icon Sampler. This view illustrates a sample of the graphic icons available to designers creating interactive interfaces.

attributes of the class. The new instances are stored on the class object. MSG can be used incrementally, so as new domain knowledge is defined and recorded in the knowledge base, this information will automatically be included in the analyses. Thus, as the system expands, the domain knowledge can easily grow. This ability to incrementally build the domain knowledge is important and increases the system's flexibility.

A tool to create, maintain, and inspect the knowledge base was incorporated into Designer. It provides a flexible graphing facility of the domain-dependent knowledge base. The structure of the graphic class hierarchy is clearly visible in the graphing window of Figure 6. The graph ranges from more abstract classes on the left to more specific classes on the right. The ability to edit and inspect classes and their instances can be accessed through mouse clicks and menu selections. The menu of commands to operate on the class, its instances, or its flavor is shown in Figure 7 for the class elements.

Elements

The MSG class of elements records all domain elements that will be used in subsequent design analyses. The following is the definition of this class which includes instance variables for a name,
a description, all roles (subdivided into slots), all abstractions (parent classes), all used-as-abstractions (classes that use this class as an abstraction), and all of the instances of the class in the current design.

```lisp
#<CLASS ELEMENTS 46622062>, an object of flavor CLASS, has instance variable values:

NAME: ELEMENTS
DESCRIPTION: "a graphic element"
ROLES: ((PROPERTIES ((COLOR NIL NIL) (SIZE NIL NIL) (LOCATION NIL NIL) (TYPE NIL NIL) (SHAPE NIL NIL)) (ELEMENT NIL NIL)))
CONNECTIONS: NIL
SYNONYMS: NIL
ABSTRACTIONS: (GRAPHIC)
USED-AS-ABSTRACTION: NIL
INSTANCES: (#<ELEMENTS DIAL-1 46645216> #<ELEMENTS DIAL-2 46644710> #<ELEMENTS DIAL-3 46644670>)
```

FIGURE 5. Designer Interface. The Designer interface provides access to all Graphics Editor commands while providing additional commands to control the design processes and related functions. Domain-dependent knowledge is displayed in a graphing window pane. State information (e.g., current values for system, subsystem, view, and style) is provided in the status line near the bottom of the screen.
FIGURE 6. Domain Knowledge Base. A graphing tool aids the creation and maintenance of the domain knowledge represented in a frame-based system. Each node in this graph represents a class in the domain of graphic design. Class inheritance is immediately apparent with classes changing from abstract to more specific as one moves through the graph from left to right.

The slots of this class include graphic properties used to distinguish the elements. These are the graphic properties of color, size, location, type, and shape. The values of these properties on an instance are in fact instances of other MSG classes (see Figure 6) that represent valid values for the class. For example, the class of color includes instances for Steamer's basic colors. The class size includes instances describing a range of sizes from very-small to very-large, while the class shape includes instances of basic geometric shapes like linear, circular, rectangular, etc. Figure 8 illustrates the current set of instances for the classes color, size, type, and shape. In addition, there is a class slot to store the domain element an instance of this class will represent.

In the above example, three instances of the class elements are stored on the instance variable instances. All three of these objects represent dial icons in the current view. One of these three objects representing a small, blue dial is shown below.
FIGURE 7. Operations on Domain Knowledge. The mouse sensitive graph nodes provide access to operations on the class, its instances, and its flavor definition through mouse clicks and menu selections. The menu is presenting operations for the selected class element.

#<ELEMENTS DIAL-1 44645216>, an object of flavor ELEMENTS, has instance variable values:

**IDENTIFICATION:** DIAL-1
STRING-FOR-PRINTING: NIL
COLOR: #COLOR BLUE 44644221>
SIZE: ((X #<SIZE SMALL 44644633>)
       (Y #<SIZE SMALL 44644633>))
LOCATION: ((X 0.846)
            (Y 0.521))
TYPE: #<TYPE DIAL 44644216>
SHAPE: #<SHAPE CIRCULAR 44644224>
ELEMENT: #<DIAL 44644230>

This example (of a class definition and description of one of its instances) illustrates that all class slots (i.e., color, size, location, type, shape, and element) become instance variables on the flavor representing the class. These variables have been initialized on the actual instances to the appropriate class values (e.g., the blue instance of class color is stored on the color instance variable, the actual domain element the instance represents is stored on the element instance variable).
<table>
<thead>
<tr>
<th>Instances for class</th>
<th>Instances for class</th>
<th>Instances for class</th>
<th>Instances for class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instances for class</td>
<td>Instances for class</td>
<td>Instances for class</td>
<td>Instances for class</td>
</tr>
<tr>
<td>Instances for class</td>
<td>Instances for class</td>
<td>Instances for class</td>
<td>Instances for class</td>
</tr>
</tbody>
</table>

**FIGURE 8.** Instances of Graphic Property Classes. Menus list the instances of classes representing the four graphic properties: color, size, shape, and type.
Relationships

Currently the graphic relationships in the knowledge base are similarity, proximity, grouping, and repetition. As can be seen in Figure 6, the relationships of similarity, grouping, and repetition are further classified by the graphic properties of the elements (e.g., grouping by color, repetition by type, etc.). The following is the description of the MSG class of similarity of color:

```lisp
#<CLASS SIMILARITY-COLOR 46622203>, an object of flavor CLASS, has instance variable values:
NAME: SIMILARITY-COLOR
DESCRIPTION: "the graphic relationship for the similarity of color"
CLASS-VALUE: COLOR
NUMBER-FOR-NAME: 0
ROLES: ((PROPERTIES ((ELEMENTS NIL NIL NIL)
(State NIL NIL NIL)
(CERTAINTY NIL NIL NIL)))
CONNECTIONS: NIL
SYNONYMS: NIL
ABSTRACTIONS: (SIMILARITY)
USED-AS-ABSTRACTION: NIL
INSTANCES: (#<SIMILARITY-COLOR SIMILARITY-COLOR-BLACK 44645357>
#<SIMILARITY-COLOR SIMILARITY-COLOR-BLUE 44645350>)
```

In the above example, there are two instances of the relation class similarity-color, one for black elements and one for blue elements. The instance representing the relation of similarity of color blue is illustrated below. Here, the previously described dials appear since they all have a blue face color. These elements are stored on the instance variable elements.

```lisp
#<SIMILARITY-COLOR SIMILARITY-COLOR-BLUE 44645350>, an object of flavor SIMILARITY-COLOR, has instance variable values:
IDENTIFICATION: SIMILARITY-COLOR-BLUE
STRING-FOR-PRINTING: NIL
ELEMENTS: (#<ELEMENTS DIAL-1 44645216>
#<ELEMENTS DIAL-2 44644710>
#<ELEMENTS DIAL-3 44644670>)
STATE: NIL
CERTAINTY: :HIGH
```

All relations know how to handle a generic message to identify occurrences in the design of the relation that they represent. When an occurrence is identified, a new instance of the class is created, stored on the class object, and initialized with all the elements participating in the relation. Relations can also build on one another. For example, elements in proximity to one another may form grouping relations, and groupings may form repetition relations (depending on the elements properties and their layout).

Constraints

Domain constraints consist of both basic graphic design principles important in the construction of two-dimensional views and view standards that are adopted for the current application. Principles are those constraints that transcend view sets and are generally accepted methods of making images consistent, unambiguous, and visually effective. The principle of Significant Difference of Size, for instance, states that when elements are a different size, they should be significantly different so as not
to create a sense of ambiguity (Sherwood, 1981) (Figure 9). Elements that are larger represent objects that are more important or physically larger in the real world. If they are not, they should be the same size as other, similar elements in the view. In Figure 1 of the Basic Steam Cycle, the dial indicating RPMs is significantly larger than the others, denoting the fact that it is the most important dial of the set. This principle manifests itself not only in terms of the size of elements but also in terms of the other properties of the elements. The principle in terms of location, for instance, tends to align elements unless there is a reason (of importance or physical fidelity) to accentuate the differences in location. Therefore, in the knowledge base there exist multiple instances of the MSG class of significant difference. Graphic design standards differ from principles because they are special constraints that tend to exist for a given set of designs for a given application. In Steamer, the use of a title is a standard. The restriction of pipes to be within an acceptable range of sizes is another standard.

Constraints can also be categorized as restrictions on properties of elements or restrictions on their relationships. Constraints on properties take the form of discrete constraints, restricting a property to be a specific value, or continuous constraints where the value can range between a minimum and a maximum value. An example of multiple discrete constraints is the standard of Title. Three constraints on the elements properties of type, size, and color restrict the values to be text, large, and yellow, respectively. The standard of Pipe Size is a continuous constraint where the value is restricted between a minimum and maximum size.

**Principle of Significant Difference of Size**

![Diagram of Principle of Significant Difference of Size](image)

**FIGURE 9. The Principle of Significant Difference of Size.** This design principle states that when elements are a different size they should be significantly different so as not to create a sense of ambiguity. Given an original design consisting of three dials, two alternatives are presented from a larger solution space. The first alternative suggests no difference in importance and therefore no difference in size. Alternative 2 takes into account the fact that the right two dials are grouped together. The other dial, being physically separate and larger, may be perceived as more important and therefore should be significantly larger.
Design Context

A design should be sensitive to the context in which it is created. It is this context that defines the external constraints which shape and guide the final solution. In Designer, this context is referred to as a style and is constructed by selecting those constraints (principles and standards) that are to be enforced within this context. Good design in one style may not necessarily be good design in another. Modifying the style within which a critique is made ultimately affects the final form of the design.

A graphic style is also defined by the visual techniques employed in the communication of information (Dondis, 1973). These visual techniques represent a vocabulary in which to describe the design. These techniques in conjunction with the constraints may suggest a variety of graphic procedures to modify an alternative. These procedures are similar to Mittal, Dym, and Morjaria’s (1986) design methods. For example, the visual technique of Regularity may take on a value of regular, neutral, or irregular, each suggesting alternatives consistent with its definition. Highly regular designs will accentuate similarities of elements and relationships, while irregular designs accentuate the differences. It is the constraints that indicate a discrepancy in the design, while the interaction of the techniques suggest the graphic procedures (maybe more than one) that will modify the design. Figure 10 illustrates a style editor which allows the user to name a style, select graphic constraints to be active within it, and choose values for the visual techniques by selecting the appropriate value.

**FIGURE 10. Style Editor.** This menu edits the graphic style, or context, in which a design critique occurs. A style is defined by the graphic constraints (i.e., principles and standards) that are active and the values chosen for the visual techniques. These visual techniques in combination with the constraints generate the graphic procedures for modifying the design.
DESIGN PROCESSES

Design involves a cycle of gathering information, making decisions based on that information, and reviewing the consequences of those decisions. New information gleaned from this process is incorporated back into the cycle for subsequent refinement of the design. This analysis-synthesis-review cycle is a general process used in all design whether it be for computer interfaces, industrial applications, or architecture. The process is domain independent.

In order for Designer to carry out the analysis phase of the process, two steps must take place. First, the system must parse the design into domain elements and relationships. This is Designer's Analysis process. Second, the system must locate areas that need to be improved. This is the Critique process. Together, these processes represent the analysis phase of the generic design process. Once the first two steps have occurred, the system is ready to suggest alternative procedures for modifying the design. This is Designer's Synthesis process.

Since the overall goal is for the system to be an online assistant and not assume control, review occurs interactively with the user selecting and confirming decisions presented by the system. Information is incorporated back through the process as output from one cycle becomes the input for the next cycle of critique. An alternative approach (Brown & Chandrasekaran, 1986), which may be incorporated in future versions of the system, is to provide a plan which specifies the order in which design steps are invoked by the motivating techniques. Each of Designer's processes is described in more detail below.

Analysis

The analysis process parses the design and locates existing domain elements and relationships. Identifying the elements is straightforward because of the use of icons in the Steamer interface and the object-oriented nature of their implementation. An instance of the MSG class elements is created for each icon. The elements instance variables are appropriately initialized with values for each property being an instance of the corresponding MSG class.

Once the domain elements have been created, the system locates instances of domain relationships. This task is easy for people but very difficult for computers. Much work has been done in the area of image analysis, but seldom with the goal of beautifying drawings (Pavlidis & Van Wyk, 1985). To maintain the independent nature of the analysis, generic messages are sent to each relation class to identify instances of the class within the design. When an occurrence is found, an instance of the MSG relation class is created and initialized. This includes the recording of the elements that participate in the relation on the appropriate MSG slot.

Critique

As Christopher Alexander suggests, the notion of a misfit is more compelling than a fit and is a driving force behind the ultimate shape of a design (Alexander, 1974). In Designer, the misfits are identified as violations of the domain constraints. The Critiquer creates a comment for each unsatisfied design constraint within the current style. These critique comments are Flavor objects that store their underlying constraint and the elements involved in the violation. Descriptions and justifications of the comments that are based on this constraint can then be presented. These comments, displayed in the scrolling pane of the black-and-white interface, are mousable and can be highlighted (graphically highlighting those elements involved) and described in terms of their underlying principle or standard. Critiques themselves are implemented as flavor objects that store the object being critiqued (the Steamer view), the style in which the critique takes place, and a list of all the relevant comments for this object and style. Figure 11 illustrates a critique based on the principle of the Significant Difference of Size of three dials shown in the original design of Figure 9. Two violations of this principle, one for
**Clique of Three Dials.** The Designer interface illustrates a critique in progress of the original design from Figure 9. This critique, being executed in the Steamer style, has generated two critique comments based on the principle of Significant Difference of Size. One comment refers to elements of a similar type while the other refers to elements similar in shape. A description of one violation is presented in the lisp pane and a menu of techniques for modifying the design based on this violation is presented in a pop-up menu.

Similar typed elements and one for similar shaped elements are displayed in the scrolling pane. A description of the first violation is presented in the lisp pane.

It thus becomes possible under this paradigm to request multiple critiques, each based on a different style. This is an especially powerful paradigm for views that may need to be presented in different media, each with different constraints. For example, a style appropriate for a high resolution color display may not be appropriate for a black-and-white hardcopy presentation where features are not as clearly distinguishable.

**Synthesis**

Design decisions are made in the synthesis phase in order to incrementally refine the elements and their relationships. Knowledge of the elements and their relationships along with the comments from the critiquing phase forms the basis for these design modifications. Each comment communicates to the constraint on which it is based via generic messages in order to determine the graphic procedures for satisfying the existing violation. More than one procedure may be available to satisfy the constraint and all possibilities are presented to the user. These procedures are a result of the interaction of the various visual techniques and the design constraints which describe the style.

Once the user decides to remedy a given comment by clicking on it with the mouse, various graphic procedures are presented in a pop-up menu (Figure 11). These procedures will all modify the design in
order to satisfy the constraint, but will do so differently. Since there is no correct solution these procedures represent alternatives and there is no attempt to suggest that one alternative may be better. The variation of alternatives are based on the definition of the style's visual techniques. For example, if a style is defined to be simple and regular, the constraint of Significant Difference of Size will generate a very different solution than if the style is defined as complex and irregular. Figure 12 illustrates alternative solutions in three different styles all defined with the constraints of Significant Difference of Size and of Location. The only difference between the styles is the articulation of the visual techniques ranging from simple and regular (Style 1, Figure 12A) to complex and irregular (Style 3, Figure 12B). With the same initial design, each style creates different solutions. These solutions satisfy the constraints but are based on varying procedures of generation from the defined visual techniques. In Style 1, the system looks for the simplest most regular solution possible. This results with all dials in each solution being the same size and aligned on an axis (similar location). Style 3, on the other hand, has chosen the opposite approach where no two dials in the final solution are the same size and no alignment occurs. Style 2 (Figure 12C) takes a more moderate approach with two distinct (and significantly different) dial sizes and some alignment. This simple example illustrates how the interaction of the graphic constraints with the visual techniques will generate alternative solutions.

These alternatives are maintained by a new form of truth maintenance system, an ATMS (De Kleer, 1986a, 1986b, 1986c). With the ATMS multiple alternatives are maintained and can be explored simultaneously. Unlike previous truth maintenance systems which just manipulated justifications, this system additionally manipulates assumption sets. As a result, inconsistent information can exist and it is possible to work effectively and efficiently in the problem space. Context switching is free, and most backtracking and all retraction is avoided. In Designer, the assumptions that are manipulated are the alternatives created by incremental design decisions. Solutions at any stage in the design process are the consistent, noncontradictory environments maintained by the ATMS. Any contradictions that arise are handled by the ATMS and will not appear in the same environment.

This new form of truth maintenance system is well suited for tracking multiple alternatives in the design space where a reasonable number of the potential solutions must be examined. Designer interacts with this system by creating an ATMS class for each domain element. Whenever an element is modified, a new ATMS node is added to this class. These classes represent the different alternatives of the original domain element. Multiple nodes coexist in the solution space but only one will be present in any ATMS solution. The justifications of what style is current and what constraint generates the modified element can be added to these nodes to further restrict the space of valid solutions.

Because context switching is free, the user can explore the design space by interactively inspecting the individual ATMS environments. Each solution can be displayed on the color screen and explain itself in terms of the underlying assumptions and justifications. Based on these assumptions and justifications, an alternative can describe its derivation and individual decisions can be described in terms of their potential contribution to a final solution. The system thus conveys design precepts while the user is viewing a specific instantiation of a design alternative. Thus, the user’s knowledge of constructing visual presentations is enhanced for future design.

CONCLUSION

There exists a number of interesting research projects similar in nature to Designer. They are all knowledge-based systems providing an environment to aid the creation and verification of design alternatives. Some provide exploratory environments in well-defined domains (e.g., Palladio, for circuit design [Brown, Tong, & Foyster, 1983]) while others, like Designer, are systems in the ill-defined domain of graphic design (e.g., ACE: A Color Expert, an expert system for the selection of colors for synthetic scene imagery [Meier, 1986]; Descriptor, a generative system for graphic layout based on shape grammars [Glenn, 1986]). Some of these systems, like Designer, try to encode the general process of design and then apply it to a prototypical domain (e.g., PRIDE, for the design of paper handling
FIGURE 12. Different Styles Generating Different Alternatives. Given the same view as input, three different styles generate three completely different solutions. All three styles include the principles of Significant Difference of Size and of Location. They differ only in the articulation of the visual techniques defined, from simple and regular (A) to complex and irregular (C). The first level of design decisions is in response to element size while the second level is based on location considerations. These alternatives represent only a small portion of the solution space.
systems (Mittal et al., 1986)), while others use sophisticated tools (e.g., production systems and logic programming) on specific design problems. Designer differs slightly from these systems in that it is a reactive system. It responds to the users actions, by analyzing, critiquing, and interactively suggesting improvements to them. Currently, it does not try to create a new design given high-level design goals.

An initial implementation of Designer is underway. A functioning system has been used on existing Steamer diagrams and has provided useful feedback. It is very encouraging that even in views that were carefully crafted, the system was able to note inconsistencies and suggest improvements. In addition, a preliminary use of the ATMS has shown the feasibility of using it to maintain solutions in the design space.

The perception of a problem and the shape of its solution are both affected by the depth and range of the design vocabulary (Ching, 1979). Therefore, it is important that the domain knowledge base continues to grow. Only a few constraints currently exist and as more principles and standards are defined, more complete and robust solutions will be presented. As more solutions become available, better techniques to explore and understand their differences will be necessary.

ACKNOWLEDGMENTS

This project began as an Independent Exploratory Development Program (IED) with the Navy Personnel Research and Development Center in San Diego. Additional funding for this research is provided by the Personnel and Training Research Program of the Office of Naval Research (Contract N00014-85-C-0133, NR 667-541) and the Office of Naval Technology (522-801-018). I would like to thank all those involved with the UCSD/NPRDC joint program for making this research possible. In particular I would like to thank members of the group for their support critiquing early versions of this paper, including Jim Hollan, Mark Rosenstein, Dave Owen, Larry West, Barbara Morris, and Kathy Farrelly.

BIBLIOGRAPHY


ICS Technical Report List

The following is a list of publications by people in the Institute for Cognitive Science. For reprints, write or call:

Institute for Cognitive Science, C-015
University of California, San Diego
La Jolla, CA 92093
(619) 452-6771


Earlier Reports by People in the Cognitive Science Lab

The following is a list of publications by people in the Cognitive Science Lab and the Institute for Cognitive Science. For reprints, write or call:

Institute for Cognitive Science, C-015
University of California, San Diego
La Jolla, CA 92093
(619) 452-6771


<table>
<thead>
<tr>
<th>Name</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dr. Michelele Chi</td>
<td>Learning &amp; D Center</td>
</tr>
<tr>
<td>University of Pittsburgh</td>
<td>3939 O'Hara Street</td>
</tr>
<tr>
<td>Pittsburgh, PA 15213</td>
<td></td>
</tr>
<tr>
<td>Dr. William Clancey</td>
<td>Stanford University</td>
</tr>
<tr>
<td>Knowledge Systems Laboratory</td>
<td>701 Welch Road, Bldg. C</td>
</tr>
<tr>
<td>Palo Alto, CA 94304</td>
<td></td>
</tr>
<tr>
<td>Director, Manpower Support and Readiness Program</td>
<td>Center for Naval Analysis</td>
</tr>
<tr>
<td></td>
<td>2000 North Beauregard Street</td>
</tr>
<tr>
<td></td>
<td>Alexandria, VA 22311</td>
</tr>
<tr>
<td>Dr. Michael Cole</td>
<td>University of California</td>
</tr>
<tr>
<td>at San Diego</td>
<td>Laboratory of Comparative</td>
</tr>
<tr>
<td></td>
<td>Human Cognition - D003A</td>
</tr>
<tr>
<td></td>
<td>La Jolla, CA 92039</td>
</tr>
<tr>
<td>Dr. Allan M. Collins</td>
<td>Bolt Beranek &amp; Newman, Inc.</td>
</tr>
<tr>
<td></td>
<td>50 Moulton Street</td>
</tr>
<tr>
<td></td>
<td>Cambridge, MA 02138</td>
</tr>
<tr>
<td>Dr. Stanley Cullera</td>
<td>Office of Naval Technology</td>
</tr>
<tr>
<td></td>
<td>Code 222</td>
</tr>
<tr>
<td></td>
<td>800 N. Quincy Street</td>
</tr>
<tr>
<td></td>
<td>Arlington, VA 22217-5000</td>
</tr>
<tr>
<td>Dr. Lyn A. Cooper</td>
<td>Learning R&amp;D Center</td>
</tr>
<tr>
<td>University of Pittsburgh</td>
<td>3939 O'Hara Street</td>
</tr>
<tr>
<td>Pittsburgh, PA 15213</td>
<td></td>
</tr>
<tr>
<td>LT Judy Crookshanks</td>
<td>Chief of Naval Operations</td>
</tr>
<tr>
<td></td>
<td>OP-112G5</td>
</tr>
<tr>
<td></td>
<td>Washington, DC 20370-2000</td>
</tr>
<tr>
<td>Dr. Kenneth B. Cross</td>
<td>Anacapa Sciences, Inc.</td>
</tr>
<tr>
<td></td>
<td>P.O. Drawer Q</td>
</tr>
<tr>
<td></td>
<td>Santa Barbara, CA 93102</td>
</tr>
<tr>
<td>Capt. Stephen Cross (Ph.D.)</td>
<td>Artificial Intelligence Laboratory</td>
</tr>
<tr>
<td></td>
<td>Wright-Patterson AFB, OH 45433</td>
</tr>
<tr>
<td>CAPT P. Michael Curran</td>
<td>Office of Naval Research</td>
</tr>
<tr>
<td></td>
<td>800 N. Quincy St.</td>
</tr>
<tr>
<td></td>
<td>Code 125</td>
</tr>
<tr>
<td></td>
<td>Arlington, VA 22217-5000</td>
</tr>
<tr>
<td>Dr. Cary Chichon</td>
<td>Intelligent Instructional Systems</td>
</tr>
<tr>
<td></td>
<td>Texas Instruments AI Lab</td>
</tr>
<tr>
<td></td>
<td>P.O. Box 660245</td>
</tr>
<tr>
<td></td>
<td>Dallas, TX 75266</td>
</tr>
<tr>
<td>Brian Dallman</td>
<td>3400 TTM/TGKS</td>
</tr>
<tr>
<td></td>
<td>Lowry AFB, CO 80230-5000</td>
</tr>
<tr>
<td>LT John Deaton</td>
<td>OMR Code 125</td>
</tr>
<tr>
<td></td>
<td>800 N. Quincy Street</td>
</tr>
<tr>
<td></td>
<td>Arlington, VA 22217-5000</td>
</tr>
<tr>
<td>Mr. Paul DiMauro</td>
<td>Commandant of the Marine Corps</td>
</tr>
<tr>
<td></td>
<td>Code 1AC-4</td>
</tr>
<tr>
<td></td>
<td>Washington, DC 20380</td>
</tr>
<tr>
<td>Dr. R. K. Dismukes</td>
<td>Associate Director for Life Sciences</td>
</tr>
<tr>
<td></td>
<td>AFSOR</td>
</tr>
<tr>
<td></td>
<td>Bolling AFB</td>
</tr>
<tr>
<td></td>
<td>Washington, DC 20332</td>
</tr>
<tr>
<td>Dr. Emanuel Donchin</td>
<td>University of Illinois</td>
</tr>
<tr>
<td></td>
<td>Department of Psychology</td>
</tr>
<tr>
<td></td>
<td>Champaign, IL 61820</td>
</tr>
<tr>
<td>Defense Technical Information Center</td>
<td>Cameron Station, Bldg 5</td>
</tr>
<tr>
<td></td>
<td>Alexandria, VA 22314</td>
</tr>
<tr>
<td></td>
<td>Attn: TC</td>
</tr>
<tr>
<td></td>
<td>(12 Copies)</td>
</tr>
<tr>
<td>Dr. Thomas M. Duffy</td>
<td>Communications Design Center</td>
</tr>
<tr>
<td></td>
<td>Carnegie-Mellon University</td>
</tr>
<tr>
<td></td>
<td>Schenley Park</td>
</tr>
<tr>
<td></td>
<td>Pittsburgh, PA 15213</td>
</tr>
<tr>
<td>Edward E. Edlowes</td>
<td>CHATRA H301</td>
</tr>
<tr>
<td></td>
<td>Naval Air Station</td>
</tr>
<tr>
<td></td>
<td>Corpus Christi, TX 78419</td>
</tr>
<tr>
<td>Dr. Jeffrey Elman</td>
<td>University of California, San Diego</td>
</tr>
<tr>
<td></td>
<td>Department of Linguistics, C-008</td>
</tr>
<tr>
<td></td>
<td>La Jolla, CA 92039</td>
</tr>
<tr>
<td>Dr. Richard Elster</td>
<td>Deputy Assistant Secretary of the Navy</td>
</tr>
<tr>
<td></td>
<td>(Manpower)</td>
</tr>
<tr>
<td></td>
<td>OASN (M&amp;RA)</td>
</tr>
<tr>
<td></td>
<td>Department of the Navy</td>
</tr>
<tr>
<td></td>
<td>Washington, DC 20350-1000</td>
</tr>
<tr>
<td>ERIC Facility Acquisitions</td>
<td>4833 Rugby Avenue</td>
</tr>
<tr>
<td></td>
<td>Bethesda, MD 20014</td>
</tr>
<tr>
<td>Dr. R. Anders Ericsson</td>
<td>University of Colorado</td>
</tr>
<tr>
<td></td>
<td>Department of Psychology</td>
</tr>
<tr>
<td></td>
<td>Boulder, CO 80309</td>
</tr>
<tr>
<td>Dr. Edward Esty</td>
<td>Department of Education, OERI</td>
</tr>
<tr>
<td></td>
<td>Room 710</td>
</tr>
<tr>
<td></td>
<td>1200 19th St., NW</td>
</tr>
<tr>
<td></td>
<td>Washington, DC 20208</td>
</tr>
<tr>
<td>Dr. Beatrice J. Farr</td>
<td>Army Research Institute</td>
</tr>
<tr>
<td></td>
<td>5001 Eisenhower Avenue</td>
</tr>
<tr>
<td></td>
<td>Alexandria, VA 22333</td>
</tr>
<tr>
<td>Dr. Marshall J. Farr</td>
<td>2520 North Vernon Street</td>
</tr>
<tr>
<td></td>
<td>Arlington, VA 22207</td>
</tr>
<tr>
<td>Dr. Pat Federico</td>
<td>Code 511</td>
</tr>
<tr>
<td></td>
<td>NROCN</td>
</tr>
<tr>
<td></td>
<td>San Diego, CA 92152-6800</td>
</tr>
<tr>
<td>Dr. Paul Feltovich</td>
<td>Southern Illinois University</td>
</tr>
<tr>
<td></td>
<td>School of Medicine</td>
</tr>
<tr>
<td></td>
<td>Medical Education Department</td>
</tr>
<tr>
<td></td>
<td>P.O. Box 3926</td>
</tr>
<tr>
<td></td>
<td>Springfield, IL 62708</td>
</tr>
<tr>
<td>Mr. Wallace Feurstein</td>
<td>Educational Technology</td>
</tr>
<tr>
<td></td>
<td>Bolt Beranek &amp; Newman</td>
</tr>
<tr>
<td></td>
<td>10 Moulton St.</td>
</tr>
<tr>
<td></td>
<td>Cambridge, MA 02238</td>
</tr>
<tr>
<td>Dr. Craig I. Fields</td>
<td>ARPA</td>
</tr>
<tr>
<td></td>
<td>1400 Wilson Blvd.</td>
</tr>
<tr>
<td></td>
<td>Arlington, VA 22209</td>
</tr>
<tr>
<td>J. D. Fletcher</td>
<td>9931 Corsica Street</td>
</tr>
<tr>
<td></td>
<td>Vienna VA 22180</td>
</tr>
<tr>
<td>Dr. Linda Flower</td>
<td>Carnegie-Mellon University</td>
</tr>
<tr>
<td></td>
<td>Department of English</td>
</tr>
<tr>
<td></td>
<td>Pittsburgh, PA 15213</td>
</tr>
<tr>
<td>Dr. Barbara A. Fox</td>
<td>University of Colorado</td>
</tr>
<tr>
<td></td>
<td>Department of Linguistics</td>
</tr>
<tr>
<td></td>
<td>Boulder, CO 80309</td>
</tr>
<tr>
<td>Dr. Carl H. Frederiksen</td>
<td>McGill University</td>
</tr>
<tr>
<td></td>
<td>3700 McTavish Street</td>
</tr>
<tr>
<td></td>
<td>Montreal, Quebec H3A 1Y2</td>
</tr>
<tr>
<td></td>
<td>CANADA</td>
</tr>
<tr>
<td>Dr. John R. Frederiksen</td>
<td>Bolt Beranek &amp; Newman</td>
</tr>
<tr>
<td></td>
<td>50 Moulton Street</td>
</tr>
<tr>
<td></td>
<td>Cambridge, MA 02138</td>
</tr>
<tr>
<td>Julie A. Gaddis</td>
<td>Information Technology</td>
</tr>
<tr>
<td></td>
<td>Applications Division</td>
</tr>
<tr>
<td></td>
<td>Admixture Research Establishment</td>
</tr>
<tr>
<td></td>
<td>Portadown, Portsmouth POE 4A</td>
</tr>
<tr>
<td></td>
<td>UNITED KINGDOM</td>
</tr>
<tr>
<td>Dr. Michael Gennsereth</td>
<td>Stanford University</td>
</tr>
<tr>
<td></td>
<td>Computer Science Department</td>
</tr>
<tr>
<td></td>
<td>Stanford, CA 94305</td>
</tr>
<tr>
<td>Dr. Dedre Gentner</td>
<td>University of Illinois</td>
</tr>
<tr>
<td></td>
<td>Department of Psychology</td>
</tr>
<tr>
<td></td>
<td>603 E. Daniel St.</td>
</tr>
<tr>
<td></td>
<td>Champaign, IL 61820</td>
</tr>
<tr>
<td>Name</td>
<td>Address</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td>Dr. Don Gentner</td>
<td>Center for Human Information Processing</td>
</tr>
<tr>
<td>Dr. Robert Glaser</td>
<td>Learning Research &amp; Development Center</td>
</tr>
<tr>
<td>Dr. Marvin D. Glick</td>
<td>13 Stone Hall Cornell University</td>
</tr>
<tr>
<td>Dr. Sam Glicksberg</td>
<td>Department of Psychology Princeton University</td>
</tr>
<tr>
<td>Dr. Joseph Goguen</td>
<td>Computer Science Laboratory 581 International</td>
</tr>
<tr>
<td>Dr. Sherris Hott</td>
<td>AFRL/MD1J Brooks AFB, TX 78235</td>
</tr>
<tr>
<td>Dr. Wayne Gray</td>
<td>Army Research Institute 5001 Eisenhower Avenue</td>
</tr>
<tr>
<td>Dr. Bert Green</td>
<td>Johns Hopkins University Department of Psychology Charles &amp; 34th Street Baltimore, MD 21218</td>
</tr>
<tr>
<td>Dr. James G. Greeno</td>
<td>University of California University of California Berkeley, CA 94720</td>
</tr>
<tr>
<td>N. William Greenup</td>
<td>Education Advisor (EO31) Education Center MCDEC Quantico, VA 22134</td>
</tr>
<tr>
<td>Donald Haggard</td>
<td>Fort Knox Field Unit Army Research Institute Steele Hall Ft. Knox, KY 40121</td>
</tr>
<tr>
<td>Dr. Henry M. Halff</td>
<td>Half Resources, Inc. 4918 33rd Road, North Arlington, VA 22203</td>
</tr>
<tr>
<td>Dr. Ronald K. Hamblin</td>
<td>Prof. of Education &amp; Psychology University of Massachusetts at Amherst Hills House Amherst, MA 01003</td>
</tr>
<tr>
<td>Dr. Bruce Hamill</td>
<td>The Johns Hopkins University Applied Physics Laboratory Laurel, MD 20707</td>
</tr>
<tr>
<td>Dr. Bruce W. Hamill</td>
<td>Johns Hopkins University Applied Physics Laboratory Johns Hopkins Road Laurel, MD 20707</td>
</tr>
<tr>
<td>Prof. John R. Hayes</td>
<td>Carnegie Mellon University Department of Psychology Schenley Park Pittsburgh, PA 15213</td>
</tr>
<tr>
<td>Dr. Barbara Hayes-Roth</td>
<td>Department of Computer Science Stanford University Stanford, CA 94305</td>
</tr>
<tr>
<td>Dr. Frederick Hayes-Roth</td>
<td>Teknowledge 525 University Ave., Palo Alto, CA 94301</td>
</tr>
<tr>
<td>Dr. Joan I. Heller</td>
<td>505 Haddan Road Oakland, CA 94606</td>
</tr>
<tr>
<td>Dr. Steven A. Hillyard</td>
<td>Department of Neurosciences University of California, San Diego La Jolla, CA 92033</td>
</tr>
<tr>
<td>Dr. Geoffrey Hinton</td>
<td>Carnegie-Mellon University Computer Science Department Pittsburgh, PA 15213</td>
</tr>
<tr>
<td>Dr. Jim Holland</td>
<td>Intelligent Systems Group Institute for Cognitive Science (C-015) UCSD La Jolla, CA 92093</td>
</tr>
<tr>
<td>Dr. Ed Hutchins</td>
<td>Intelligent Systems Group Institute for Cognitive Science (C-015) UCSD La Jolla, CA 92093</td>
</tr>
<tr>
<td>Dr. Zachary Jacobson</td>
<td>Bureau of Management Consulting 365 Laurier Avenue West Ottawa, Ontario K1A 0S5 CANADA</td>
</tr>
<tr>
<td>COL Dennis W. Jervi</td>
<td>Commander AFRL Brooks AFB, TX 78235-5601</td>
</tr>
<tr>
<td>Dr. Robin Jeffries</td>
<td>Hewlett-Packard Laboratories P.O. Box 10490 Palo Alto, CA 94303-0971</td>
</tr>
<tr>
<td>Margaret Jerome</td>
<td>c/o Dr. Peter Chandler 83, The Drive Hove Sussex UNITED KINGDOM</td>
</tr>
<tr>
<td>Dr. Joseph E. Johnson</td>
<td>Assistant Dean for Graduate Studies College of Science and Mathematics University of South Carolina Columbia, SC 29208</td>
</tr>
<tr>
<td>CDR Tom Jones</td>
<td>ONR Code 125 800 N. Quincy Street Arlington, VA 22217-5000</td>
</tr>
<tr>
<td>Dr. Douglas H. Jones</td>
<td>Thatcher Jones Associates P.O. Box 6640 10 Trafalgar Court Lawrenceville, NJ 08648</td>
</tr>
<tr>
<td>Dr. Marcel Just</td>
<td>Carnegie-Mellon University Department of Psychology Schenley Park Pittsburgh, PA 15213</td>
</tr>
</tbody>
</table>
Dr. Milton S. Katz
Army Research Institute
5001 Eisenhower Avenue
Alexandria, VA 22333

Dr. Dennis Kibler
University of California
Department of Information
and Computer Science
Irvine, CA 92717

Dr. David Klaras
University of Michigan
Technical Communication
College of Engineering
1223 E. Engineering Building
Ann Arbor, MI 48109

Dr. Peter Kincade
Training Analysis
& Evaluation Group
Department of the Navy
Orlando, FL 32813

Dr. Walter Kintsch
Department of Psychology
University of Colorado
Campus Box 345
Boulder, CO 80302

Dr. Paula Kirk
Oakland University
University Programs Division
P.O. Box 117
Oakridge, TN 37831-0117

Dr. Hazel Knerr
Program Manager
Training Research Division
HumRRO
1100 S. Washington
Alexandria, VA 22314

Dr. Ronald Knoll
Bell Laboratories
Murray Hill, NJ 07974

Dr. Janet L. Koldner
Georgia Institute of Technology
School of Information
& Computer Science
Atlanta, GA 30332

Dr. Sylvan Kornblum
University of Michigan
Mental Health Research Institute
205 Washtenaw Place
Ann Arbor, MI 48109

Dr. Stephen Kosslyn
Harvard University
1236 William James Hall
33 Kirkland St.
Cambridge, MA 02138

Dr. David H. Krantz
2 Washington Square Village
Apt. #167
New York, NY 10012

Dr. David R. Lambert
Naval Ocean Systems Center
Code 441T
271 Catalina Boulevard
San Diego, CA 92152-6800

Dr. Pat Langley
University of California
Department of Information
and Computer Science
Irvine, CA 92717

Dr. Marcy Lamsman
University of North Carolina
The L. L. Thurstone Lab.
Davie Hall 013A
Chapel Hill, NC 27514

Dr. Jill Larkin
Carnegie-Mellon University
Department of Psychology
Pittsburgh, PA 15213

Dr. Robert Lawler
Information Sciences, FRL
GTE Laboratories, Inc.
40 Sylvan Road
Waltham, MA 02254

Dr. Paul E. Lehner
PAR Technology Corp.
7926 Jones Branch Drive
Suite 170
McLean, VA 22102

Dr. Thomas Leonard
University of Wisconsin
Department of Statistics
1210 West Dayton Street
Madison, WI 53705

Dr. Alan M. Leavold
Learning R&D Center
University of Pittsburgh
Pittsburgh, PA 15260

Dr. Alan Lashever
Deputy Division Director
Behavioral and Neural Sciences
National Science Foundation
1800 G Street
Washington, DC 20550

Dr. Jim Levin
University of California
Laboratory for Comparative
Human Cognition
DO03A
La Jolla, CA 92037

Dr. Clayton Lewis
University of Colorado
Department of Computer Science
Campus Box 410
Boulder, CO 80309

Library
Naval War College
Newport, RI 02840

Library
Naval Training Systems Center
Orlando, FL 32813

Science and Technology Division
Library of Congress
Washington, DC 20540

Dr. Frederic M. Lord
Educational Testing Service
Princeton, NJ 08541

Dr. Don Lyon
P. O. Box 44
Higley, AZ 85236

Dr. William L. Maloy
Chief of Naval Education
and Training
Naval Avulsion
Pennacola, FL 32508

Dr. Evan Mandes
Dept. of Psychology
George Mason University
4400 University Drive
Fairfax, VA 22030

Dr. Sandra P. Marshall
San Diego State University
San Diego, CA 92102

Dr. Manton M. Matthews
Department of Computer Science
University of South Carolina
Columbia, SC 29708

Dr. Richard E. Mayer
Department of Psychology
University of California
Santa Barbara, CA 93106

Dr. Kathleen McKeown
Columbia University
Department of Computer Science
New York, NY 10027

Dr. Gail McKoon
CAS/Special Education
Northwestern University
1859 Sheridan Road
Kleage #230
Evanston, IL 60201

Dr. Joe McLachlan
Naval Personnel R&D Center
San Diego, CA 92152-6800

Dr. James McMichael
Assistant for MPT Research,
Development, and Studies
OP 0187
Washington, DC 20370
ONR DISTRIBUTION LIST

LCDR Cory deGroot Whitehead
Chief of Naval Operations
OP-112G1
Washington, DC 20370-2000

Dr. Christopher Wickens
Department of Psychology
University of Illinois
Champaign, IL 61820

Dr. Michael Williams
IntelliCorp
1975 El Camino Real West
Mountain View, CA 94040-2216

A. K. Winterbauer
Research Associate
Electronics Division
Denver Research Institute
University Park
Denver, CO 80208-0454

Dr. Robert A. Wischer
U.S. Army Institute for the
Behavioral and Social Sciences
5001 Eisenhower Avenue
Alexandria, VA 22333

Dr. Martin F. Wiskoff
Navy Personnel R & D Center
San Diego, CA 92152-6800

Dr. Frank Withrow
U.S. Office of Education
400 Maryland Ave. SW
Washington, DC 20202

Mr. John H. Wolfe
Navy Personnel R&D Center
San Diego, CA 92152-6800

Dr. Wallace Wulfeck, III
Navy Personnel R&D Center
San Diego, CA 92152-6800

Dr. Joe Yasutake
AFHRL/LRT
Lowry AFB, CO 80230

Mr. Carl York
System Development Foundation
181 Lytton Avenue
Suite 210
Palo Alto, CA 94301

Dr. Joseph L. Young
Memory & Cognitive Processes
National Science Foundation
Washington, DC 20550

Dr. Steven Zornetzer
Office of Naval Research
Code 1140
800 N. Quincy St.,
Arlington, VA 22217-5000