

Presentation Planning Using an Integrated Knowledge Base*

Yigal Arens
Lawrence Miller
Norman Sondheimer

USC/Information Sciences Institute
4676 Admiralty Way
Marina Del Rey, CA 90292

1 Introduction

ISI is involved in user interface research aimed at bringing together multiple input and output modes in a way that handles mixed mode input (commands, menus, forms, natural language), interacts with a diverse collection of underlying software utilities in a uniform way, and presents the results through a combination of output modes including natural language text, maps, charts and graphs.

Our system, **Integrated Interfaces**, derives much of its ability to interact uniformly with the user and the underlying services and to build its presentations, from the information present in a central knowledge base. This knowledge base integrates models of: the application domain (Navy ships in the Pacific region, in the current demonstration version); the structure of visual displays and their graphical features; the underlying services (data bases and expert systems); and interface functions. The emphasis in this paper is on a presentation planner that uses the knowledge base to produce multi-modal output.

There has been a flurry of recent work in user interface management systems (we list several recent examples in the references). Existing work is characterized by an attempt to relieve the software designer of the burden of handcrafting an interface for each application. The work has generally focused on intelligently handling input. In our paper we deal with the other end of the pipeline - presentations.

1.1 Presentation Planning

Presentations are put together by a **Presentation Planner**. The presentation planner decides what output mode or combinations of output modes to use for each piece of information. This involves recognition of the topic of the information, classification of the topic, a check of the user's

*This research is supported by the Defense Advanced Research Projects Agency under Contract No. N0014-87-K-0130. Views and conclusions contained in this report are the authors' and should not be interpreted as representing the official opinion or policy of DARPA, the U.S. Government, or any person or agency connected with them.

ORIGINAL PAGE IS
OF POOR QUALITY

preferences for presentation, and a coordinated delegation activity to assign tasks to the various output modes. This is done by rules that map between concepts and display modes.

In moving from an interface with a single output device to an integrated multiple output device interface, output processing changes substantially. Even in single-mode systems, we find that some preparation is necessary beyond the mere determination of the contents of presentations. For example, an information retrieval system may use tables exclusively for the display of retrieved data. Such a system may still decide to split information between tables in a report to control the length of the tables before the final output is generated. In an integrated presentation system, such planning activity grows considerably. The system must be able to decide what output mode to use for each piece of information.

The research issues that must be addressed in this context include determining what constitutes a good presentation of information, how to recognize information presentation situations, how to build knowledge that can be shared across several modalities, and how to choose the mode and form of output.

1.2 Planning as a Paradigm

In Presentation Planning, the use of the term *planning* is intended to bring to mind the AI sense of planning, where a system attempts to achieve a goal, executing certain operations at its disposal. The goal of our presentation planner is the presentation of some information, e.g., the status of a fleet of ships. The actions are commitments to present part of that information in some form, e.g., to present a map of the ships' positions showing the direction and speed of each by the direction of arrows, while showing their sailing schedules in associated text.

Certain constraints (called *grouping constraints*) must be considered in the process. In our naval demonstration domain, one constraint is that ships traveling together as a unit (*task force*) must be shown with a single symbol. Another is that ships in port are shown in a way that depends on how "familiar" the port is to the anticipated viewer. So while we never have the case where some ship in port is shown on a map while others in the same port are shown in a separate table, ships in different ports may well be shown differently.

We must also allow for coordination and cross reference. Text in a graph and in a natural language explanation should try to use similar vocabulary. The text might need to refer to the part of the map it is describing. This requires that the planning for individual media must be given enough advice by the overall planner to assure consistency and coordination.

We are exploring the planning paradigm by developing rules for good presentations and expressing them in the formalism of an AI planner.

2 Knowledge Bases & Rules For Presentation Planning

Presentation planning is achieved in our system by the application of a system of antecedent-consequent rules. The rules are used to classify the information that needs to be presented and to map types of information to appropriate types of presentations. Specifically, rule application involves *realizing* the categories that a given piece of information fits within, i.e., finding the rules whose antecedents describe the information; *selecting* the most germane category for the information, i.e., finding the most specific rules; and *redescribing* the information in appropriate textual and visual forms, i.e., using the consequents of the rules to structure the presentation.

ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS OF POOR QUALITY

We cannot at this point claim that we have a complete theory of what constitutes a good presentation, since such a theory would have to explain aesthetic considerations involved in the preparation of presentations. While we cannot handle such considerations in general, we have been able to provide heuristics useful in certain situations. The Integrated Interfaces system contains rules that structure forms so that they contain what we consider appropriate amounts of information. Users whose aesthetic judgements differ from ours can modify these explicit rules to achieve different behavior. In this sense our system can be considered a presentation *shell*.

2.1 Example

The U.S. Navy's Pacific Fleet prepares a daily report on the situation and plans of the fleet. This report conveys current ship locations, courses, current activities, and the activities planned for ships in the near future. The person putting this situation report together has available for presentation a graphics system for ocean surface maps, a business graphics system for time tables, and methods for adding text to maps and tables.

Such a report could be presented in many ways. A map with lines showing each ship's course with a label at each point where the ship starts a new activity; or a map with points showing each ship's initial location and a timetable for each ship; or a map with points showing each ship's initial location and a label in English explaining its sailing plans. The Pacific Fleet uses the third form.

The Navy's report-generating activities can be described as following a process and rules similar to those encoded in our system. Information concerning ships is realized as belonging to certain known categories, e.g., the ship's planned activities. Rules for translating such information into a component of a report, e.g., an indication on a map or a textual description, are then examined, and a rule appropriate for the desired mode of presentation is selected. The information about the ships is then redescribed as part of the presentation being prepared.

2.2 Design

2.2.1 Models

Our models characterize or define the categories of *entities* our user interface can deal with. One of the models identifies the categories of objects and actions in a common-sense view of the domain of our system. We indicate subclass relations present between categories, as well as relationships between objects and actions. For the Navy application we described above, we include the various categories of ships and sailing activities. We also include specific knowledge, such as that *Tankers* are a type of *Ship*, and that a *Repair* activity involves a *Disabled Ship*.

Another model describes the categories of objects and actions of the interface world. The objects here include windows, tables, maps, text strings, and icons. The actions here include creation, deletion, movement, and structure of displays.

A final model (not crucial for this discussion) describes the functions and data structures of the available application services. Included here are descriptions of underlying application software, and any database schemas.

2.2.2 Rules

The presentation rules are simple: they map object from the application domain model into objects in the interface model. So the entity that describes a daily status report may be mapped into a

map. A position report may be mapped onto a point. A ship's planned future activities may be mapped onto a text string.

These rules are arranged according to the class subsumption hierarchy of the models. So the rules applicable to all ships are further up the hierarchy than those applying only to tankers.

A system that constructs a visual display based entirely on an analysis of the details of the data to be presented (cf. Mackinlay [5]) holds considerable appeal. However, in a domain as complex as ours, it is probably impossible to design such a presentation system. We thus allow both "low-level" rules, such as those that map the various types of ships to their icons, and "high-level" ones, which given a particular type of presentation request provide a script to be followed in fulfilling the request.

2.2.3 Rule Application

Presentation planning can now be described as the task of recognizing the domain categories within which a request for information presentation falls, selection of the appropriate rules that apply to those categories, and mapping of the domain terms in the request into appropriate presentation terms.

The three phases which we refer to as *realization*, *selection*, and *redescription* are implemented in our system as described below.

Realization relates the facts about instances to the abstract categories of the model. For example, the concrete facts about *Sprite*, a ship with a malfunctioning radar, must lead to the realization that it is a *Disabled Ship* (assuming *Disabled Ship* is defined in the domain model). Selection works by allowing for the appropriate mapping rules to be chosen, allowing for additivity. Selection also assures that all aspects of the demand for presentation are met by some rule. Redescription applies the rules, mapping each aspect of a common-sense view of a presentation into an equivalent presentation form.

The forms produced by rule application are not actually the commands to the output subsystems (i.e., the map graphics system, text generator, and the business forms system). Instead, they are interpretable by *device drivers* that control these systems. This design allows the forms produced by the rules to serve as a model for the contents of the screen. Although we do not currently do so, user input activity on the screen could be interpreted with this screen model serving as a context. So our design has the additional advantage of allowing, in principle, the use of the same knowledge base and many of the same inference mechanisms for analysis and presentation planning.

3 Knowledge Representation Tools

Our implementation of presentation planning depends on two knowledge representation systems: NIKL and KL-TWO. NIKL holds our models. KL-TWO automatically carries out realization. KL-TWO also holds the demands for presentation and receives the forms read by the device drivers. This section provides a brief introduction to these tools.

3.1 NIKL

NIKL [3] is a network knowledge-base system descended from KL-ONE [1]. This type of system supports description of the categories of entities that make up a domain. The central components

**ORIGINAL PAGE IS
OF POOR QUALITY**

ORIGINAL PAGE IS
OF POOR QUALITY

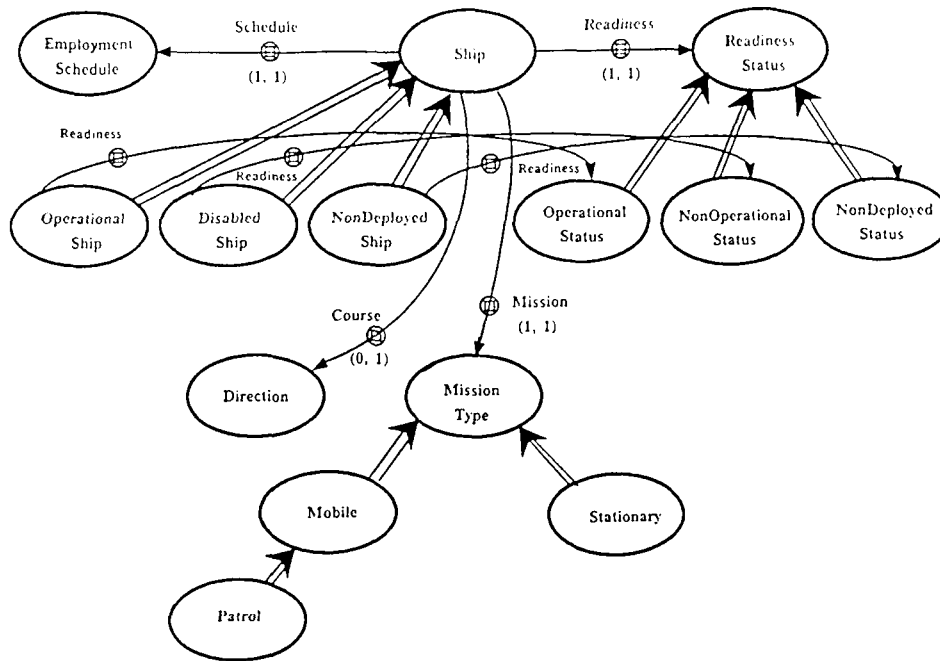


Figure 1: Fragment of Domain Model Containing *Ship*.

of the notation are sets of concepts and roles, organized in IS-A hierarchies. These hierarchies identify when membership in one category (or the holding of one relationship) entails membership in (or the holding of) another. The roles are associated with concepts (as *role restrictions*), and identify the relationships that can hold between actual individuals that belong to the categories. The role restrictions can also hold number restrictions on the number of entities that can fill these roles.

We have been experimenting with a naval assets domain model for the naval briefing application mentioned above. It has a concept *Disabled-Ship* that is meant to identify the ships that are unable to carry out their missions. *Disabled-Ship* IS-A type of *Ship* distinguished from *Ship* by having a role restriction *Readiness* that relates *Disabled-Ship* to *NonOperational-Status*, i.e., all ships with nonoperational status are disabled. All *Ships* can have exactly one filler of the *Readiness* role restriction. The concept of *NonOperational-Status* is partly defined through the IS-A relation to a concept *Readiness-Status*. This situation is shown graphically in Figure 1 in the typical network notation used for KL-ONE knowledge bases.

In flavor, NIKL is a frame system, with the concepts equivalent to frames and the role restrictions to slots. However, the NIKL representation can be given a formal semantics. In fact, we could translate our NIKL knowledge bases into predicate calculus expressions and use a theorem prover to make the same inferences we do. However, NIKL is optimized for the limited inferences it makes and a general purpose theorem prover would be less efficient.

3.2 KL-TWO

KL-TWO is a hybrid knowledge representation system that takes advantage of NIKL's formal semantics [8]. KL-TWO links another reasoner, PENNI, to NIKL. For our purposes, PENNI, which

is an enhanced version of RUP [6], can be viewed as restricted to reasoning using propositional logic. As such, PENNI is more restricted than those systems that use first order logic and a general purpose theorem prover.

PENNI can be viewed as managing a data base of propositions of the form $(P a)$ and $(Q a b)$ where the forms are variable free. The first item in each ordered pair is the name of a concept in an associated NIKL network and the first item in each ordered triple is the name of a role in that network. So the assertion of any form $(P a)$ is a statement that the individual a is a kind of thing described by the concept P . The assertion $(Q a b)$ states that the individuals a and b are related by the abstract relation described by Q .

NIKL adds to PENNI the ability to do taxonomic reasoning. Assume the NIKL database contains the concepts just described in discussing NIKL. Assume that we assert just the following three facts: $(Ship\ Sprite)$, $(Readiness\ Sprite\ C4)$ and $(NonOperational-Status\ C4)$; $C4$ is a U.S. Navy readiness code. Using the knowledge base, PENNI is able to deduce that any *Ship* whose *Readiness* is a *NonOperational-Status* is a *Disabled-Ship*. So if we ask if $(Disabled-Ship\ Sprite)$ is true, KL-TWO will reply positively.

PENNI also provides a truth maintenance system that keeps track of the facts used to deduce others. When our rules are used to determine aspects of a presentation from facts about the world, the truth maintenance system records the dependencies between the domain and the presentation. For example, $(Readiness\ Sprite\ C4)$ triggers a rule which asserts $(Disabled-Ship\ Sprite)$. If $(Readiness\ Sprite\ C4)$ is retracted, PENNI's truth maintenance system will automatically retract the assertion that the Sprite is a disabled ship.

4 Examples

The power of Presentation Planning is in its flexibility. The designer of a system does not specify rigidly in advance in what form information will be requested from the user, and how data and results will be displayed. Instead, our models contain descriptions of the types of information the application programs deal with, and of the types of graphical tools and instruments available. The rules for presentation enable the system to generate on-demand displays appropriate for given needs. Here are some concrete examples.

4.1 Construction of a Visual Representation of an Object

Consider the knowledge about ships and about graphical instruments encoded in the NIKL models in Figures 1 and 2. Besides the aspects of Figure 1 already indicated, note that *Ships* have *Missions* and that *Patrol* missions are a subclass of *Mobile* missions. Note also that all *Ships* have *Schedules*. Figure 2 describes some *Graphical-Instruments*. This includes *Text* for language output, *Icons* for maps and isolated forms, and *Visual-Enhancements* that could apply to icons and text. *Icons* have *Text* as their *Tag*. Several specific *Icons* and *Visual-Enhancements* are included.

Let us assume that the user wishes to show ships engaged in a *Mobile* mission with a special *Icon*, and that the icon should be oriented in a direction identical to the ship's course. In addition, assume that *Disabled-Ships* are to be shown with *Red* icons and that the *Schedule* of a ship is to be shown in the natural language *Tag* of the *Icon* representing it. A version of the rules that we would use to achieve this is shown in Figure 3.

ORIGINAL PAGE IS
OF POOR QUALITY

**ORIGINAL PAGE IS
OF POOR QUALITY**

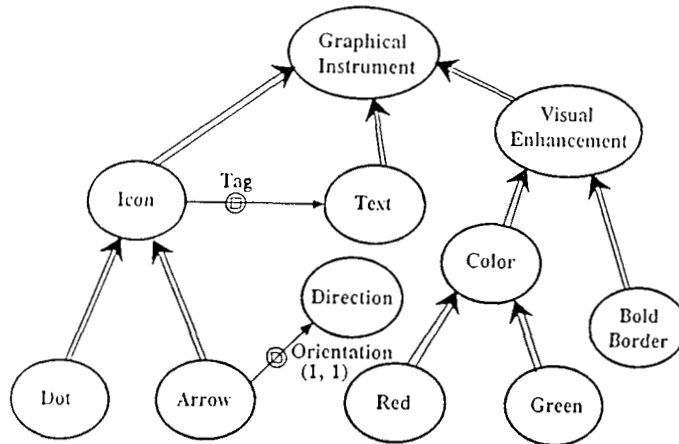


Figure 2: Fragment of Interface Model Containing *Graphical-Instrument*.

-
1. IF (Operational-Ship x) or (NonDeployed-Ship x)
THEN (Icon-Color Image(x) Green)
 2. IF (Disabled-Ship x)
THEN (Icon-Color Image(x) Red)
 3. IF (Ship x) and (Mission x y) and (Course y z)
THEN (Orientation Image(x) z)
 4. IF (Ship x) and (Mission x y) and
(Mobile y) THEN (Icon-Type Image(x) Arrow)
 5. IF (Ship x) and (Schedule x y)
THEN (Tag Image(x) Textual-Description(y))
-

Figure 3: Sample Presentation Rules

The antecedent considers the categories of one or more individuals and their relationships, all in terms of the NIKL models. The consequents provide assertions about the graphic representation of objects for the PENNI database. These rules are asserted into PENNI so that the truth maintenance system may keep track of the dependencies between antecedent facts and their resultant consequents, as explained in the previous section.

The functions *Image* and *Textual-Description* map the constants of the common sense world into constants of the visual and textual world, respectively. For example, Rule 5 above states that if some individual, *x*, is a *Ship* and another individual, *y*, is its *Schedule*, then the *Tag* of the image of *x* is the textual-description of *y*. The textual-description of *y* will be created by the invocation of our text generator.

To complete the example, suppose that the following set of facts was asserted into the PENNI database: (*Ship Sprite*), (*Readiness Sprite C4*), (*NonOperational-Status C4*), (*Mission Sprite X97*), (*Patrol X97*), (*Schedule Sprite U46*), (*Course X97 220*), and (*Employment-Schedule U46*). Suppose further that the NIKL model defined *Patrol* to be a subclass of *Mobile* missions. Then realization would recognize the 'Sprite' as a *Disabled Ship* and one engaged in a *Mobile* mission on a course of 220 degrees. Selection would identify that Rules 2, 3, 4 and 5 apply. Redescription would result in the addition to the PENNI database of the description of the image of the 'Sprite' as a red arrow with an orientation of 220, and with a textual representation of its schedule as its label.

If any of the facts pertaining to *Sprite* is retracted, an automatic change in the description of its graphic image will occur.

4.2 Recognizing Special Cases

For many requests for information encountered in our domain the presentation required is far more complex than the rules of the kind listed above could provide for. The construction of these complex presentations requires, among other things, a global evaluation of the coherence of the display. It would therefore be hopeless, at this point, to attempt to write rules that would attempt to derive an elaborate presentation entirely from low-level information about the objects to be described. Our approach provides us with a partial solution to this problem.

The availability of models of the domain and of displays to our Presentation Planner gives it the advantage of being able to recognize collections of data as representing information of a certain known type. The Presentation Planner can then make use of presentation techniques specialized for this type of data to provide the user with more appropriate displays.

For example, Figure 4 provides portions of our model that include the class *Pacific Situation*, a display of data about ships and ports in the *Pacific Region*, which includes certain specific information from the ships' employment schedule.

When provided with data about ships in the Pacific region and their employments, the Presentation Planner would classify the data in its model of the domain. A spatial reasoner deduces the region containing all of the ships which would be included in the *Pacific Region* and the Presentation Planner recognizing that it has received a collection of data belonging to the class *Pacific Situation*. Once the classification of the data is accomplished the Presentation Planner will use specific presentation rules appropriate for displaying the information. In the domain we have considered there is a preferred way for presenting this information, to which we try to conform. This preferred presentation has developed in the Navy in the course of years of handcrafted situation briefing presentations.

ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

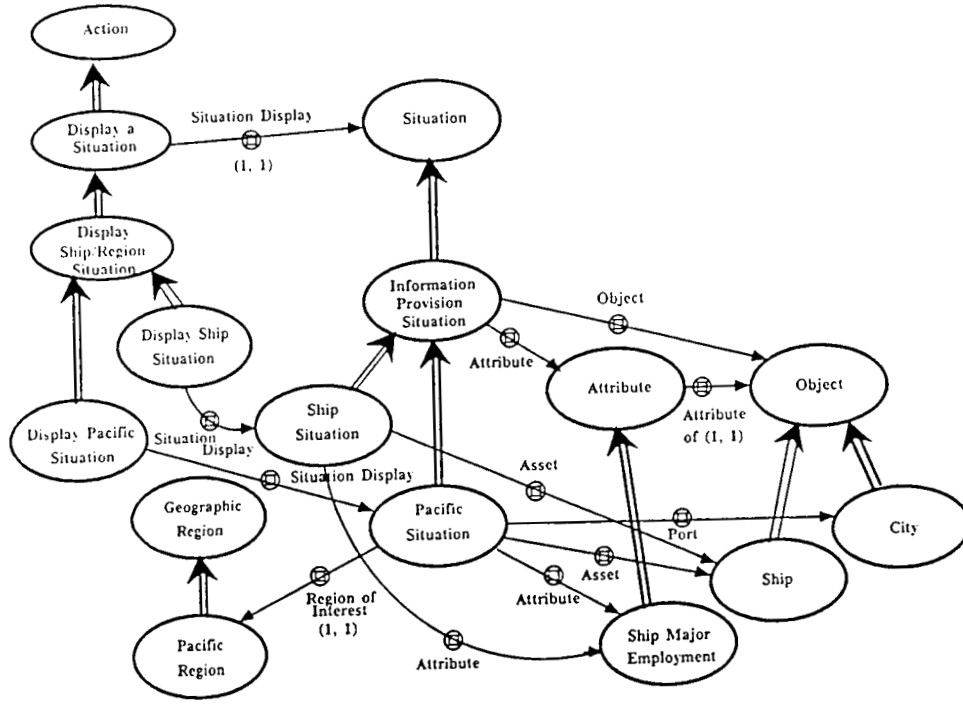


Figure 4: Fragment of Domain Model Including *Situation*.

The specific presentation rules appropriate only for a situation briefing will combine the entities created by more general rules, of the kind described in the previous section, to produce the final presentation.

4.3 Generation of an Input Display

The Presentation Planner must also deal with the preparation of displays for the purpose of soliciting necessary information from the user. Here, again, the models of all aspects of the task and the domain are indispensable.

At some point the user may indicate a desire to view data concerning ships in some region. In terms of our model (see Figure 4), that would mean indicating a preference for *Display a Situation*. As it turns out, the Presentation Planner does not have any rules that can be used to redescribe this general request into a presentation, but there exist ways of satisfying more specific requests. For example, requests to have the *Pacific Region* or any of its subregions displayed can be satisfied. As we see in Figure 4, the situation involves specific ships and ports, which may also be displayed.

In this case, the Presentation Planner collects all options the user can choose among to construct an executable request. The Presentation Planner then plans a display form that will be used to present these options to the user. The result of this plan is a set of assertions in PENNI that the device driver for a separate form management package (QFORMS) [2] will use to prepare the input form.

The form below, presented to the user, allows the user to make one of several specific choices:

Pacific Regions:

Western Pacific	<input type="checkbox"/>
South China Sea	<input type="checkbox"/>
Indian Ocean	<input type="checkbox"/>
Eastern Pacific	<input type="checkbox"/>
Pacific Command Region	<input type="checkbox"/>

Ship: _____

It is instructive to examine precisely how this form is created. Specifically, how does the choice "Ship" become part of the form? It is not a Pacific Region, but Navy personnel request that this possibility be supported.

We thus included in our model the concept *Display Ship/Region Situation*. Since this has two subclasses of actions, namely *Display Ship Situation* and *Display Regional Situation* our system considers the possibility of generating an intermediate two item submenu, something like:

Situation in Pacific Region	<input type="checkbox"/>
Situation of Ship	<input type="checkbox"/>

We considered this unsatisfactory from a human factors standpoint. We therefore formulated a general rule saying that if the choices on a proposed menu can be further subdivided, and if the number of choices is less than N , then the proposed menu should not be displayed. Instead, a more detailed form should be generated, one based on the subchoices. Our prototype uses the value 3 for N , so in this case the rule causes the Presentation Planner to immediately generate the more specific form. A user is free to change the value of N , thus modifying the design of forms the system generates in situations like the one above.

ORIGINAL PAGE IS OF POOR QUALITY

Note that the geographic regions available were specified by name in the form created, while ships were not. Rather, the user is allowed to specify the desired ship by typing it on the form (Figure 5). This distinction is a result of information concerning the cardinality of the relevant collections of objects – information encoded in our models. Since the number of possible choices for region is small, they are enumerated. However, the number of ships is larger, so the user is provided with a way to specify a choice explicitly instead.

Generating interfaces by models and rules is time consuming and tedious. But it forces the designers to think out every aspect of an interface. The decisions are not hidden in the code, they are explicit – observable, modifiable – in the rules and the model.

5 Related Work

The literature contains numerous examples of User Interface Management Systems. However, we see our contribution as being our emphasis on *Presentation Planning*, and very few systems are concerned with this aspect of the interface. Perhaps the best known previous work dealing with this issue is that of Mackinlay [5].

Much like part of our system, Mackinlay's *APT* uses information about characteristics of data provided to it, to produce a graphical representation of that data. The differences between the two systems become clear when we consider the variety of data each deals with and the variety of presentations they produce. *APT* produces graphs of various kinds, and much of its effort goes into deciding which axes to choose, and how to indicate the values along each axis. Data dealt with is limited to what can be presented using such graphs. Consequently, Mackinlay has succeeded in producing a system which can generate graphical presentations automatically using only "low-level" information about the objects and their attributes.

Our system is expected to generate a much wider variety of displays, many that would require considerable design work even from an expert human graphic artist.¹ In addition, certain display layouts are often chosen simply to conform to pre-existing preferences of Navy personnel. Consequently, unlike Mackinlay, we are required to provide for the possibility of following pre-set stereotypical instructions in certain cases. We thus must devote considerable effort to recognizing which cases require these special displays.

A further significant difference between the systems is the complexity of the data we are required to present. In order to handle this range of data we must represent it using a sophisticated knowledge representation language, NIKL, a facility which Mackinlay finds unnecessary in *APT*. Both systems make use of sophisticated reasoning facilities.

6 Problems

We believe our approach to the problem of presentation planning is a viable one. Indeed, as illustrated in the examples of the previous section, we are using it to generate various interesting displays. However, there are still numerous outstanding problems which remain to be solved. In this section we will list some of the more difficult and interesting ones; some of them are inherent to presentation planning while others are specific to choices we have made in our system.

¹As in fact they do. Maps of the kind produced by our system take Navy personnel approximately 4 hours to produce every day.

6.1 Lack of Coordination between Output Modes

Currently, we are using off-the-shelf programs for the low level production of output. This places us in a position of having to divide our data between the available facilities without having access to the internal decisions made by those facilities. In reality linguistic considerations may play an important part in the decision to use a pointing gesture. For example, in a situation where using language to describe an object to the hearer would be difficult or awkward, we prefer to point to it. Our existing setup does not permit the Presentation Planner to become aware of such difficulties.

The proper solution to this problem would probably require a uniform approach to all methods of communication and a more complete understanding of their relative capabilities. This appears to be a hard problem. We are not aware of any existing efforts in this direction.

Related to the problem mentioned above is the question, Which information about presentation planning can be shared across media and modalities and which is unique to each medium?

6.2 Modeling Difficulties

The domain of graphical displays is not yet well understood. We are facing difficulties in developing a model that expresses all the information needed to plan presentations. Certain idiosyncracies of NIKL have added to the difficulty of representing some of the knowledge. Several of these problems will be resolved with the development of Loom [4] to which we intend to switch as soon as it becomes available.

6.3 Lack of a User Model

A user model will enhance the Presentation Planner. For example, knowledge about a user's familiarity with a certain geographic area will allow the system to label only unfamiliar ports and regions, thus reducing screen clutter. While incorporating a user model is in our longer range plans, we have not yet begun to do so.

6.4 Lack of a Dialogue Model

A dialogue model will allow the presentations to be more closely tailored to the user's requests. Currently, the Presentation Planner is simply provided with data to display. It is not aware of the purpose of the display, nor even of the user request that prompted it. Keeping track of such information is also in our future plans.

6.5 Complexity of Correspondence between Domain and Interface Models

Many of our presentation rules assume a simple correspondence between domain objects and their graphical icons. This may turn out to be an oversimplification. It might be necessary for us to posit intermediate levels between the domain and the display; a common-sense reasoning level, for example.

7 Current Status

A demonstration version of the Integrated Interfaces system is now available at ISI. The current version models the domain of Navy ships in the Pacific Ocean. A user may use the system to access

**ORIGINAL PAGE IS
OF POOR QUALITY**

ORIGINAL PAGE IS OF POOR QUALITY

information about ships' locations, tasks, readiness status, and more. The resulting information is displayed using combinations of maps, menus, tables, and natural language output (Figure 5).

The system is written in Common Lisp and runs in the X windows environment under UNIX on HP 9000 Model 350 workstations. Displays are presented on a Renaissance color graphics monitor. The map graphic modality is supported by ISI's Graphics Display Agent. Menus and forms are created using QFORMS [2]. Natural language output is produced by ISI's Penman system [7].

8 Acknowledgements

We wish to acknowledge the crucial help provided by others working on the Integrated Interface project. Stu Shapiro helped in developing the general framework of the system and contributes to its implementation. Paul Raveling has developed the graphical interface and continues to maintain the GDA. Chin Chee has ported QFORMS and Penman to the IIP workstation and is responsible for coordinating the various parts of the system. Jim Geller has contributed to the implementation of the spatial reasoner.

9 References

- [1] Brachman, Ronald J. and James G. Schmolze, "An Overview of the KL-ONE Knowledge Representation System," *Cognitive Science* 9(2), 1985, pp. 171-216.
- [2] Kaczmarek, Tom, "CUE Forms Description," ISI Internal Report, USC/ISI, Marina del Rey, CA., 1984.
- [3] Kaczmarek, Tom, Ray Bates and Gabriel Robins, "Recent Developments in NIKL," *Proceedings, AAAI '86*, Philadelphia, PA., August, 1986.
- [4] MacGregor, Robert and Ray Bates, "The Loom Knowledge Representation Language," *Proceedings of the Knowledge-Based Systems Workshop*, St. Louis, MO., April, 1987. (Also available as ISI Reprint Series ISI/RS-87-188.)
- [5] Mackinlay, Jock D. "Automatic Design of Graphical Presentations," Ph.D. Thesis, Department of Computer Science, Stanford University, Stanford, CA., December, 1986.
- [6] McAllester, D. A., "Reasoning Utility Package User's Manual," Massachusetts Institute of Technology, AI Memo 667, Cambridge, MA., April, 1982.
- [7] Sondheimer, Norman K. and Bernhard Nebel, "A Logical-Form and Knowledge-Base Design For Natural Language Generation," *Proceedings, AAAI '86*, Philadelphia, PA., August, 1986, pp. 612-618.
- [8] Vilain, M., "The Restricted Language Architecture of a Hybrid Representation System," *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, Los Angeles, CA., August, 1985, pp. 547-551.
- [9] Walker, E., R. Weischedel and N. Sondheimer, "Natural Language Interface Technology," *Proceedings of the Strategic Systems Symposium*, Monterey, CA., October, 1985.

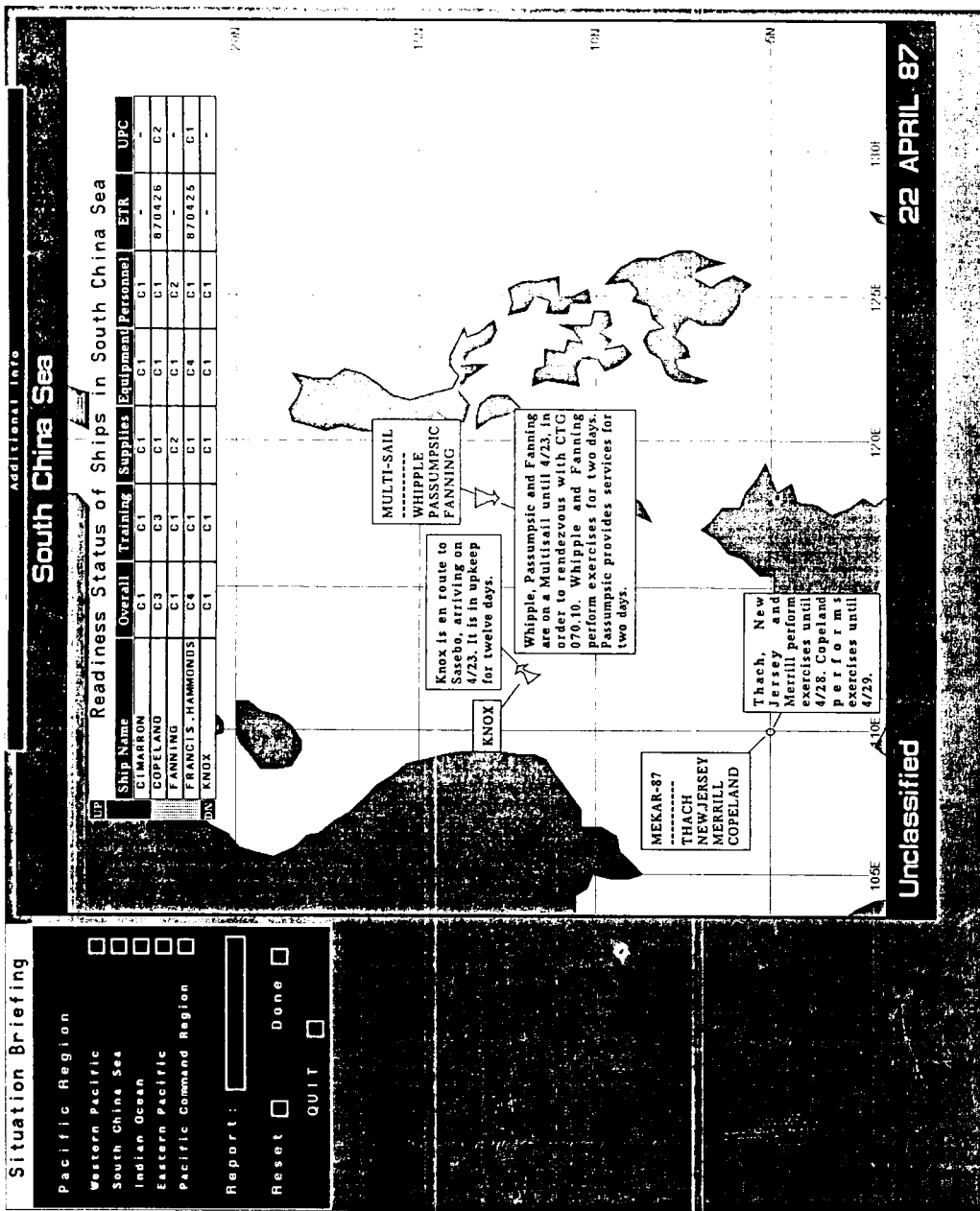


Figure 5: South China Sea Situation Display.