AD-A106 801 BOLT BERANEK AND NEWMAN INC CAMBRIDGE MA F/G 15/4

USING AI TECHNIQUES FOR THREAT DISPLAY AND PROJECTION, INCLUDIN--ETC(U)

SEP 81 J VITTAL, R BOBROW, O SELFRIDGE N00039-80-C-0659

UNCLASSIFIED BBN-4789 NL

1 of 2

AD A106801

Bolt Beranek and Newman Inc.

Report No. 4789

Using AI Techniques for Threat Display and Projection, Including Tactical Deception Indication

J. Vittal, R. Bobrow, and O. Selfridge

September 1981

Prepared for:
NAVELEX

Report No. 4789

USING AI TECHNIQUES FOR THREAT DISPLAY AND PROJECTION, INCLUDING TACTICAL DECEPTION INDICATION

J. Vittal, R. Bobrow, O. Selfridge

30 Sep [redacted] 81

Final Report

Contract No. N00039-80-C-0659

Prepared by:

Bolt Beranek and Newman Inc.
10 Moulton Street
Cambridge, Massachusetts 02238

Prepared for:

Naval Electronic Systems Command
ELEX 6133
Washington, D.C. 20360
Attn: Robert Kahane

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. AD-A106 801	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Using AI Techniques for Threat Display and Projection, Including Tactical Deception Indication		5. TYPE OF REPORT & PERIOD COVERED Final Report
		6. PERFORMING ORG. REPORT NUMBER BBN Report No. 4789
7. AUTHOR(s) J. Vittal, O. Selfridge, R. Bobrow		8. CONTRACT OR GRANT NUMBER(s) N00039-80-C-0659
9. PERFORMING ORGANIZATION NAME AND ADDRESS Bolt Beranek and Newman Inc. 10 Moulton Street Cambridge, MA 02238		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Electronic Systems Command ELEX 6133 Washington, D.C. 20360		12. REPORT DATE 30 September 81
		13. NUMBER OF PAGES 128
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

threat display, tactical deception indication, knowledge-based systems simulation, command and control, data fusion, knowledge representation, artificial intelligence, sensor integration

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

Good decision making is most needed in Naval command centers at times of crisis when personnel are easily overwhelmed by the amount of data flowing into the center. Knowledge-based systems which are tuned to look for certain situations and make inferences can help the action officer make decisions. Such systems can be built using current techniques in artificial intelligence.

OVER

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

This report covers the design of a computer-based system for threat display and projection (TDP) including tactical deception indication (TDI). It discusses AI techniques which are applicable to the development of such a system, such as knowledge representation languages and knowledge-based simulation. It also presents an operationally relevant scenario showing some interaction with a system which can perform TDP tasks.

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

The last decade has seen remarkable technological and operational changes in sea warfare; targets move faster, sensors increase in power and range, and offensive weapons are harder to detect and carry more punch. Operational personnel can easily be confused by the flood of data that pours into the CIC in times of crisis, so that good decision-making is hardest just when it is most needed and most urgent. Soviet tactical deception techniques, for example, have been developed to take advantage of this indecision and to increase the odds that decisions will be made that could endanger the operations.

The system designed and demonstrated by this contract is intended to demonstrate the utility of Artificial Intelligence and Knowledge Representation in C3 operations by running an operationally convincing scenario involving a situation where early detection of the enemy's tactical deception is essential to the operational decisions. It is expected to perform threat display and projection (TDP) tasks, including Tactical Deception Indication (TDI), aimed at providing the commanding officer with tactical warning of possible deceptive operations against him. The capabilities provided will not only facilitate the decision-making, but also augment the reasoning that can be employed to support effective decision-making.

This document covers part of the design and implementation of a computer based system for Threat Display and Projection with Tactical Deception Indication -- the design and building of certain sensor files, history files, and files representing the purposes and intentions of the participants; the notions of knowledge-based simulation; and the development of a scenario and a breadboard demonstration of a system satisfying that scenario.

Much of this report is recognizable from the initial design document for the TDP/TDI system [5]. However, as we have expanded our understanding of the problem being addressed, the resulting design of the system has also changed. These changes are reflected in the design section which follows, which is a modification of the design section of that report.

The nature of tactical deception is discussed in Section 2.3, and other background information is presented elsewhere in Section 2.

The TDP/TDI system will integrate several computer science technologies, spanning the range from Artificial Intelligence and knowledge representation techniques to interactive graphics. An explanation of the relevant technologies to the implementation of the system, some of which are represented by ongoing projects, is given in Section 3.

The system itself is composed of procedural sub-systems which will:

- Assess the potential threat of a given situation,
- Provide descriptions of likely future situations,
- Provide graphic and tabular displays of actual and projected situations,
- Integrate sensor information and long-term information to provide the user with tactical deception indicators, and
- Give users the capability to control the activities of the other sub-systems.

A more complete description of these procedural sub-systems, the design of the data bases, and some of the user-interaction options for the TDP/TDI system are presented in Section 4. This provides the framework for the TDP/TDI system.

The use of the knowledge representation language KL-ONE as a tool for the implementation of the system is described in Section 5.

A scenario was developed to show an example of deception and the kind of system we envision which will perform TDP tasks and recognize and inform the commander of some attempts of the enemy to deceive. This scenario and the facilities required to build a demonstration system of it (including a videotaping of that demonstration) are described in Section 6.

Issues of compatibility between this TDP/TDI demonstration system and other Navy supported projects are critical to the acceptability of the design, the demonstration system, and perhaps an eventual production system. How this research fits into the Navy programs is described in Section 7.

This report concludes with a summary of a few short-term recommendations for future work in Section 8.

2. BACKGROUND

2.1 Combat Information Centers

It is in the nature of naval warfare that command centers are tasked most heavily, have the highest communication load, and work under the highest stresses of all kinds, just when the need is maximum for carefully considered, urgent, and timely decisions. This is especially true now; it will be even more so in the coming decade as we see technological advances in weapons, platforms, and communications systems, and as sophisticated deception techniques become widely adopted and practiced in potentially hostile naval forces.

The decision makers in the CIC make two basic kinds of decisions, the first dealing with the interpretation of data and intelligence, the second with the choice of actions to be taken. For example, threats must be evaluated and categorized; and then, perhaps, the decision to fire must be made. These decisions are never easy; they require their makers to deal with large quantities of information at different levels of generality, to make projections of future courses of action, to suggest plans and procedures, and to infer intentions from extremely inadequate data. The ambiguities of the rules of engagement, operational orders, and so on, also have heavy impacts on the tactical decisions.

With the advent of reliable high capacity satellite links and other forms of reliable communication, the role of shore stations is increasing. Admirals ashore have speedier access to data from more sources, including the intelligence processed at the national centers, like SOSUS. It is tempting to conclude that command decisions ought therefore to be made ashore as a general rule; no doubt there are times when they should be. Nevertheless, there will always be times when the responsibility and viewpoint of the command center afloat must be paramount.

2.2 Symbolic processing and C3 Systems

Much of the information processing that goes on in a command center is symbolic in nature; and when people handle it they tend to use natural language. The rising capabilities of computers in handling symbolic processing in richer ways, with greater speed, and interacting with users with far better displays and input devices mean that computers can begin to alleviate the difficulties of C3 systems in the next decade. The key developments are:

- New micro-hardware tuned to symbolic processing (often implementing a dialect of Lisp) which is commercially available.
- Bit-map displays with intelligent microprocessors that can handle much of the display processing locally, including interpretation of many of the user commands.
- Flexible, powerful, and very high capacity communication techniques, usually packet switched networks.
- Powerful new techniques in distributed processing and distributed databases.
- Software capable of representing hierarchically linked conceptual structures; much of this work is subsumed under the domain of Artificial Intelligence.

Many of such developments are directed at ease of programming. That is of course important, but for the operational users ease of use, reliability, and speed of response must be the crucial factors. Programming should not be delegated to operational personnel under almost any circumstance; any system should, of course, be responsive to individual differences among the users.

2.3 The anatomy of deception

The essence of deception, as we have seen it, is to impart a false notion of intentions or fact. This section discusses the practice of deception, especially by the Soviets, and specifies the dimensions that we are preparing to handle.

Note that usually the concept deception excludes mere concealment, like a smoke screen, unless it is also intended to impart a specific false impression. Nevertheless, some aspects of concealment must be considered because concealment, like jamming, may be used in contexts where deception is appropriate. This will not be the central issue, however.

Intentions must usually be inferred from evidence, rather than read directly, even if the COMINT is that reliable. The evidence consists of input from sensors, both timely and long past, visual sightings, both surface and overhead, and other derived data in our data bases. In this section we consider first the individual forms of evidence, and how they can be corrupted by deception techniques, and the interaction of different forms of information, like sensors and text.

2.3.1 Sensors

This section considers the active and passive sensors used by US forces in gathering information on enemy platforms, and how they can in principle be used by an enemy with deception techniques. The actual practice, as described bv various sources, will be considered elsewhere. The sensors include:

- Sonar: active
- Sonar: passive
- Radar
- ELINT: this is really a form of passive radar, bv analogy with sonar above
- COMINT
- Visual, both surface and overhead

Other information that contributes may include

- Platform capabilities
- Platform histories
- Platform personnel histories
- Environmental situation, including history: e.g., cartographic information, weather

These all influence the judgments made about deception. The next sections will discuss the sensors, the information thev provide, and the ways that information can contribute to deception.

2.3.1.1 Radar

It is easier to enhance a radar echo than to diminish it, at least in tactical operations. At the end of WWII there was public discussion of certain radar absorbing materials that had been built by both the Axis and the Allies; recently the Stealth program has revived such discussions. But it seems clear that to apply them or to remove them from platforms under tactical conditions would be intimidatingly difficult.

To enhance an echo, one may use passive or active means. A typical passive means is the corner reflector, which returns a speciously large echo in the direction the radar signal came from. The effective gain depends on the square of the reflector aperture diameter in wave lengths. Such devices are used, for example, in small fiberglass pleasure boats when they travel in shipping lanes. The use of such devices in Soviet exercises is discussed elsewhere. At the wavelengths used for US search radars, a corner reflector that would increase the apparent echo of a surface vessel by 10 db need not be a formidable device. At least for use by ground forces, such devices have been observed in operation; they can be erected and dismantled within minutes.

Active devices can be much smaller, obviously, and consist of transmitters that are tuned to the radar frequency; they deliver pulses of the right shape broadcast more or less in all directions. Past devices sometimes merely echoed a received pulse, but that technique has the failing that it transmits with a lag of some microseconds. An alert radar operator will detect such a magnified pulse as being not genuine. US radars generally have a very accurate pulse repetition frequency (prf) clock, and it is fairly easy to synchronize with the pulses, overlaying larger echoes. This also requires a constant radar frequency, which is the usual case. Such a scheme can be beaten, of course, by either changing the pulse interval or by changing the radar frequency; neither is particularly difficult technologically, but they are little practiced in US tactical operations.

2.3.1.2 ELINT

This section discusses the imitation of specific radar characteristics in order to convey the impression that a certain set is where it is not. Radar sets are characterized by certain parameters, some of which are under the control of the operators, and some of which are not:

- Electromagnetic frequency
- Pulse width
- Pulse repetition frequency
- Antenna rotation frequency
- Signal polarization

Those are the standard characteristics. There are, however, a number of other ones, less easily quantified, but far more valuable for individual set recognition.

- Pulse shape
- Frequency stability
- Antenna pattern

Each of those may provide the attentive ELINT operator with clues about the individual radar set. Very often different sets differ in the details of the pulse shape, in both amplitude and frequency. Such clues may be very valuable, but they can be copied as well, and it would be unrealistic to suppose that the Soviets are not aware of them.

It is our supposition here that the Soviet radar technology is not quite up to being able to duplicate radar set characteristics exactly enough to deceive an alert operator with properly functioning ELINT equipment. This is not to say that the ELINT capabilities are as powerful as they ought to be. Furthermore, the extent to which the US is keeping track of the frequency stability of Soviet radars, for example, is not yet clear to us. We intend to explore such points further.

2.3.1.3 Sonar: active

Much the same considerations hold for active sonar as for active radar, except that the possibilities for sound absorbing material are substantially less; that is because the hulls of platforms in the ocean must have a structural rigidity that almost guarantees good sound reflectivity. In any case, sound absorbing materials seem to have never been tried, at least on vessel hulls. Similarly, although sonar jamming techniques are more technologically feasible than that, there is little history of their being experimented with. One can imagine a torpedo with an active sonar responder; that is far more feasible than with radar, because the slow speed of sound allows ample electronic time for amplified responses. The question of whether such devices have been developed or used will be discussed elsewhere.

2.3.1.4 Sonar: passive

There has been a certain amount of publicity about the listening arrays (SOSUS) that the US maintains; it is merely the technical details about operation and performance that are classified. It seems clear that submarines traveling at speed, whether submerged or on the surface, can be detected and tracked from very long distances. Surface vessels are similarly audible.

One difficulty with deception techniques is that the amount of sound radiated by a vessel moving at speed is very large. At certain frequencies the attenuation in the ocean is very low, because there is a funneling of the sound caused by thermoclines; so that the sound attenuates according to a roughly inverse linear law. That means that vessels may be detectable at ranges of thousands of miles. The frequencies used for such long distance tracking are generally below 150 Hz.

The energy radiated by a vessel comes from the engines and the propellers, and is concentrated in very narrow frequency bands: usually multiples of the rotation frequencies of the main shafts. It is perfectly feasible to construct sound projectors at such frequencies, with the requisite power levels. It would be harder to ensure that the sidebands and other characteristics matched those of real vessels; however, since such devices have never been used operationally, it is not clear that our operators could use such characteristics to distinguish them from real vessels without a great deal of operational practice.

The detailed parameters provided by the underwater arrays include primarily frequencies and amplitudes of the energies. The ubiquitous multipath found in oceans means that there is little phase constancy between different frequencies; but the frequencies themselves may be extraordinarily steady. Indeed, a slight change in frequency usually indicates a change in the vessel's course, which imparts a new Doppler component to the received signal.

It would be an obvious deceptive technique to alter engine speeds slightly to suggest a change in course; or, if there is a change in course, also to alter them so as to deny listening system information about the amount of change.

2.3.2 COMINT

Communications Intelligence personnel are not usual on small vessels; in task groups, they often provide valuable information.

Communications cannot be expected to be in the clear, though it often is. Questions of COMSEC are beyond the scope of this document, but it is worthwhile pointing out that one may sometimes infer much without being able to read the traffic. For example, we may know that a submarine may not launch its missiles without approval; that means that lack of communication traffic of a suitable range can sssure us that a missile attack is not imminent.

The possibilities for deception, here, are obviously very great.

2.3.3 Visual

While the use of visual decoys is feasible and practiced in tactical operations on land, the size of ocean platforms seems to preclude their wide use at sea. In any case, they have not been observed, save for possible false targets at docks in the Soviet Union. It would not in principle be impossible for a large submarine on the surface to disguise itself to resemble a surface ship of comparable size, but in practice it seems not to have been tried.

Other visual disguising operations are common practice. For example, hull numbers are a common way to identify vessels. Yet, the Soviets often change the hull numbers on vessels, often even using different numbers on each side of the vessel.

There is also a practice to disguise drones as scout planes in order to disguise the possible location of a group, or to send scout planes to a location that is not dictated by normal doctrine. In general, scout planes stay within a 150 mi. range of the group in the forward direction. But, if they are seen to be elsewhere, false conclusions about the possible locations of a specific enemy group could be drawn.

2.4 Intention Structures

Intention structures is a term used here to describe the complex of goals and purposes that governs the decisions of participants in Naval actions. Although intentions are held by people, in particular the command echelons, it is convenient to attribute them to the platforms they inhabit and control.

In the context we deal with here, most of the intentions structures can be treated merely as constraints; like a platform's tendency to take the shortest path to travel from A to B, in order to conserve fuel. But sometimes that constraint must not be taken as absolute: a vessel will not travel at flank speed for long unless either the command perceives an emergency, or it wishes to give the impression that it perceives an emergency. That is, the observation of a vessel's traveling faster than its usual maximum provides information about the intentions.

Similarly, it is usually over-riding for a captain to preserve the integrity of his vessel and the safety of his crew; but that is not necessarily true in tactical engagements.

The intentions that act as constraints can be dealt with that way. Our concern here is with the intention structures that change, and that modify the projected activities of an enemy. That is, the concern is with the intentions that are going to alter the optimum command decisions that our vessels ought to make.

The moving intentions, therefore, are those that derive from the overall goals or missions of a fleet. To be able to interpret these from the observed actions of an enemy is a prime responsibility of a command; that is one reason why timely and responsive intelligence is vital.

The fundamental intentions are the missions assigned to the fleet elements. From them are derived the subordinate missions assigned to the individual platforms, and to the individual commands on each platform. In time of peace, the whole question of indicators and warning is an attempt to infer accurately whether the fundamental mission of a potential enemy is hostility or continued peace. In wartime, that question is still fundamental: in a naval element observed not in engagement with our forces, is its mission a hostile one directed at a particular command?

An exemplary top level intention structure is shown in Figure 1, below.

MISSION	Hostile action against US fleet
SEQU. PLAN	Approach fleet Target, Arm, and Launch Missiles Evade Retaliation, and Withdraw
VESSEL X	Arrive position A by Time T
VESSEL BRIDGE	Course B, Speed S
VESSEL CIC	Ready Armament, Secure
...	
VESSEL Y	Arrive position A' by Time T'
...	

FIGURE 1. TOP OF INTENTION STRUCTURE FOR OFFENSIVE ACTION

Corresponding to the levels in Figure 1, Figure 2 shows the questions that command has to answer. Inferences about the higher level intentions can be made by observing whether the lower level observables are more consistent with the left hand or

MISSION

Offense?	Evasion?
What is plan?	What is plan?
Next destination?	Destination if not attacked?
What armaments deploy?	Destination if attacked?
When deploy them?	If attacked,
	Scatter - continue evade?
	Retaliate?
What course to dest.?	What is course and speed?
What resources?	ETA?
What data on us?	Aware of surveillance?
...	

FIGURE 2. QUESTIONS ABOUT ENEMY INTENTION STRUCTURE

the right hand column. That illustrates the interaction of the TDP with the command decisions, because the projections provide the details of the columns that are matched with the observables. That is, if the intention of the enemy is fundamentally evasive, then the projections of his courses and other actions will differ from those if his intention is fundamentally hostile.

The requirements at each level are either

- to provide outputs for projection calculations
- to provide further breakdown of intentions at lower levels

For our purposes, we assume that the enemy mission at any one time is either of the two shown in Figure 2 -- that is, either offensive or evasive. The point of enemy deception at the highest level, then, is to cause us to believe that is evasive when it is truly offensive, or vice versa.

3. APPLICABLE TECHNOLOGIES

There are several technologies which can be brought to bear on potential solutions to the TDP/TDI problem:

- o Artificial intelligence (AI) techniques, specifically knowledge representation and inferencing
- o Graphical displays and their interfaces
- o Databases
- o Programming languages and their support tools -- programming systems on personal symbolic processors
- o Simulation

A great deal of the work being performed in the latter four areas is at least peripherally related to AI.

Much current work in AI is directed at finding useful and cost-effective applications that take proper advantage of the skills and insights provided. Many of the problems are representational -- how can meanings be captured so as to be manipulated, perhaps symbolically, by the computer? Other aspects that are clearly relevant to the discussion here are:

- o Representing and manipulating symbolic hypotheses
- o Representing judgmental information
- o Combining mathematical and judgmental inferences
- o Collecting, describing and representing expert procedures and reasoning
- o Handling uncertain or probabilistic inferencing

A special problem in representation is how to consider the purposes and intentions of the actors in a scenario, usually the platforms and their commanding officers. AI has considered goals in a few contexts in the past, for example, Brown [11] in his analysis of the hypotheses adopted by students in SOPHIE; and the use of subgoals in theorem proving [24, 30, 37]. There are no current techniques that can deal with the complexities of purpose and intention structures of the actors in the environments being discussed here.

The following sections describe several of the applicable technologies and systems which have been developed, and how they might apply to a system for TDP/TDI.

3.1 Programming systems

3.1.1 Lisp and symbolic processors

There are various dialects of the Lisp language [26, 27] which have been developed primarily as a tool for AI researchers. The primary two dialects are MacLisp and its derivatives [28, 39] and Interlisp [36, 21, 29]. Most Lisp systems go far beyond just being a standard language and compiler.

There are significant differences between Lisp and more standard programming languages. Lisp uses a uniform representation for most data and code. Internal to the system, code can be represented as source which can be interpreted, or compiled; interpreted functions can call compiled functions and vice versa. All Lisp systems have an interpreter; most also have a compiler. The result is that programs can <u>create other programs and evaluate them dynamically</u>. Variables, rather than being lexically scoped, are scoped dynamically.

Lisp systems generally provide a powerful programming environment containing a collection of tools to facilitate the development and debugging of large programs. This is made possible by the above features of the language. The sophistication and scope of the tools for editing, modifying and managing programs, along with the ability for users to modify the actions of specific of the tools without affecting the others, are what make Lisp systems such powerful programming environments.

A recent development is the appearance of a new class of LSI-based, micro-programmed computing hardware: personal symbolic processors. These personal machines are generally capable of running the various dialects of Lisp. All of these machines support very large virtual address spaces (for example, 2**24 words), a requirement for much of the current work in AI. R. Greenblatt and some of his colleagues at MIT developed the MIT Lisp Machine (also known as the CADR), which runs a dialect of MacLisp. This machine is now being marketed by two organizations: Symbolics, Inc. and Lisp Machine, Inc. Xerox PARC has developed two machines, the Dolphin and the Dorado, both of which run Interlisp. The Dolphin is being marketed as the Xerox 1100 Scientific Information Processor by Xerox Electro-Optical Systems. BBN has developed a personal machine runs Interlisp, called the Jericho [16]. The decision whether to market the Jericho has not yet been made.

In terms of Lisp processing speed, most of these processors are roughly comparable to a large, third generation time-sharing mainframe (such as a DECSystem-20/40) running stand-alone. Their purchase price is generally within the range of $50-150K, a small fraction of what would have been expected only a few years ago for such performance.

3.1.2 Object-oriented programming

Object-oriented programming is the basis of such languages and systems as Smalltalk [13, 20, 32] and Simula [2]. Rather than just being an arbitrary set of functions and data structures, objects in the language contain both the definition of the data structures relevant to the object and the allowable processes which can be invoked on that data. Specific objects are actually instances of a class of objects. Each class consists of a name, its parents which form a class hierarchy for purposes of inheritance of both data and procedures, a set of local instance variables, and a set of locally defined operator definitions.

Object-oriented programming can be used as a tool for simulation. An important feature of a simulation language such as SIMULA is to provide for the declarative description of simulation objects and classes of simulation objects. This declarative, object-oriented treatment gives greater power for decomposing and describing the internal state of a complex model, makes explicit the object structure on which model procedures are based, and provides a mechanism for linking model procedures with (and limiting their application to) the objects for which they are relevant.

One of the primary questions one must ask is when to choose object-oriented programming over conventional programming. During the design process, the data structures and the operations which will be performed on them are defined. If there is a significant amount of operator overlap, then naming conventions indicate that object-oriented programming is more efficient and understandable. Hierarchical inheritance allows for the definition of the operations on global kinds of objects exactly once.

The Learning Research Group at Xerox PARC [23] and Jim Schmolze [31] provide much more detailed discussions into the general notions of object-oriented programming and the guidelines for using such a style.

3.2 Pattern-directed programming and Rule-based systems

A focus of interest in AI research in the past few years has been a program organization based on data- or event-driven operations. Data-driven programs respond directly to a wide range of possibly unanticipated data and events, rather than simply using prespecified and inflexible control structures to perform operations on a range of expected data represented in known formats. Such an organization can be considered to be pattern-directed: rather than code deciding what data to access and manipulate, changes in the data determine what pieces of code are relevant to run.

One type of a pattern-directed inference system is a rule-based or production system. Here, the activities of examining data and modifying data are clearly separated. Elementary procedural items are embodied in "rules", each of which specify the conditions in which they apply and the modifications to the data they are to perform. The structure of a rule-based system consists of three elements:

- the rules
- the data structures that the rules access and modify

o an interpreter that controls the selection and activation of relevant rules

A good introduction to pattern-directed inference systems is given by Waterman and Hayes-Roth [38]. Systems of this kind have been used mainly to allow standard program procedures to be executed in an order-independent fashion and for deductive inference. To a lesser extent, this kind of system has been used for inductive inference or learning.

Two example production systems which have been used as tools in the C2 environment are: STAMMER [1] and TECA [10].

3.3 Display systems

Navy Command and Control (C2) decision makers are in need of a powerful, general-purpose graphics interface to a collection of C2 decision aids and information sources. Most do not exist to the level of flexibility which we feel they require.

Traditional graphics systems such as the Graphics Language (GL) [3, 4] interface between an application program and some sort of display hardware. The kinds of hardware are split between vector graphics devices, or those based on raster scan graphics technology. GL is a particularly novel system since it is a terminal independent graphics language.

The TDP/TDI system will be integrated with either the AIPS or VIEW (a derivative of SDMS) systems. AIPS (Section 3.3.1) provides a generalized presentation mechanism for arbitrary kinds of graphical data. VIEW (Section 3.3.2), while having many of the same aims of AIPS, is currently more display-oriented than knowledge based.

3.3.1 AIPS

The goal of the Advanced Information Presentation System (AIPS) project [43, 44, 45] is to provide a system that can present arbitrary information in the form of graphic displays automatically synthesized in real time, in accordance with advice from the user. This advice could be provided with natural language.

Currently, the interface between an application program and the graphics system is at a relatively low level. Communications between the two systems proceeds in terms of graphics primitives. The knowledge about the presentation format and the construction of the information presentation are built into the application program which then calls the graphics system through the use of the graphics primitives. Users, then, often have the capability to communicate their desires for presentation format directly to the application program. Where these desires have not been provided for in advance, the user must either make do with what has been provided or prevail upon the application programmer to make changes in the system.

The intent is for advanced information presentation systems to be very different from this paradigm. The application program communicates with the presentation system in terms of a domain model, describing what information is to be presented rather than what is to be drawn on the terminal. It is only responsible for providing the information to be presented. The functions for selecting presentation format and synthesizing and displaying presentations are assumed by the presentation system. A user communicates with the application program to specify what information is to be shown, and communicates with the presentation system to influence how it is to be shown. Since the presentation system "knows about" a wide variety of presentation formats, presentations can be specified in terms of these rather than graphics interface programs.

There are several implications that advanced information presentation will have on future systems and users of those systems:

- More than one application program can be interfaced to the presentation system. Presentations can be specified and generated that combine information from several sources.
- The application programmer is relieved of the burden of implementing a presentation system custom designed for a particular application.
- The user is presented with a unified interface for *controlling* the presentation of information.
- Unanticipated presentations can be interactively specified by the end users and thereafter automatically generated by the presentation system.
- By generating displays suitable for collecting graphic input from the user, the presentation system can also provide a powerful means of communicating with the application program.
- The presentation system becomes the effective center of the decision maker's personal computing environment.
- The same techniques that are used for graphical presentation can be applied to the management of non-graphical presentation media such as synthesized voice or text.

One of the reasons that this methodology has not been applied to date in other than a research context is that the automatic generation of graphic presentations is computationally ambitious. It is only feasible in the context of a large and powerful symbolic processor dedicated to the service of a single user. Advances in hardware technology and programming environments such as those provided by Interlisp are starting to provide the basis for a suitable computing environment.

Another key to providing a presentation system of the requisite flexibility is the declarative representation of knowledge about the application domain, the user, graphic formats, geometric constraints and graphic display capabilities. The declarative knowledge structure makes explicit the system's models, thus allowing the system to make enlightened decisions about its own behavior. The structure also provides an organizational skeleton for the system's procedural knowledge. The system's behavior can be usefully changed or augmented simply and directly by manipulating the declarative knowledge structure. The declarative representation of the information to be presented simplifies the task of interfacing the presentation system to a variety of application programs and information sources.

A third key is the utilization of a knowledge representation language that supports the inheritance of characteristics and defaults among a taxonomic hierarchy of descriptions. Such a knowledge representation language promotes compactness and consistency in the declarative knowledge structure. KL-ONE (see Section 3.4) is such a language.

The AIPS project is actively researching the areas described above. An initial system was designed and a demonstration system implemented on BBN's experimental bitmap graphics terminal [15]. The system has been reimplemented with a completely re-designed declarative knowledge structure written in a new version of KL-

ONE, and runs on the Jericho. The current issues of concern are the representation of the graphics "world", of canonical presentation formats such as map, graph, and so forth, and the interactions of these formats with viewing organization entities such as windows and display regions.

3.3.2 SDMS/VIEW

The Spatial Data Management System (SDMS) is a system whose current goals are similar to AIPS. It was initially developed by N. Negroponte at MIT, and generalized by the Computer Corporation of America [17, 18, 19, 40, 41].

3.3.2.1 Concepts

Spatial Data Management is a technique for organizing and retrieving information by positioning it in a Graphical Data Space. Its original motivation came from the needs of people who require access to information in a database management system but who are not trained in the use of such systems. The information in an SDMS is expressed graphically and presented in a spatial framework. The database is used to generate the view; how it is displayed is separate.

The current display mechanism uses three color, raster-scan displays. One is used to display a "world-view" of the entire data surface. Another is used to display a magnified portion of the data surface. The location on the data surface of the magnified portion is indicated by a highlighted rectangle which appears on the world-view map. A joy-stick is used to move around the world-view map. Since the information is stored in varying levels of resolution, motion can also be "in" and "out", providing the user with successively more or less detail, respectively.

The third screen is used for ancillary information and echoing of user type-in.

The data presented to the user can come from an amalgam of several sources. Three are:

- o images stored as bit-arravs on a digital disk
- o an optical videodisk
- o a symbolic database management system

The current system, VIEW [19, 41], is more general than simply a graphical display system. Views onto the database which the user sees are created by a <u>view generator</u>. The view generator obtains the data to be displayed from a system which interacts with the DBMS to obtain the data and then formats it into the proper abstractions; this component is called the <u>symbolic view filter</u>.

Work is currently proceeding on how to provide good answers to questions posed to a database of greater complexity than current DBMSs. An example of such a database is one which uses a knowledge representation mechanism of some form, such as semantic networks or KL-ONE.

3.3.2.2 VIEW Compatibility with TDP/TDI

In very general terms, the user interface problem is that of mapping the internal states of one model onto the internal states of another; of mapping one description of the "world" onto another. In our case, it is mapping the internal models which we will be representing using KL-ONE to world models which a display system such as VIEW or AIPS could manipulate.

The process of mapping one description onto another is obviously much simpler if both descriptions are written in the same language. From the point of view of a simulator, it is highly advantageous that all models be written in a common simulation language. From the point of view of a decision support environment, it is highly advantageous that all components share a common knowledge representation language.

To date, VIEW is not implemented using a knowledge representation language, although current directions are to include a knowledge-based component in the VIEW system; KL-ONE has recently been chosen as the knowledge representation language to be used for this component, which will be available shortly on the VAX.

There are several ways in which the TDP/TDI system could interface to VIEW. One is to develop our system on the VAX in Franz-Lisp. Another is to develop a protocol to interface between the Jericho Interlisp implementation of our system and the VAX running VIEW. In any case, there will probably be some translation necessary between different views of how the respective data of the two systems is represented; the main problem is the extraction of data to be displayed from one representation and the translation of that data into another representation for display. We are lucky that the two representations will use the same language.

Although much of the user-interface of the TDP/TDI system is built around output and output styles, there must be a fairly large user-input component. VIEW is primarily an output system whereby a joystick is used to move around the "world" which has been formatted for output. The databases it uses are assumed to be fairly static and change infrequently. This is fine if the user is perusing a fairly static display corresponding to one situation, examining the characteristics of the objects in the area of interest, and so on.

On the other hand, the TDP/TDI system requires that the user-interface for projection be built around a simulation driving dynamic display update at frequent intervals. This could mean that the display is to be updated once every 5 minutes of simulated time which should correspond to every 5 or 10 seconds of real time. The kind of interface which we would have to have to VIEW makes this speed of interaction infeasible at this time.

3.4 Knowledge representation language

There are several tasks which must be addressed when attempting to represent some segment of knowledge:

- o At what level of abstraction do we begin to express this knowledge? This can be characterized as the "representational grain".
- o What are the basic types for the conceptual objects which we are trying to build?

It is easy to design data structures to be manipulated by a relatively simple-minded program. It is much harder to determine the conceptual size of the units of knowledge when trying to capture the details of knowledge about a particular area of expertise. This knowledge will be used to support a general cognitive system whose goals in manipulating we cannot completely determine in advance.

An obvious starting place is to encode the data (knowledge) using a knowledge representation language. Each language provides its user with a set of object types and syntactic conventions. These together suggest how to factor concepts of the domain. The primitives of a language implicitly embody the epistemology which the language's author believes is the way to look at the conceptual world.

We do not go into a discussion here of the theoretical basis and historical framework of knowledge representation languages; such a framework can be found elsewhere [44, Chapter 5]. Rather, we just discuss briefly one of the currently vogue formalisms, KL-ONE, which is available on the Jericho.

KL-ONE is a uniform language for the explicit representation of conceptual information based on the idea of structured inheritance networks [7, 8]. Several of its prominent features are of particular importance to the TDP/TDI system -- its semantically clean inheritance of structured descriptions, taxonomic classification of generic knowledge, intensional structures for functional roles (including the possibility of multiple fillers), and procedural attachment (with automatic invocation).

The principal representational elements of KL-ONE are Concepts, of which there are two major types: Generic and Individual. Generic Concepts are arranged in an inheritance structure, expressing long-term generic knowledge as a taxonomy. A single Generic Concept is a description template, from which individual descriptions (in the form of Individual Concepts) are formed. A Generic Concept can specialize one or more other Generic Concepts (its superConcepts), to which it is attached by inheritance Cables. These Cables form the backbone of the network and carry structured descriptions from a Concept to its subConcepts.

KL-ONE Concepts are highly structured objects. A subConcept inherits a structured definition from its parent and can modify it in a number of structurally consistent ways. The main elements of the structure are Roles, which express relationships between a Concept and other closely associated Concepts (i.e., its properties, parts, etc.). Roles themselves have structure, including descriptions of potential fillers,[1] modality information, and names.[2]

There are basically two kinds of Roles in KL-ONE: RoleSets and IRoles. RoleSets have potentially many fillers and may carry a restriction on the number of possible fillers (e.g., the **officer** Role[3] of a particular COMPANY would be filled once for each person who is an officer of that company). A RoleSet on a Generic Concept represents what is known in general about the

---

[1]These limitations on the form of particular fillers are called "Value Restrictions" (V/R's). If more than one V/R is applicable at a given Role, the restrictions are taken conjunctively.

[2]Names are not used by the system in any way. They are merely conveniences for the user.

[3]In the text that follows, Roles will be indicated as boldfaced names and Concepts will be indicated by all upper case expressions.

fillers of that Role. A RoleSet on an Individual Concept stands for the particular set of fillers of that Role for that individual (e.g., the officers of a particular company).

IRoles (for 'Instance Roles') appear only on Individual Concepts, and are used to represent particular bindings of Roles to Individual Concepts (e.g., the president of a particular COMPANY). (There would be one IRole for each officer position in a particular company, regardless of the actual number of people playing those Roles.)

There are several inter-Role relationships in KL-ONE, which relate the Roles of a Concept to those of a superConcept. Such relationships are carried by the inheritance Cables mentioned earlier. They include:

- restriction (of filler description and/or number); e.g., that a particular kind of COMPANY will have exactly three officers, all of whom must be over 45;

- differentiation (of a Role into subRoles); e.g., differentiating the officers of a COMPANY into president, vice-president, etc. This is a relationship between RoleSets in which the more specific Roles inherit all properties of the parent Role except for the number restriction (since that applies to the set and not the fillers);

- particularization (of a RoleSet for an Individual Concept); e.g., the officers of BBN are all COLLEGE-GRADUATEs; this is the relationship between a RoleSet of an Individual Concept and a RoleSet of a parent Generic Concept;

- satisfaction (binding of a particular filler description into a particular Role in an Individual Concept); e.g., the president of BBN is STEVE-LEVY; this is the relationship between an IRole and its parent RoleSet.

Figure 3 illustrates the use of Cables and the structure of

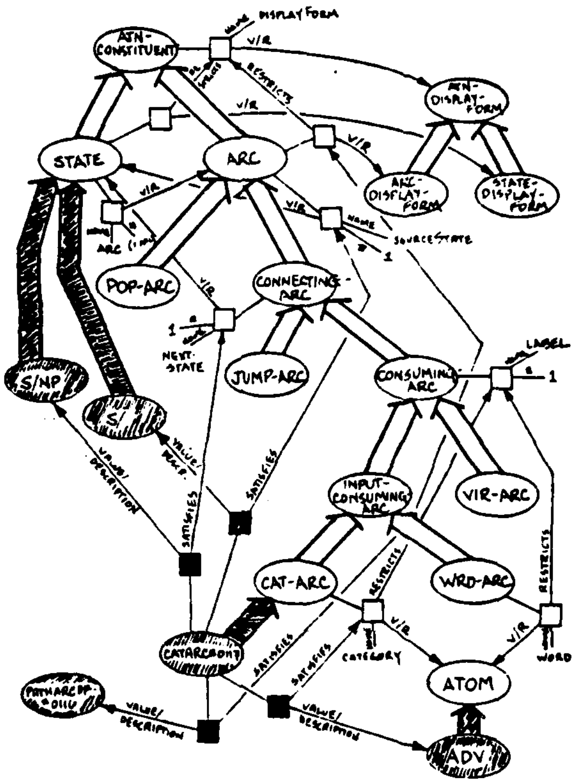


FIGURE 3. A PIECE OF A KL-ONE TAXONOMY

Concepts in a piece of the KL-ONE taxonomy describing an ATN grammar[4]. Concepts are presented as ellipses (Individual

[4]More specific examples to the TDP/TDI system are discussed in the section on system implementation, Section 5.

Concepts are shaded), Roles as small squares (IRoles are filled in), and Cables as double-lined arrows. The most general Concept, ATN-CONSTITUENT, has two subConcepts - STATE and ARC. These each inherit the general properties of ATN constituents; namely, each is known to have a displayForm associated with it. The subnetwork below ARC expresses the classification of the various types of arcs in the ATN and how their conceptual structures vary. For example, a CONNECTING-ARC has a nextState (the state in which the transition leaves the parsing process), while for POP-ARCs the term is not meaningful (i.e., there is no nextState Role). Links that connect the Roles of more specific Concepts with corresponding Roles in their parent Concepts are considered to travel through the appropriate Cables. Finally, the structure of an Individual Concept is illustrated by CATARC#0117. Each IRole expresses the filling of a Role inherited from the hierarchy above -- because CATARC#0117 is a CAT-ARC, it has a category; because it is also a CONNECTING-ARC, it has a nextState, etc.

KL-ONE carefully distinguishes between purely descriptional structure and assertions about coreference, existence, etc. All of the structure mentioned above (Concepts, Roles, and Cables) is definitional. All assertions are made relative to a Context (another type of KL-ONE object) and thus do not affect the (descriptive) taxonomy of generic knowledge.

To be a little more specific, Contexts are collections of structureless entities called Nexus, which serve as loci of coreference statements. A Nexus is a simple object that holds together "wires" from various descriptions, all of which are taken to specify the same object in the world outside the system. The description wires that connect Nexuses to Concepts in the description language are also taken to be in the Context. Thus, a Context can act as a "possible world" which comprises a set of statements about description coreference[5].

We anticipate that Contexts will be of use in reasoning about hypotheticals, beliefs, and wants, and maintaining the temporary "world models" required as output by the projection (simulation) sub-system of the TDP/TDI system (as described in Section 4.1).

The final feature of KL-ONE relevant to our discussion is the ability to attach procedures and data to structures in the network. Such procedures are written in the language of the interpreter and are invoked in particular prespecified situations. We expect attached procedures to be very useful for implementing the object-oriented simulation tool that we are envisioning for the TDP/TDI system.

For a more complete description of KL-ONE, consult [7, 9] or [8].

KL-ONE is an active research area being pursued by two groups at BBN: Bill Woods' natural language research group and the AIPS project. Groups at the USC Information Sciences Institute, University of Pennsylvania, Schlumbeger-Fairchild, the University of Massachusetts, Burroughs, and Xerox PARC are also working with KL-ONE and helping to extend it. It is currently implemented as a set of Lisp functions, and is available in

---

[5] Co-"reference" is not quite the right term, since the objects "referred to" need not exist. Co-specification of a description is probably a better term.

Interlisp on DECSystem-20, Xerox 1100 (Dolphin), and Jericho computers, and is being converted to run under Franz-Lisp on the VAX. These functions provide access to a KL-ONE data base implemented as user-defined data types. Mechanisms for rules and inferencing are currently being developed.

3.5 Knowledge-based Simulation

Because so much of military decision making has to do with the anticipation and control of future events, simulation would seem to have wide application in C2:

- Planning (e.g. formation planning; airstrike planning; in which simulation is used to evaluate alternative plans in alternative scenarios)
- Reaction (e.g. force deployment; using simulation to project the current situation into the near future in order to test possible responses)
- Interpretation of Enemy Activities (i.e. determining what situations the enemy can bring about in the future based on his current activities -- recognizing situations before they occur)
- Training (i.e. a means for gaining practice and experience with C2 situations and decisions)

But in fact, simulation is not heavily relied upon in C2 decision making because it is a difficult tool to build and use. To the extent that these difficulties have been surmounted, however, it has proven to be an extremely valuable one. We believe that it may now be possible to enter a new paradigm for the use of simulation, one in which simulation is used interactively in real time by the individual, unassisted C2 decision-maker to answer arbitrary questions about hypothetical futures, for the purpose of analyzing alternative courses of action.

3.5.1 Simulation models

There are basically two different types of models that we are concerned with in military simulation: situation models (models of the physical world and physical objects) and decision models (models of decision makers, events, and decisions).[6] Simulation that is used for C2 decision support must include both types of models.

To understand why this is so, consider tactical simulators (such as WES) that are fundamentally dedicated to highly detailed situation models. In proportion to the complexity of their models, these simulators are uniformly noted for two characteristics:

- It is difficult to interpret their output in terms of decisions and their outcomes.
- The user is forced to provide a large number of decisions.

On the other hand, consider the sort of simulator that is

---

[6]This distinction presented here is motivated in terms of the pragmatics of military simulation. However, it echoes the more basic distinction which may be made between models that deal with relatively continuous processes and models which are forced to characterize discontinuities, or events. Situation models tend more to the former category, and decision models more to the latter, but there are exceptions.

used for strategic planning. Some of these have exhaustive decision models. This facilitates the evaluation of decision policies because it makes explicit the structure of plans, decisions, and events. Unfortunately, these simulations tend to rely on simplified views of the world. Good simplifications are difficult to arrive at, and marginal ones vitiate the value of simulation as a decision aid.

Simply put, the problem we face is that it is difficult to characterize decision models that deal with highly detailed situation models. To some extent this is due to the fact that we do not completely understand what C2 decisions are made of. To a very large extent, however, it simply reflects the fact that current simulation languages serve decision models poorly.

Despite the emergence of object-oriented descriptions and simulation, most simulation languages are still heavily biased toward the procedural characterization of models. This is because procedural representations can be easily and efficiently implemented in terms of computational processes. Situation models are relatively unaffected by this bias because physical processes are relatively amenable to procedural characterization.

Decision models, on the other hand, are much more difficult to express in procedural terms. Decision models frequently involve linking prospective courses of action with abstractly characterized configurations of the world. This paradigm of decision-making by situation recognition is difficult to deal with in a procedural language because the language does not provide an adequate formal mechanism for describing the configurations. The model builder is forced to describe them indirectly, in terms of the recognition process rather than in terms of what is to be recognized.

Then, too, the decisions are going to be affected by the intention structures of the participants (discussed in Section 2.4 above). Adequate projections of situations where the participants are making real decisions requires inferences about the intention structures of the other participants.

The problem of characterizing decision models is a fundamental concern of Artificial Intelligence research.

3.5.2 Model Acquisition

By acquiring models, we mean the process by which they get inferred or deduced from information about the behavior of the real world, expressed in terms of a formal language, and integrated into a simulation. Model construction is an activity which involves two types of highly specialized knowledge:

- Knowledge about what is to be modeled
- Expertise in the model language, in the process of making useful simplifications, and in specifying a model

The direction of current research is to work at getting the computer to take on more and more of the responsibility of the second role. Some work has been done in this area:

- KL-ONE makes it easy to describe new models in terms of existing ones
- KL-ONE provides a good underlying representational formalism; it is not likely to limit the knowledge acquisition process in unseemly ways
- JARGON [42] and related work on knowledge acquisition have provided a basis for further exploration

- o Constraints and constraint languages [6, 12, 33] provide yet another methodology

Several efforts are being aimed at the knowledge acquisition problem; that is, the problem of creating and modifying knowledge based structures, keeping track of loose ends. Groups at several organizations are working on this problem: SRI International, the USC Information Sciences Institute, Stanford University, Xerox PARC, and BBN. A major emphasis of the work at BBN has been on the development of a language, JARGON, for evolving a complex knowledge base. This language is an English-like lexical notation for KL-ONE which is a combination of an input/output language and a knowledge structure editor. JARGON performs both editing and input functions with syntactic constructions that follow closely the form and structure of natural English. It makes radical simplifications in the range of syntax that it permits, and it preserves the underlying conceptual structures of the sentences that it understands.

JARGON is intended to serve as a surface (lexical) language for an underlying structured inheritance network. It is intended to be able to represent both assertional and descriptive information, but as yet only allows users to specify some descriptive information. It does this by utilizing a formalization of the English words "be", "have", and "satisfy", which together with a few other verbs (such as "called") appear to constitute the bulk of an epistemologically complete foundation for an open-ended range of natural concepts.

3.5.3 The User Interface

The user interface is the means by which the user establishes desired model states and controls the simulation process. It is also the medium through which the user perceives the simulation activity. Both functions are critical determinants of the ultimate usefulmess of any simulation. Unless it is relatively easy both to pose the questions and interpret the answers, simulation cannot be used for interactive decision support.

In many applications, questions will often be posed in terms of the current state of the external world. In these cases, a simulator that is used as a decision aid must have access to a model or description of the external world that can be referred to for the purpose of initializing simulations. In the context of an integrated, knowledge based C2 decision support environment, we can assume that such a model already exists. The problem is to interpret this detailed description of the world in terms of states of the simulator's (possibly simplified) internal models.

In its most general terms, the problem is that of mapping the internal states of one model onto the internal states of another; of mapping one description of the world onto another. This problem arises not only at this particular interface to the simulator, but also within the simulator itself, wherever models meet or overlap phenomenologically. It also arises throughout the entire decision support environment, at the places where environment components with independent descriptions of the world interface to each other or to a shared knowledge-base. In either case, how well the problem is solved has much to do with the power and flexibility of the total system.

The second principal function of the user interface is to support the user in inspecting simulation state and monitoring simulation process. Primarily, this support function consists of generating and updating graphic information displays at the request of the user. BBN is currently developing a powerful display generation system, called AIPS (for Advanced Information Presentation System, see Section 3.3.1), which allows the user to request graphic displays in terms of their semantic content -- in terms of what information is to be depicted rather than in terms of how the display is be constructed. AIPS eliminates the usual requirement that the user must select from among specific displays programmed in advance into the user interface: it lets the user request displays that combine and depict information in unanticipated ways.

3.5.4 Real Time Performance

It is tempting to disregard real time performance requirements, on the grounds that taking them into account at this point is like trying to invent the automobile by designing a race car. Indeed, the functional requirements for decision support simulation are poorly understood, and it is impossible to say much for now on the matter of meeting these requirements efficiently. All we can safely assume is that performance (e.g. response time) will be traded off against many of the other design goals which we have in view.

Unfortunately, the issue of real time performance is extremely critical to the entire notion of decision support simulation. Unless a simulation can be initialized, run, and inspected quickly (on the close order of one minute or less) it cannot be used for real time decision support, although it might still be used as a planning aid. If the time required is much greater than ten minutes, the simulation begins to lose its value even as a tool for developing plans. If a simulation takes an hour or more to run, its uses are probably limited to training and the research and development of doctrine.

Some research still needs to be done, but there are several opportunities which, taken together with some restraints on the complexity demanded in models, could result in simulation that compresses long intervals of simulated time into only a few seconds of real time. Personal symbolic processors, such as BBN's Jericho, are one such opportunity. With such speed, we would truly enter the paradigm of question answering through simulation, and simulation might finally emerge as a dominant cognitive vehicle for military C2 tacticians and planners.

3.5.5 Current status

For the most part, the work described above on knowledge-based simulation may be described as "research to be done." Some initial directions of a knowledge-based simulation research effort are described in Section 8.1.

Some techniques which are applicable to the view of simulation presented have been developed at the Rand Corporation [22, 14, 25]. To the best of our knowledge, they are the only other group working on this problem.

The work being done at Rand is based on an object-oriented simulation language named ROSS [25], which embodies a procedural modelling approach to simulation. However, they do not have the ability to reason as one would using KL-ONE and inferencing as we suggest. Nor is their knowledge-base structured or hierarchically based as seems to be required by the more ambitious system.

4. THE DESIGN OF THE TDP/TDI SYSTEM

There are several procedural sub-systems which make up the TDP/TDI system. They communicate with each other through the use of data-bases. The relationship between these databases and the procedural sub-systems is shown in Figure 4.

First, the sub-systems are briefly described. This description is followed by the description of the various components of the data bases. Finally, we present some of the considerations for the User-Interaction procedural sub-system.

4.1 Procedural Sub-systems

The proposed TDP/TDI system will have five major procedural sub-systems:

- Threat assessment. Given a description of a situation (the identity, location, velocity, fuel-state, EMCON state, etc. of a set of platforms relevant to a tactical situation), this sub-system will be able to characterize the nature and extent of the possible threats to various platforms, and the extent to which they can be countered;

- Projection/Simulation. Given a description of a situation, this system will provide descriptions of likely future situations, at specified time intervals, or when specified classes of events are expected to take place. This projection will be based on information provided by the user about the expected behavior of individual platforms and groups, including plans, goals and intentions. Such plans will include expected course legs, EMCON conditions, fuel constraints, search patterns, etc.;

- **Display.** This system will provide both graphic and tabular displays of situations whose description is produced by the projection sub-system, with the user specifying which situations and what characteristics of the situations are to be displayed, and the format to be used;

- **User-interaction.** Provides the user with overall control of the activity of the TDP/TDI system, including specification of projection tasks to be performed, types of threats to be analyzed and displays to be produced and perhaps saved;

- **Sensor and Knowledge integration.** These two sub-systems are responsible for integrating source data (long and short term) into the world database, and for providing the user with Tactical Deception Indicators. The Sensor-Integration module will take in preprocessed sensor data in a format to be specified by the TDP/TDI system, and determine if there are indications that the information provided by the sensors has been corrupted by tactical deception attempts of the opposing force; in other words, it will perform constraint satisfaction. It will suggest alternative interpretations of such data and, by interface with the threat assessment and user-interface modules, will indicate possible tactical threats posed by such alternative situations.

  On the other hand, the Knowledge-Integration module will take in other forms of knowledge, such as weather information or intelligence data, and integrate that data into the database.

In addition to these procedural sub-systems, there are several data bases through which they communicate.

4.2 Data Bases

The data bases used in the TDP/TDI system are intended to support the activities of the five procedural modules listed in Section 4.1. The information to be represented can be broken into two broad categories:

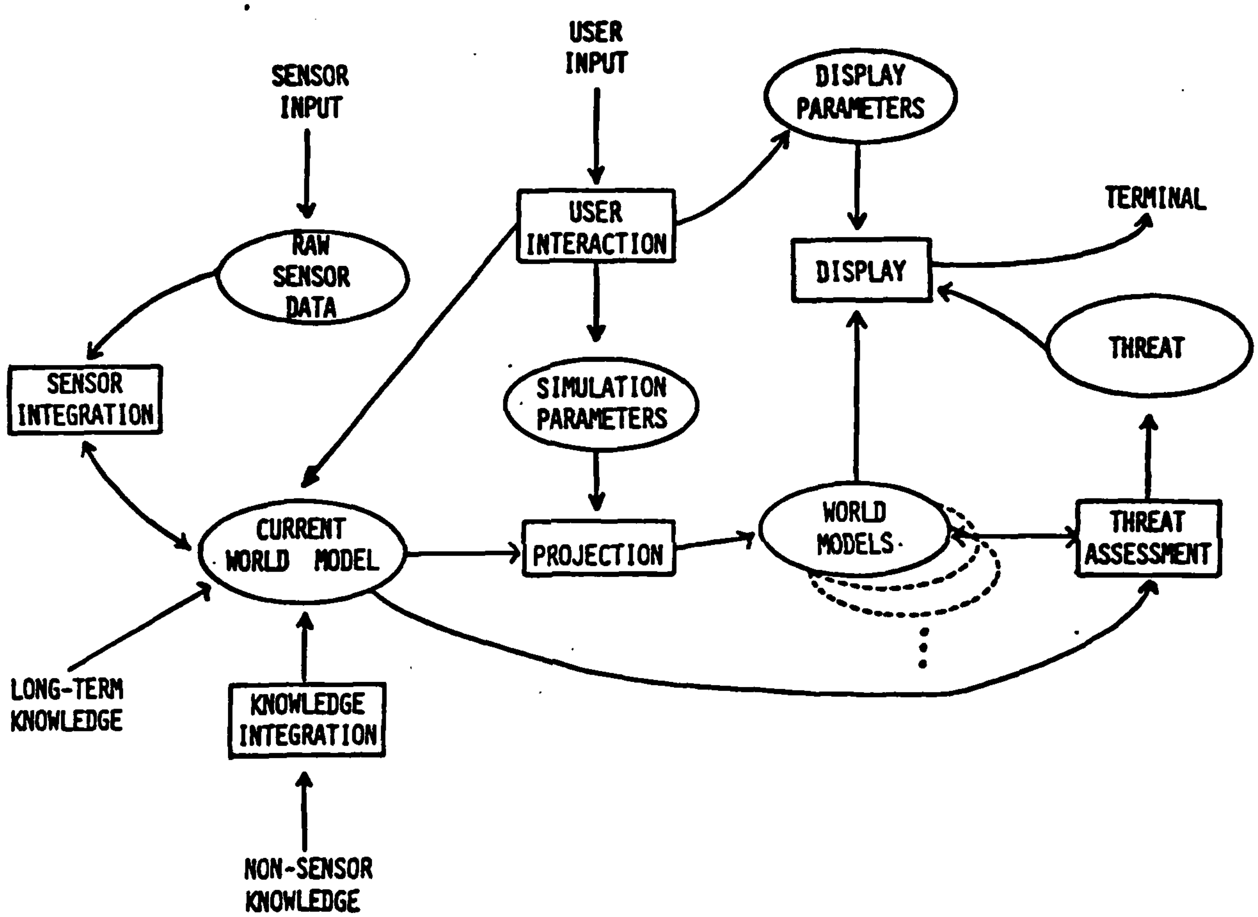


FIGURE 4. SYSTEM OVERVIEW

1. long term knowledge -- such as general capabilities and characteristics of different classes of platforms, weapons and sensors; friendly and enemy doctrine; special properties of known individual platforms, etc.

2. situational knowledge -- properties of the platforms, sensors and weapons that make up a given tactical situation, such as position and velocity of platforms, EMCON status, etc.

Long term knowledge can be assumed to be fixed for any given tactical situation. It changes only slowly with time as new classes of objects are entered into the data-base, or new or corrected information is obtained about the properties of classes of objects already in the data-base. It will contain information relevant to many possible tactical situations, and for any given situation much of the data may be irrelevant.

Situational knowledge consists of the information that varies over time during any given tactical situation. It is updated by the sensor integration sub-system in response to sensor, surveillance and current intelligence reports.

The way in which situational knowledge is represented will have a substantial impact on all of the sub-systems of the TDP/TDI system. It is used by the threat assessment sub-system to evaluate the threat potential of a given situation. Since a primary purpose of the simulation sub-system is to provide information to evaluate possible future threats, the simulation sub-system will also represent states of the simulated world in the same language/data structure used to represent the current tactical situation. The display sub-system uses situational knowledge to produce displays such as platform position, bearing, status and threat potentials for both the current situation and projected future situations.

Situational knowledge is logically connected to long term knowledge. Constraints on the values of various situational parameters (e.g. speed for platforms, frequency and prf for radars) are specified by the characteristics of various classes of objects. Threat possibilities are determined not only by the current status (position, bearing, etc.) of the objects (e.g. platforms, weapons and sensors) taking part in a tactical situation but also by the characteristics of the particular classes of objects involved.

Because of the logical interdependence of the two categories of data, they will have to be represented in a uniform formalism. An alternative would have been to use a standard data-base management system (e.g. INGRES, a relational data base system [34, 35]) for the long term, slowly changing data, and to use specially designed internal data structures for situational data. Since information from the long-term knowledge data base will be repeatedly accessed by each of the procedural components of the TDP/TDI system, it will be advantageous to use internal data structures for such data. For tactical situations the amount of such data should not prove to be too burdensome, especially if we can use sophisticated data structures in a large virtual memory system to represent the TDP/TDI system's total knowledge base.

In general, much long term knowledge has the property that it describes the properties shared by classes of objects (such as platforms and sensors) in all situations. The situational knowledge consists of specific information about individual objects that holds at some time in a particular tactical situation. We would like a to represent these two types of information in such a way that in any given situation all the information about each object is readily accessible, without having to store multiple copies of information that is independent of the situation. Thus, the data structures we want to use should provide an inheritance mechanism. Such a mechanism allows an object in any given situation to inherit both properties from the class of objects to which it belongs and specific properties unique to that situation.

The inheritance mechanism should allow a single object to be an instance of more than one general class of object, and for class membership to be an object attribute that can vary from one situation to another. This will allow an object to inherit some properties in all situations (e.g. the Wilson is a DDG in all situations) and inherit different classes of properties depending on its orders and details of a particular situation (it may be a task group leader in some, a picket in others, etc.).

Since knowledge of the properties of an object in a given situation is often incomplete, the representation should have the ability to provide default values for any parameters needed by processes and not explicitly available in the data base.

In addition to the modularity needed to support inheritance of object properties from one or more classes, the representation must support the clustering of properties relevant to various tasks performed by the five procedural sub-systems. For example, the various characteristics of a given class of ship are used in different ways by different processes. Maximum speed, cruise speed, fuel capacity, fuel consumption, fuel state, and so forth are used by the kinematics model within the projection sub-system. The effective radar cross-section is used by the sensor detection model, along with information about antenna height and radar power, resolution and sensitivity. Radar power, frequency, prf and antenna rotation characteristics are used by the ESM portion of the sensor integration module to determine alternative emitters which might account for a given observed signal.

Furthermore, each of these has constraints that are more or less inherent. For example, a vessel may reduce the power in its radar emissions, but cannot increase it. On the other hand, it may increase its radar cross-section with a corner reflector, but cannot decrease it. That is, the specification of the constraints must satisfy the other requirements.

Knowledge representation languages make it possible to represent information about complex situations in a way that facilitates general reasoning about the situations, implementation of efficient situation recognition algorithms, and such flexible programming styles as data-driven pattern directed inferencing. They provide both inheritance mechanisms and modularization mechanisms suitable to meet the demands we have imposed on the representation scheme for situational and long term knowledge.

The KR language most suitable for our purposes and most readily available to build the demonstration system is the KL-ONE system (see Section 3.4). KL-ONE is implemented within the Interlisp programming environment at BBN. For the purposes of this design effort we will assume that all relevant data bases will be represented as KL-ONE structures. Long term knowledge will be represented primarily as KL-ONE generic structures, since the major use of such knowledge is to characterize the properties of classes of objects. The KL-ONE inheritance mechanism will facilitate the retrieval of such information. Situational knowledge will be represented through the use of the KL-ONE Context mechanism. The situation at any instant of time will be represented as a Context, with Nexuses for the individual objects (primarily platforms, sensors, weapons, communications systems, weather regions) relevant to the tactical situation. Information about each such object will be represented by Individual Concepts which describe the known or inferred properties of the object.

4.2.1 Long term knowledge

The following describes the long term knowledge needed for the TDP component. It contains information about platform performance characteristics, platform weapons characteristics, platform communications characteristics and sensor characteristics.

4.2.1.1 Platform characteristics

For each class of platform we will need several different types of information, used by different process modules in the TDP/TDI system. Some of this information is best represented by giving values for certain parameters which define characteristics of the platform (e.g. waterline length, beam, height, radar cross-section, etc.). Other information is best represented by a procedure which can be called to compute required values as a function of relevant situational information (e.g. maximum speed as a function of sea state). This information will be used both as a source of information to various inference and pattern-recognition processes, and as the basis of the object-oriented simulation/projection sub-system.

- Identification parameters: Each platform will have several parameters associated with it that will serve to identify it and permit the TDP/TDI system to interface with the user, and with other sources of information. These parameters will include:

  - **Names:** The standard alphanumeric name for the platform if it is known (e.g. Forrestal, Eisenhower, Long Beach). The platform name is useful for user interaction. Names may be provided for aircraft not standardly identified to facilitate user interaction. We will also allow for code names assigned within an operational context.

  - **Hull/tail number:** This is used to correlate platform information with visual sightings and other reports.

  - **VCN or UIC:** The standard ID numbers for US and foreign vessels, used for interchange with other data-bases, and correlation with reports.

  - **Class:** The platform class is primarily represented by an inheritance cable to a generic concept representing information about the class of platform. The TDP/TDI system may also provide an alphanumeric name and/or abbreviation for the class to facilitate user interaction.

- **Maneuver capability characteristics:**

  - **Maximum speed:** Maximum platform speed is a function of the class of platform and perhaps weather and position (sea-state for surface vessels, altitude and possibly turbulence or storm conditions for aircraft, operating depth for submarines). The state of the propulsion system is also a determining factor of maximum speed, and may be important information for planning and for detecting deception. Propulsion system state assumes the reception of surveillance/intelligence reports which have some bearing on these factors (e.g. damaged screw inferred from passive sonar signature). Maximum speed will also be a function of the configuration of the platform (see below).

- Cruise speed: The cruise speed is the speed for optimal fuel efficiency. Like maximum speed, it is a function of the class of platform, weather, the propulsion system and platform configuration.

- Agility: Platform agility might be expressed either as radians/second turning rate or as a minimum turning circle in yards. It is possibly a function of sea-state; certainly it is a function of speed and class of vessel. Agility information may be relevant to the detection of deception, if we are lucky enough to detect a vessel performing maneuvers which are impossible for the class of platform which it is intending to emulate. It is not clear at this time if there is much payoff in monitoring for such slip-ups. Platform agility may not be relevant to the level of simulation and threat assessment dealt with under this contract, or may be factored in to other models.

- Navigational constraints: The primary physical constraint is a restriction on depth (vessels are either shallow draft or deep draft); in the case of submarines, it is a more complicated matter captured with an operating envelope expressed in terms of depth and (vertical) velocity. In the case of surface vessels, sea-state, current and wind direction may impose constraints on heading.

  We include under this category the obvious constraint that vessels cannot have planned courses that intersect land-masses! There are also doctrinal constraints on operating regions which may be modelled in this manner, e.g. submarine operational areas, and on operation in other than international waters.

  A likely representation for these constraints is for each class of platform (where we include specialized sub-classes to include doctrinal constraints in special situations) to have an associated procedure which will indicate if a navigational constraint is violated by a proposed course leg; if so, it will give the position at which the constraint is first violated and the nature of the constraint (e.g. depth constraint or doctrine).

- **Fuel consumption:** For each class of platform we must represent fuel capacity, as well as the function relating fuel consumption to speed, configuration, weather, etc. We may also want to include information as to standard doctrine for a class of platform and as to what fuel states call for special action (return to base or rendezvous with tanker). These may be a function of situational features (e.g. assignment, possibly expected weather conditions for carrier aircraft).

- **Configuration possibilities:** There may be special configurations possible for the platform, such as towing or dangling sensors. They may have significant effect on other platform parameters such as speed, fuel consumption, external signal generation (environmental disturbance), and sensor capability.

- **Detectability and external signal generation:** In order to determine the detectability of a platform under various conditions, we must have at least approximate models for the generation of detectable signals (acoustic, EM, IR, MAD) under various operating conditions. Acoustic disturbance is a function of ship class and a non-linear function of speed, and may be a function of sea-state (as it affects S/N ratio). Certain other signal generation is unavoidable, such as thermal and MAD effects, and it is possible that other EM signals are generated even when communications and radar are silent. Generation of signals by active sensors/communications devices must also be modelled, so as to account for changes in detectability based on various EMCON policies.

  Parameters relevant to radar detection must be included, e.g. radar cross-section and/or the parameters from which an approximate cross-section can be computed, such as length, beam and height.

- **Platform related sensor characteristics:** We include here the type and critical location parameters for various sensors such as radar, active and passive sonar, and ESM equipment. The properties of the various types of sensors themselves may reside in the long-term knowledge structure defining each type of sensor. The properties of the sensors will be functions of placement (critically, antenna height and gain). It is possible that better simulation performance may be obtainable if each class of sensor installation is modelled separately, to take account of difficult-to-parameterize effects. The sensor models, whether derived from using parameterized generic models for equipment class, or special models for equipment installation, will have to take into account environmental conditions such as weather and possible interference from the sensor platform or nearby platforms.

- Other equipment characteristics: Information about the parameters of various classes of communication equipment related to detectability should be included. We may need other information such as the typical use of such links to allow specification of certain EMCON restrictions in simple ways; since the type of HF/UHF/Satellite equipment may vary on a per ship or per class basis, we need to know the ways such equipment is likely to be referred to in various EMCON restrictions.

  It is unlikely that such characteristics as information rate or voice/data restrictions will be necessary for the modelling to be done in this project. We may want to model certain properties of C2 equipment (capabilities of NTDS on board a ship, ability to share data, control various operations, etc.) and what types of communications links can be used to support such operations. Only the most elementary models of this form are at all likely to be useful in the initial system (if any), but the functional hook on which to hang such information may be useful to provide for later extensions.

- Platform-related weapons characteristics: This includes the types of weapons carried aboard the platform, the number of launchers for each weapon, the rate of fire or cycle time of each launcher, and the total number of weapons of a given type carried. In addition, the restrictions on weapons utilization induced by sensor and C2 capabilities may need to be modelled. There are operational limitations on the use and rate of fire of weapons that span platform boundaries; ships operating close to each other may not be able to simultaneously control more than some given number of a class of weapon, perhaps depending on direction of launch, because of the interaction between sensors and control systems. All these effects would have to be taken care of in computing both the fire power and defensive capabilities of a coordinated group of platforms. The actual models used in the TDP/TDI system may have to aggregate many of these effects because of manpower limits, security issues and perhaps computational efficiency.

- Other characteristics: Several other characteristics may need to be given on a per class basis. For example, tankers have cargo capacity, and aircraft have altitude limits and altitude-dependent speed and fuel consumption behavior.

4.2.1.2 Sensor characteristics

As indicated above, we may separate out those aspects of sensor performance that are not dependent on installation parameters.

- Identification parameters: This includes EM parameters for radars that may be used for identification of radar type from ESM data. As mentioned in the section on deception, there are possibly platform-dependent EM parameters which might be relevant to detecting deception. Because of the fact that the demonstration is intended to run on a non-secure machine, we will probably assume the existence of some such parameters and experiment with the way in which the detectability of deception tactics depends on the accuracy/reliability of ESM reports on variations of signals from expected parameters.

- Sensor capabilities: These are functions of sensor class, operating state (power output limitations), installation characteristics (antenna height and placement), environmental conditions (clutter from sea-state and rain, effect of sea-state on figure of merit for hull-mounted sonars) and target characteristics. It may be useful to provide overall range limits to reduce computational load, and to provide specific functions for each sensor installation in order to take into account installation-dependent parameters. The fidelity with which various sensors can be modelled within the manpower, security and computational resource constraints on this contract has yet to be determined.

One important use of such sensor information is to facilitate "lack of data" inferencing. It is often as vital to know that certain sensors have not picked up targets as to know that they have. Under appropriate conditions on sensor capability and environmental conditions such negative information indicates that no platforms with given characteristics exist in a certain area. It is critical to distinguish these conditions from those in which "no data" cannot be used to infer "no target."

4.2.1.3 Weapons characteristics

Depending on the detail/fidelity of the models necessary for threat assessment, there are many parameters which might be represented for each class of weapons. One advantage of the symbolic/knowledge-based approach is that it should make it relatively straightforward to vary the fidelity of models depending on the class of weapon and the type of inference/assessment requested.

The classes of information relevant for modelling threat potential of weapons include:

- o The targeting envelope. Range is not properly representable as a scalar in most cases of interest, especially in the case of AAW weapons. Most often, these characteristics are subsumed in a simple probabilistic model based on weapon pairs (AW weapon vs. AAW weapon) plus a scalar range factor. Some simple characterization of effects due to orientation of weapon and target altitude may also be included. How far to go depends on how important it is to treat the interactions with individual vessels as opposed to the probable aggregate effect of an attack on a group of vessels.

- o The types of sensors used by the weapon system at different phases of the attack process include sensors and communications used by the launching platform and/or designator platform, as well as sensors used by the weapon itself.

- Limitations or effects dependent on the general characteristics and/or state of the target include depth charges useless against aircraft, and radar-seeking missiles useless against silent targets.

- Environmental effects due to launching, control and guidance of the weapon. This ties to sensors/communication used by weapon and controlling platform.

- There should possibly be a characterization of weapons that summarizes how many applications are required in order to completely reduce the target. This number is one (1) for most weapons of interest. The effect of launching a weapon at a target is to invalidate most of the physical state information about the target. For example, it is a problem to determine whether the weapon actually reached the target or not. Then there is the problem of the resulting capability state of the target. There is no reliable way of modelling these effects; they must be observed.

4.2.1.4 Oceanographic data

This pertains to the behavior of sensors in various oceanographic and climactic states. In general, the performance degrades in bad weather; performance is sporadic and reliable ranges are reduced. This will have an effect on the interpretation of sensor data, and perhaps on the evaluation of threat potentials.

4.2.1.5 Deception categories

The nature of some aspects of deception have been discussed in Section 2.3. Some of the possible categories which will be enumerated in the knowledge structure are:

- o Enhanced radar echoes. This includes the actual effects of both active and passive enhancement techniques, and the corresponding image quality assessments.
- o Sound absorbing materials.
- o Active sonar responders.
- o Passive sonar.
- o COMINT.
- o Sensor jamming.
- o Speed/position anomalies.
- o Decoys.
- o Visual disguising.

Also included must be a history of deception techniques used by both enemy forces in general, and specific commanders in particular. Often, platform intentions may be determined by knowing who the commander is and how that commander acts in certain situations. Section 2.4 discusses intentions in general, and Section 4.2.2.1 below discusses the database structure for platform intentions.

4.2.2 Situational knowledge

This includes information such as platform status, weapons status, communications and sensor status, weather, and so forth.

4.2.2.1 Platform status

- Localization/velocity history: Information about previous sightings and sensor contacts with the platform, giving localization/velocity information, is included. This will contain a set of localization reports, with position (area), course and speed if determined, and time of report. The reports will also include information about its source, and perhaps some indication of its (assumed) reliability. Estimated positions inferred from the projection model may also be included, with the projection model given as source.

- Current localization/velocity: This information is what is known about the location of the platform at the time represented by the context. Included is information generated by the projection model.

- Current configuration: This is an indication as to whether the platform is in a special configuration, e.g. towing or dangling sensors.

- Fuel state: The fuel state is represented as the per cent of maximum fuel remaining. The estimated cruise range and range at maximum speed may also be included.

- Acoustic emissions: The noise intensity derived from platform class and speed may also be recorded for surface vessels and submarines.

- Intentions: The scope of the model for intentions which will be supported by the TDP/TDI system is not yet determined. It should at least include planned course legs and EMCON behavior on those legs. It may include more complex course models such as search patterns and task force maneuvers, and/or planned response to the detection of enemy forces. Simple procedural models for such behavior could be implemented, but there is a question as to how much of a declarative representation of such intentions is necessary and/or useful within the initial TDP/TDI demonstration. Some model for Rules of Engagement may be included, but this is likely to be severely limited in the demonstration system.

4.2.2.2 Communications and Sensor Status

- Emissions state: This is a list of all operating emitters, including their frequency, power, and other parameters relevant to the ESM and passive sonar models.

- Communications state: With which other platforms, bases is the platform communicating -- including simply receiving data. This may be critical for determining if the platform has information about targets derived from NTDS or other sources.

4.2.2.3 Weather

- History: This may be relevant to determining if a platform which is suddenly detected might have been previously detected, or to explain previously observed maneuvers if the weather data is less timely than maneuver information.

- Current status: We need to model the effects of various weather conditions on maneuverability of platforms and capabilities of various sensors. This includes information on sea-state, wind speed and direction, precipitation, cloud-cover and visibility in various regions. A likely representation is to provide descriptions of regions of whose weather parameters are similar with regards to the effects considered in the remainder of the system (i.e. weather variations too small to cause significant differences in effects on maneuverability and sensor performance).

- Predicted weather: This must be modelled at some level to give information needed for the projection/simulation component. Two possibilities are to include velocity information with current status of weather systems (to model moving storm areas and pressure systems) or to provide descriptions of expected weather at specific future times and interpolate for times not covered.

4.2.2.4 Order of Battle

This information is necessary to indicate which classes of platforms and/or individual platforms are expected to be in the operational area during the time of the tactical situation to be analyzed. This will have an effect on the interpretation of sensor data, and perhaps on the evaluation of threat potentials (e.g. if there is strong evidence for the presence or absence of particular classes of cruise missile submarines, certain types of combined attacks are more or less likely). If a vessel is identified as belonging to a certain class on the basis of sensor data, the OB information may enable the system to propose specific identification, which in turn may lead to making available information about characteristics of the platform not shared by others of the same class. This would also include information about basing, likely assignment and C2 information which might affect the likely activities of observed platforms. The extent to which the demonstration TDP/TDI system will make use of such data has yet to be determined.

4.3 Sensor data representation

The goal for representation of sensor data is to produce as much of a unified format as possible for the different classes of sensors, so that they can be used interchangeably insofar as possible. To do this we distinguish four different aspects of information that is (potentially) available from different classes of sensors.

- Localization: This varies widely from one type of sensor to the next. Shipboard and airborne radar provide very accurate position fixes, relative to the position of the radar platform, and we assume that problems of gridlock will be resolved by the time the TDP/TDI system is actually in operation. Thus we assume that local radar can give almost exact positions. OTHR on the other hand, is likely to give a larger region with error bounds, and passive arrays like SOSUS will also give regions with error bounds. Active sonar is probably more like local radar than OTHR and SOSUS, although there are error possibilities due to ducting and multipath. Passive sonar and ESM tend to give more bearing information, but this can be represented as localization information, since it is generally possible to estimate minimum and maximum ranges within the bearing.

Since the shape of the localization region will depend on the sensor and on the environmental conditions, the representation scheme that is most reasonable seems to be to represent one or more regions by means of a polygonal approximation; this leads to efficient algorithms for combinations of regions due to different sensors. Several schemes have been developed for representing polynomial regions in KL-ONE for the AIPS project, and we may use one of these, or a simpler one using a more conventional data structure, depending on further analysis of computation space and time requirements.

- **Velocity:** Doppler techniques may be applied to many sensors and so some sort of velocity estimate may be available. This is probably best represented as a primary velocity vector with error bounds.

- **Platform (source) identity parameters:** Particularly for ESM, it is likely that parameters related to the identity of the source will be available. Frequency, prf and other components mentioned in the section on deception techniques (Section 2.3) will be included here. For radars, information about amplitude of echo might be useful, since it gives information about likely target size (or use of reflectors or active amplifiers for deception). High resolution radar techniques might produce more information and so we leave available a parameter section to accommodate such data. Passive sonar will also provide information about target class.

Since the demonstration system will operate in an unclassified environment, we will probably have to provide dummy parameters and some sort of dummy matching system. We can experiment with the sensitivity of the performance of the entire TDP/TDI system to variations in the ability of sensors to provide good identification data.

For the present, we shall ignore the identification techniques that might be provided by SAR and ISAR, and also other kinds of radar imagery.

- Platform (source) operational parameters: Information about operating conditions can be obtained from ESM measurements and sonar (e.g. number of screws operating, damage to screws, operating mode of radar or communications transmitter). Again, due to the computing environment, we will not provide accurate models of this information in the demonstration, but will provide the data structure and procedural hooks necessary to experiment with the effect of having such data on the performance of the TDP/TDI system.

Each sensor report will contain the following information: Time of sensor contact, class of sensor, localization information, velocity information, identity parameters and operational parameters. Other information such as overall reliability estimates and special environmental conditions may also be appended.

4.4 Fusion data

Several Navy programs are aimed at the fusion of data as a means of vessel identification and tracking: Integrated Ocean Surveillance, Ocean Tactical Targeting, Outlaw Shark, and Integrated Tactical Surveillance System. They are explained briefly in Section 7.

In all of these programs, various databases will be accessible giving such information as identity, position, track and position reliability information for all known platforms in a given region. Currently, there is no access protocol to allow retrieval from these databases of information relative to specific localities. Outlaw Shark, however, is primarily a broadcast system intended to provide targeting information to submarines (although it can be used more generally); information updates are provided automatically or on request, and are relative to specific theatres of interest.

The importance of interfacing to the databases provided by these programs by the TDP/TDI system cannot be underestimated; this data will provide threat information to group commanders before their own sensors are able to detect the presence of an enemy, even though the actual location data mav not be completely accurate. All of the information gained by accessing data from the fusion center(s) will fit into the situational knowledge database as has been previously described in Section 4.2.2. Data from these sources needs to be integrated with the long-term and situational databases. The exact mechanism to do this has not yet been designed, but should be one of the goals of a future project.

4.5 User Interaction

The user interface is the means by which the user establishes desired model states and controls the simulation process. It is also the medium through which the user perceives the simulation activity.

User input will be used to specify several varied aspects of system behavior:

- o Descriptions of the objects relevant to the world model
- o The intentions of some of those objects
- o Parameters for the simulation (e.g. simulate from now until some end condition is met, displaying the state every five minutes of simulated time)
- o Parameters for display (e.g. how to display certain situations, area of interest, etc.)

Rather than go into the specifics of all of the available commands and interactions which could be performed to completely specify all of the required data for each of these parts of the system, we first briefly describe the various interaction styles which we expect to support in the final system, and then show the uses of user-input.

What is defined here is only an expedient of the design for the purposes of the demonstration system. It is an interim solution until either AIPS or VIEW is available for interfacing at the appropriate level and reliability. The problems of interfacing to AIPS or VIEW are described in Section 4.5.3.

4.5.1 Interaction style

The interaction style that will be used depends on the user's goals and on the amount of data which needs to be displayed on the screen at any one time. For this reason, it ought to be possible to use various styles which may be intermixed, and for the user to adopt his interaction style to his changing needs.

The Jericho has a mouse[7] with three buttons which is used as a pointing device. A mouse can be used for selecting operations to be performed from a menu, drawing land masses or tracks, or specifying a narrowing of the context of the displav.

Menus are a mechanism for packaging several operations or arguments together. A menu operation can be selected (i.e. invoked or executed) by selecting it in some manner. One way is to use a pointing device like a mouse. Another is to specify that you want to select an operation from a specific menu (for example, by moving the cursor to the menu with the pointing device, with cursor moving keys, or by typing some simple meta-command), and then to select the operation by typing the operation's number or pressing a button on the pointing device.

Menus are not the only way to specify an operation to be performed. Commands can also be specified using the terminal's keyboard. The command language in the TDP/TDI system will be that made available by using the Interlisp function ASKUSER. This function supports multiple command language styles including user-completion and auto-completion. User-completion means that during command name or argument specification, when the user types a completion character such as a space or carriage return, the rest of the name is typed out if it has been unambiguouslv specified; otherwise, a bell is sounded. Auto-completion is the mode where the rest of the name is typed out when a certain number of characters has been typed by the user and recognized by the system to be valid and to completely specify the name.

---

[7]A mouse is a small hand-held device attached to the terminal that one moves around the top of a desk, for example. As it moves, a cursor moves correspondingly on the screen.

All menu invoked operations will also be accessible through typed commands. This will provide users with a choice of interaction styles, and allow them to change styles as they become more familiar with the system.

A final interaction style may be dictated by the kind of data which will be collected. Some objects have certain parameters which define them more completely. For example, a platform has a type, kinds of weapons, velocity, several kinds of sensor with different operational characteristics, and so forth. All of these characteristics of each of the objects is known. User-specified tables are one interaction style that can be used to provide an interface for specifying the actual data which will go into the data base and world model for the specific characteristics of instances of objects. Various mechanisms can be used to specify "unknown" values, such as null entries in the table, or typing a specific value. So that wrong information will not be entered into the data base using this mechanism, a constraint system will be provided. This allows the data base to have built in constraints on the kind of data which any slot can have given the rest of the slots in the definition of a particular object.

4.5.2 Uses of user input

User input will be used for both specifying intentions and as a general system input mechanism. Specifying intentions is the specification of a set of actions that a particular object (such as a vessel or task force) is intending to take, and how to modify those actions. Examples of the kinds of intentions which we intend to handle are:

- **Geographic tracks.** This is a specification which can be made either to the world model or for a specific simulation. It indicates that a ship, for example, will follow a particular path and is heading to a specific destination. Such a path might be a search pattern followed by a scout ship while that ship is in the role of a scout. Other intentions could apply at other times.

- Bearing. Similar to a track, an intention that an object might wish to follow is a particular bearing until such a time as some condition is met. For example, bear 353-degrees at 25 knots until we detect an enemy vessel.

- Constraints on relative position. A particular vessel might want to skirt the coast at a distance of no more than five miles. When a vessel is part of a task force, an intention might be to maintain a specific distance and location relative to the main ship of the force.

- Task force. The intentions of a task force are not necessarily globally the intentions of the entire force, but could be the intentions of the main vessel of the force and intentions of the other members of the force (auxiliary vessels) relative to the main vessel. The relationship of the auxiliary vessels to the mail vessel could be either relative positions or motions.

Generalized system input will be directed to the appropriate sub-system depending on the command and the context of the interaction. Some of the uses to which input could be put for each module are:

- Sensor data. User input could be to define or modify a scenario, or in the process of a simulation or exercise, to simulate a "message" arriving from somewhere. Such a message might be input to a sensor or intelligence data.

- Current world model. As previously mentioned, users should be able to specify which objects are to make up a world model of interest, and what the characteristics of those objects are.

- Simulator. Various parameters which will drive the simulation for determining potential future threats. Some of these parameters are:

- Limits on the time for the simulation to happen (e.g. the next six hours)
- When to have the simulator display its model of the state of the world (e.g. every 5 minutes of simulated time)
- What is being looked for in the simulation, such as paths of vessels crossing, vessels reaching a certain range of each other, and in weapons range of a vessel.

o Display. Set or examine various display characteristics:

- What objects are important. Sometimes only vessels are; at other times, the vessels and their detection ranges; at other times, just task forces.
- How various objects are to be displayed. For example, a vessel might have an icon representing its class, and other icons for the kinds of weapons it carries. Color might indicate how many weapons of a particular kind the vessel is carrying. Sensor range is similar -- each sensor might have a color associated with it, and the ranges might be displayed as shadings, circles or pie-slices. The human factors of such displays are well understood, reflecting the large number of studies that have been performed.
- Area of interest. Where or what the center of the display should be, and the radius from that center. The area of interest might be a specific location, or might be a moving object.

o User profile. It might be possible, in the time allotted, to specify a profile mechanisms for various aspects of the system. This is to mimic what we expect the final system to be capable of in terms of allowing individuals to specify the general characteristics of behavior. Some of these behavioral characteristics are:

- Display. How to display objects. This includes a general characterization of the specifications allowed above. Here the specification conditions might be slightly more flexible by allowing the display method to be contingent on various states in the data base.
- Simulation. What some general characteristics for a simulation and associated display might be.

4.5.3 Interfacing to User-Interaction Modules

There are three choices for a user interaction mechanism for the TDP/TDI system:

1. Build our own
2. VIEW
3. AIPS

Should neither AIPS or VIEW be available in the time frame of the implementation of the system, a simple user-interface could be built. This, however, would only be an interim solution until a more robust interface became available.

If the VIEW system were to be used as the mechanism for user-interaction in the TDP/TDI system, an interface would have to be constructed between the two systems. The protocols which might be used between the TDP/TDI system and VIEW could be at any of three levels:

- o Graphics language. The TDP/TDI system would interact with VIEW using a graphics language style of interface. That is, the interface will allow operations such as the display (or erasure) of an object at a particular location on the screen, draw a line, shade a region, and so forth.

- **Data base.** Two databases would be kept: the VIEW database and the more complete database of the TDP/TDI system. All objects that would be of interest to a user at a particular moment in time would be contained in both databases, but the VIEW version might not have all of the ties to other parts of the TDP/TDI database. This data would be used to drive the display and some of the user interaction; complex user interaction requests would have to be handled by the TDP/TDI system. Interactions between the TDP/TDI system and VIEW would be in a database update style.

- **File transfer.** Here, the VIEW system would have a complete model of the TDP/TDI database which is to drive the display at any particular moment in time. As new world models are created and require displaying to the user, the entire model would be shipped to VIEW for displaying.

There are two levels of tradeoffs that will affect the choice of which protocol would be optimal for VIEW-TDP/TDI interaction:

1. The amount of data to be transferred, and the complexity of the process which must convert the data (in each direction) to the appropriate formats.

2. The decision by CCA to use KL-ONE as the knowledge representation tool.

Nevertheless, the operability of the system will be different from if an integrated user-interaction module were used, primarily because a communications protocol would still have to be developed.

Interfacing to AIPS would be much easier. It is implemented in Interlisp and KL-ONE, and is currently running on the Jericho. No special communications mechanism need be constructed to interface to it. However, we believe that it currently is not sufficiently robust for use in the TDP/TDI system.

5. THE USE OF KL-ONE IN TDP/TDI AND SIMULATION

This section is an extension of the description of KL-ONE given above in Section 3.4, concentrating on its use as a tool in the TDP/TDI system. This description is intended not to be exhaustive, but rather indicative of our use of KL-ONE to represent the objects and perform simulations in this system. The examples given are highly simplified for this discussion; those used in the actual implementation will be more complex.

In the following, we will show how the features of KL-ONE allow us to organize the set of descriptions for the objects of interest to the TDP/TDI system (such as platforms, sensors and weapons). This organization allows us to efficiently store substantial amounts of information with minimal redundancy, and to gain access to it where it is needed in the operation of the various subsystems of the TDP/TDI system. The KL-ONE organization of long-term knowledge is a classification network or taxonomic lattice; the TDP/TDI system will use this network in analyzing particular tactical situations by:

- classifying platforms, platform activities, weapons and sensor states, tactical situations, etc., into patterns which have important implications to the operation of the system
- drawing inferences such as potential threats and counters in a given situation
- performing simulations, that is projecting the likely future states of affairs
- representing the intermediate states during a simulation so that reasoning processes (such as those mentioned above) can be performed

The taxonomic lattice organizes descriptions, bringing together those which must be treated similarly by various system components such as projection, threat assessment and display. Thus, for a typical NTDS-type display, we wish to distinguish kinds of vehicles (e.g. surface, aircraft and submarine) and to whom they belong (e.g. enemy, friendly and neutral), while for simulation we must distinguish between nuclear and conventionally powered vehicles because of inferences based, for example, on cruise range. KL-ONE allows arbitrarily complex descriptions of platforms, sensors, situations, and so forth to be formed and used. It also provides mechanisms whereby only those portions of a complex description relevant to a given process need be accessed during the operation of that process. Thus, distinctions can be made when they are critical, and ignored when they are irrelevant.

We will also show how KL-ONE might be used to represent various types of information needed by the TDP/TDI system, concentrating on the relevant features of KL-ONE and not the details of the Naval information. We will first discuss the ways in which KL-ONE stores long-term knowledge, and then the way in which specific tactical situations might be represented.

KL-ONE provides a means for distinguishing between objects, referred to as Nexuses, and descriptions which can apply to objects, referred to as Concepts. The descriptions are hierarchically structured. Each description may have component descriptions, which themselves may have components, indefinitely. Therefore, examining the concepts defined in Figure 5, the description of a Vehicle may have several components such as a Propulsion Unit for a vehicle, the Emitters on the vehicle (only one is shown in the diagram), and so forth. Each of these components may in turn have components; the propulsion unit may specify the power source, the propulsion mechanism, and so forth.

Complex descriptions may be built up from other descriptions using inheritance of various descriptive features. Having characterized the class of vehicle descriptions as is shown in Figure 5, we can expand that description (see Figure 6) to include a definition of the class of Platform descriptions where Platform is a Vehicle that contains sensors and weapons. The fact that a platform has a propulsion unit and various emitters is inherited because of the sub-concept relation between the concepts of Platform and Vehicle.

Also, classes of sub-structures may be differentiated, as is shown in Figure 7, a further enhancement of Figure 5. In this diagram, the Sensor role of Platform can be differentiated into two roles to represent the fact that a platform can have two classes of sensors, active and passive. In addition, we can show that the active sensors are also a type of Emitter; a Radar, therefore, is not only a sensor, but also a source of emissions from the platform.

KL-ONE permits us to define sub-classes of objects based on restrictions on particular attributes. Figure 8 shows that a Nuclear Powered Vehicle is one whose Propulsion Unit is Nuclear. We can indicate that a particular class of objects is exhaustively sub-categorized so that, for example, all propulsion units are either nuclear powered or conventionally powered, as in Figure 9.

Each concept appearing in the hierarchy may have roles which do not appear on any higher concepts. This allows us to indicate that particular properties are relevant only to certain classes

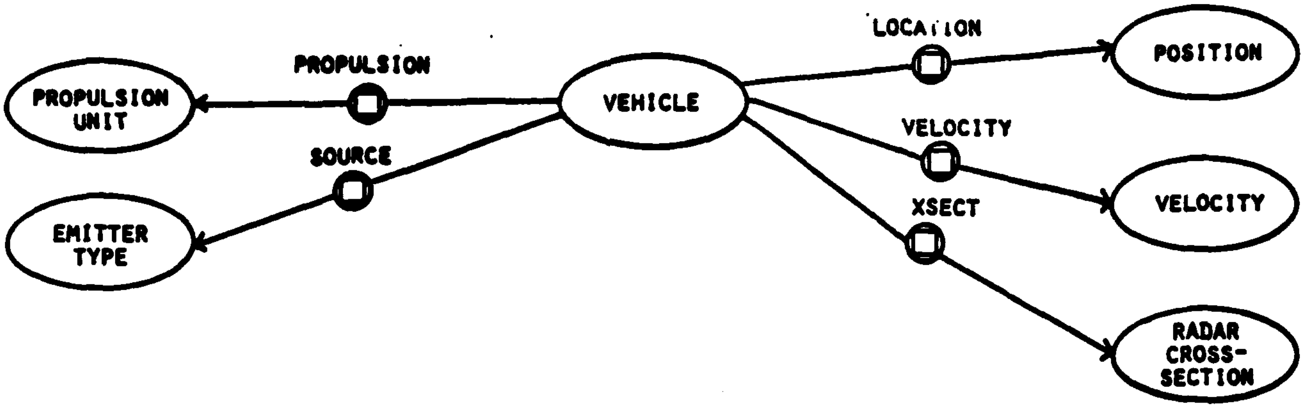


FIGURE 5. STRUCTURED DESCRIPTIONS -- HIERARCHICAL DECOMPOSITION

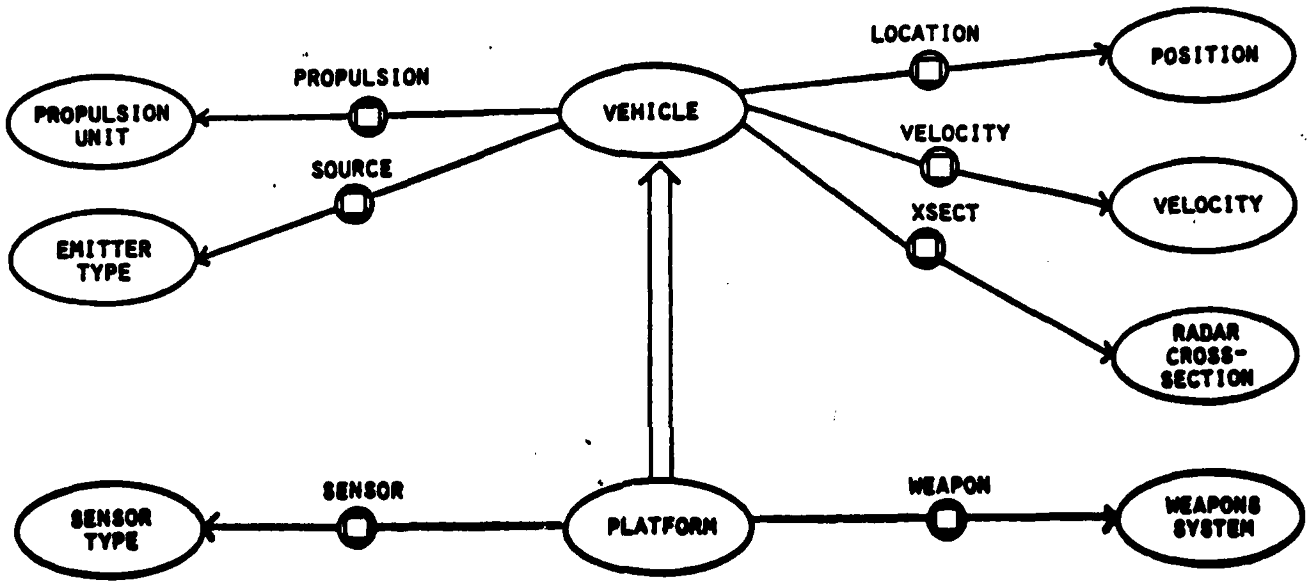


FIGURE 6. DESCRIPTION INHERITANCE AND STRUCTURE EXTENSION

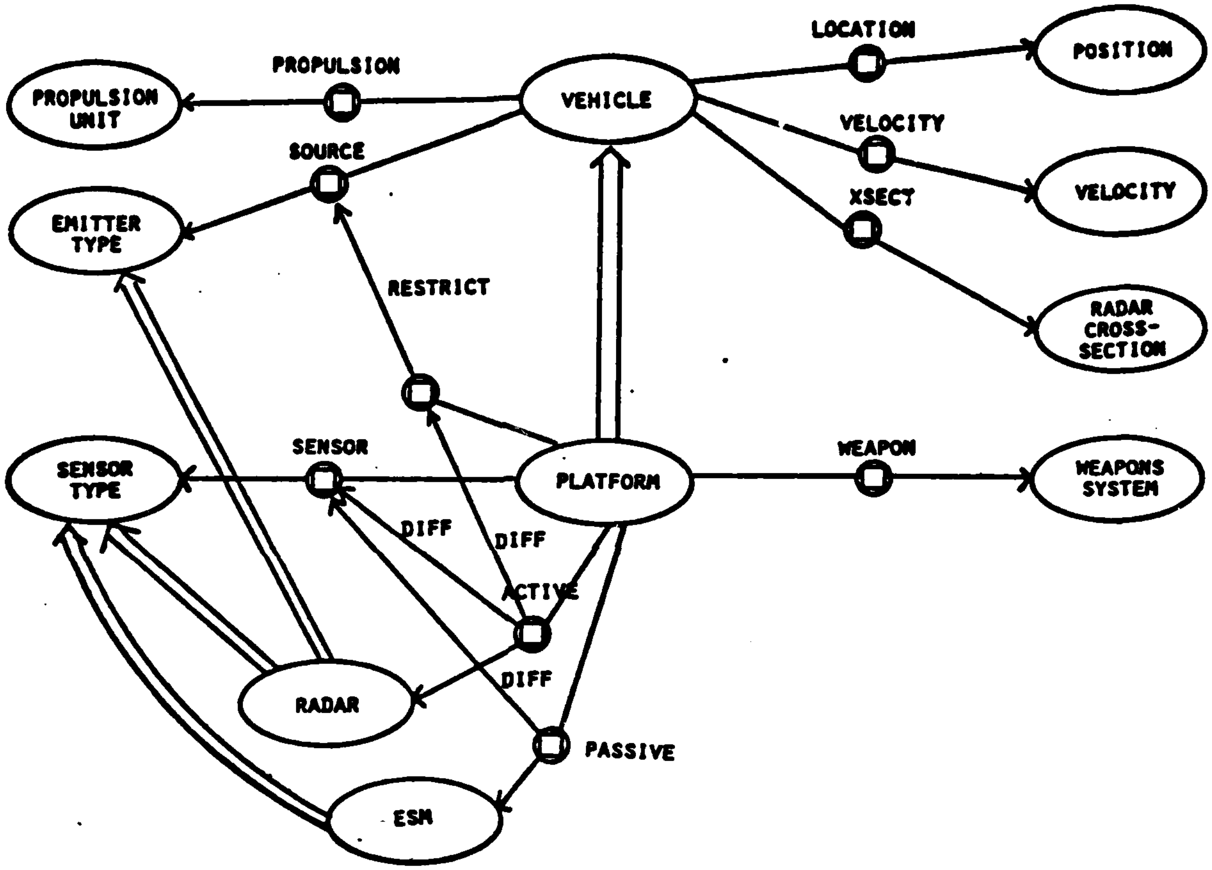


FIGURE 7. DIFFERENTIATION OF SUB-STRUCTURES

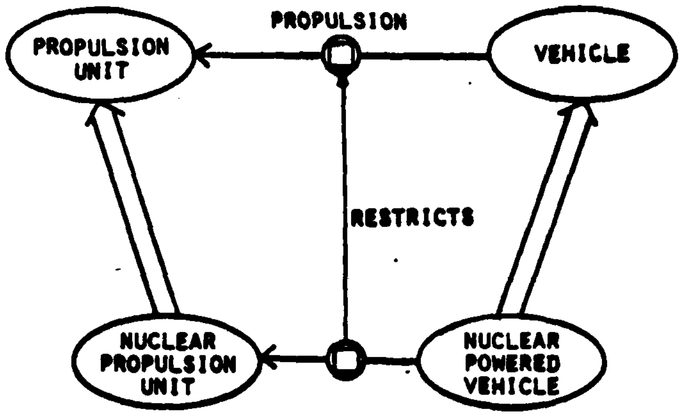


FIGURE 8. SUB-CATEGORIZATION BASED ON ATTRIBUTE

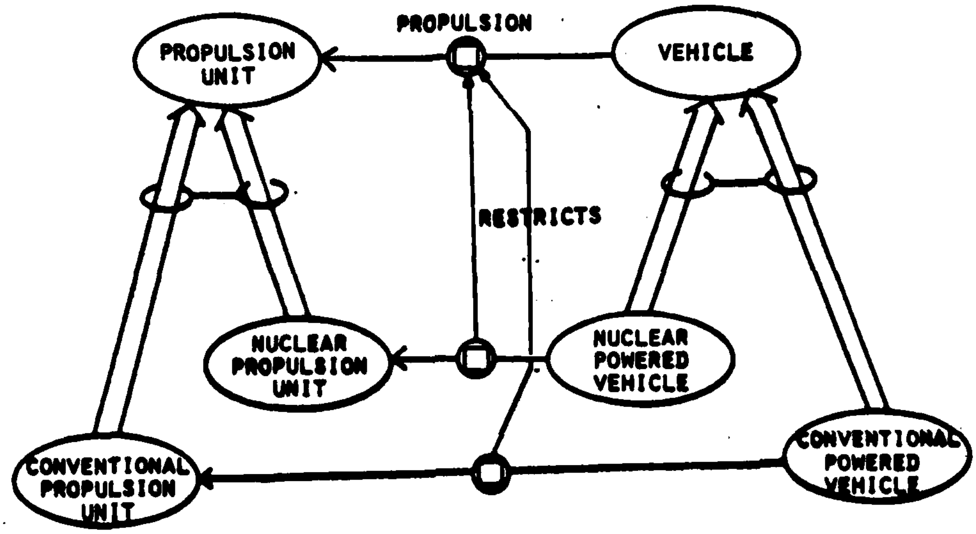


FIGURE 9. PARTITIONS

of objects. Therefore, for practical purposes, only conventionally powered vehicles need be described in terms of cruise range which is based on the fuel state in relation to fuel capacity, as is shown in Figure 10.

A single concept in KL-ONE may be exhaustively sub-categorized in several ways, as can be seen in Figure 12. Vehicles may be either Nuclear or Conventional powered. They may also be characterized as Aircraft, Submarine, or Surface, as in Figure 11. The ability of a single concept to inherit from more than one super-concept allows us to produce the notion of a Nuclear surface vehicle.

The features which we have just described allow us to organize the set of descriptions for the objects of interest to the TDP/TDI system, such as platforms, sensors, and weapons, and to store this information with minimal redundancy.

So far, we have only discussed descriptions applicable to classes of objects. The ability to form these into a hierarchical taxonomy with minimal redundance provides a good technique for representing long-term information for the various classes of objects which must be dealt with in the system. As important as this long-term knowledge is, the system must be able to reason about specific situations in which particular platforms are maneuvering. KL-ONE allows us to distinguish between the objects which are present in particular situations from the descriptions of those objects. This is critical because a single object may have many descriptions at any given time; some of these may be applicable for a very short time while others may be applicable throughout an entire tactical situation.

At one point in a tactical situation, a platform, say a Russian missile cruiser, may have several descriptions, such as:

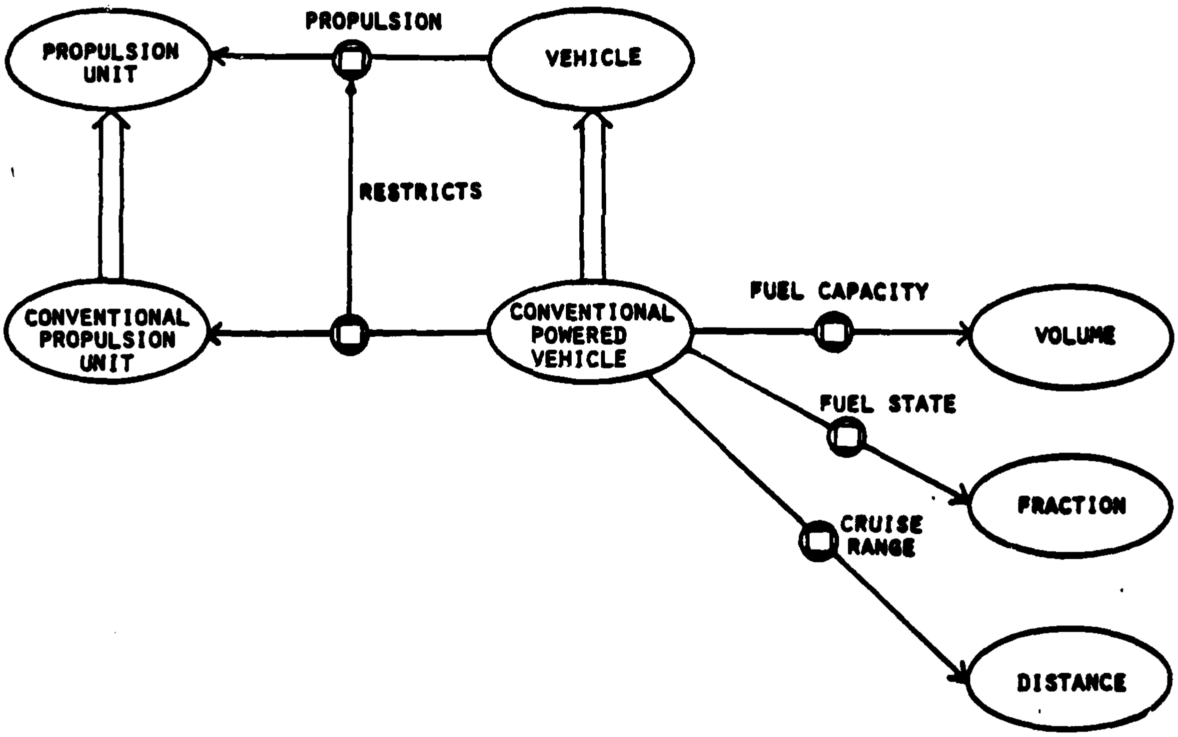


FIGURE 10. NEW ROLES FOR SPECIALIZED CONCEPTS

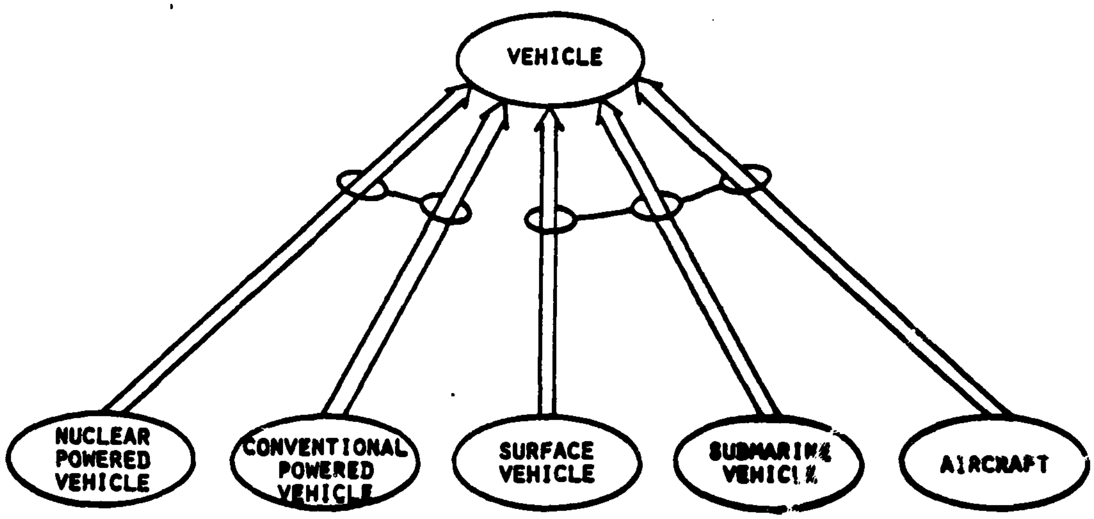
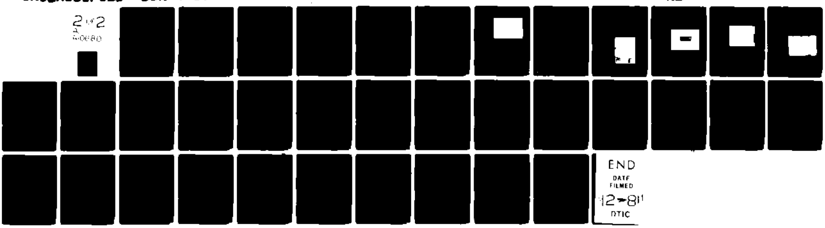


FIGURE 11. MULTIPLE PARTITIONS

AD-A106 801 BOLT BERANEK AND NEWMAN INC CAMBRIDGE MA F/G 15/4
USING AI TECHNIQUES FOR THREAT DISPLAY AND PROJECTION, INCLUDIN--ETC(U)
SEP 81 J VITTAL, R BOBROW, O SELFRIDGE N00039-80-C-0659
UNCLASSIFIED BBN-4789 NL

2 of 2



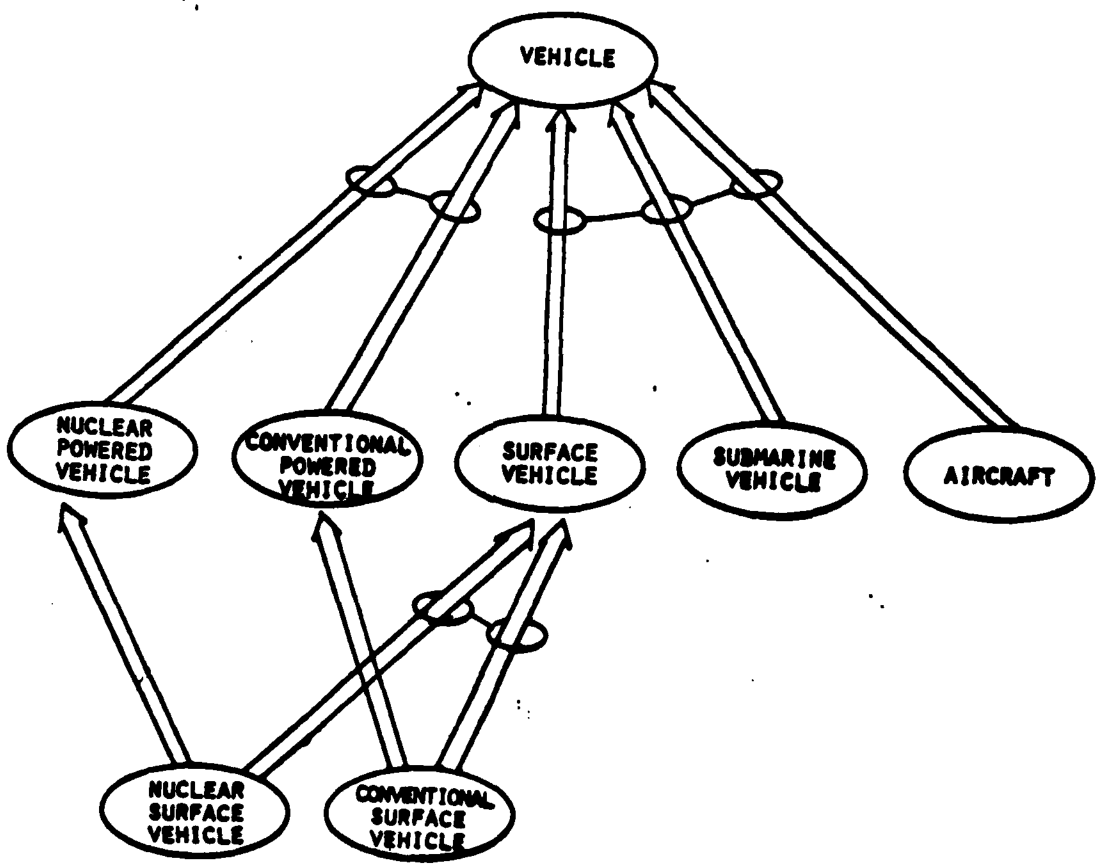


FIGURE 12. LATTICE

- A Kresta II class cruiser
- The flagship of battlegroup 17
- The source of radar contact R38
- A platform moving at 18 knots, bearing 135
- The platform located at position P_1.

The fact that the object in question is a Kresta II class cruiser will remain true throughout the tactical situation; all inferences which can be drawn from this may be used at any time. Other properties, such as the fact that it has a particular speed and bearing, will be true for an indeterminate amount of time, and can be used for inferences such as dead reckoning. Finally, descriptions may be valid only for a particular instant in time, such as the position of a moving platform.

In many respects, knowledge-based simulation as we view it is similar to object-oriented simulation as is practiced in Simula [2], Smalltalk [32] and ROSS [25]. The operation of the simulator is mediated by the passing of messages[8] among objects, such as the fact that a given aircraft has just entered the range of a particular air-search radar. In an object-oriented system, both the fact that an object receives a message and that object's response to the message is determined by the class to which the object belongs.

The class of object does not vary over time although certain parameters such as position and speed may vary. This does not

---

[8]We should note here that the term "message" is a formal computer science term and not normal Navy message traffic, such as Rainform.

provide a convenient way of representing the fact that the response to a message may depend on the role which that object is playing in the given simulation, such as its tactical assignment. For example, the response to a radar message will differ for a ship assigned to handle CAP, than for the flagship. Using the distinction made in KL-ONE between objects and their descriptions, we can associate patterns of behavior with classes of descriptions. Thus, when an object receives a message, its response will depend upon the descriptions applicable to that object at the time the message is received.

Another distinction between the knowledge-based and object-oriented paradigms is the representation of instantaneous states during the simulation. The existence of a taxonomy of descriptions allows us to classify a given tactical situation within categories defined by long-term knowledge.

6. SCENARIO

A major effort of this contract was the development of an operationally relevant scenario, the production of a breadboard surface-level system to show how a computer system for that scenario might look, and then the use of that system as the basis of a videotape.

The scenario in Section 6.1 below is a transcript of the narration of the videotape with some pictures of parts of the relevant screen images at various critical points. A short description of the language used to describe scenarios and implement them in a highly surface-level fashion follows in Section 6.2.

6.1 Scenario

6.1.1 A Short Introduction by Robert Kahane

Mr. Kahane: I am Robert Kahane, a program manager in code 613 of the Naval Electronic Systems Command, responsible for Command, Control and Surveillance systems R&D.

In the operation of Combat Information Centers, operational personnel can easily be confused by the flood of data that pours into the CIC in times of crisis, such that good decision-making is hardest just when it is most needed and most urgent. Soviet tactical deception techniques have been developed to take advantage of this situation and to cause decisions to be made that could hamper effective US operations. The intent of the system being demonstrated here is to identify possible threats, taking into account deception techniques used by the enemy. The Soviets have a strong doctrine in deception; this is a demonstration of what a system to recognize and to counter that doctrine might look like.

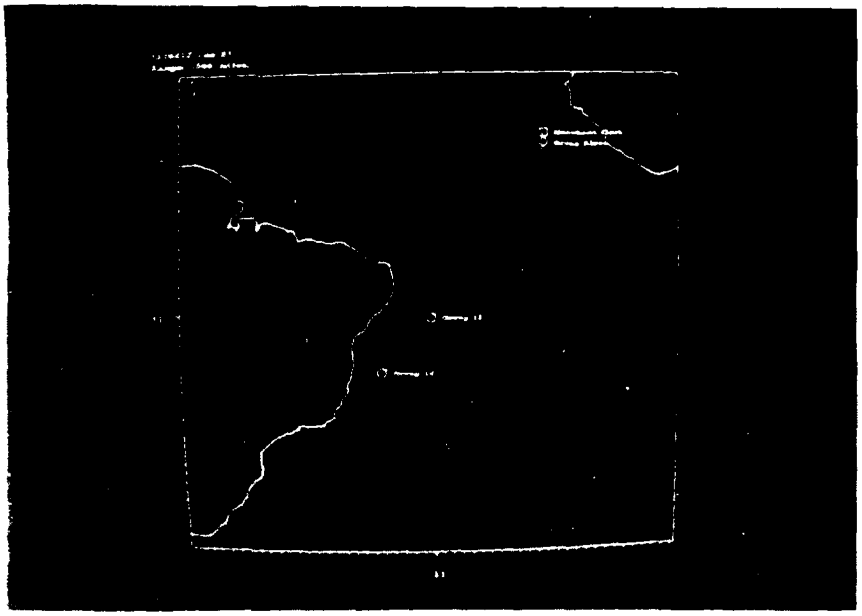
What you are about to see, then, is a demonstration that simulates the behavior of a system to do Threat Display and Projection with Tactical Deception Indication. We have plans to implement such a system, similar to that which is being demonstrated. This system will be based on the application of technology from various areas of computer science, including real-time graphics, digital communications, networking, simulation, and artificial intelligence.

This project is associated with the joint Navy/DARPA Ocean Tactical Targeting (OTT) program, the Integrated Tactical Surveillance System (ITSS), and also has applicability to the Outlaw Shark program currently being implemented in the fleet. These programs are generally aimed at the problem of threat identification by means of the fusion of data obtained from many sources, such as OTH (over-the-horizon) radar, HFDF, SOSUS, and national collection systems. Much of the demonstration which you are about to see assumes afloat access to multiple sensor data from organic and remote sources, and afloat correlation capabilities which are projected under current ITSS program alternatives.

The scenario to be shown will contain three parts. The first provides an overview of the tactical situation. The second shows the use of simulation and how new data can be added to the databases and utilized for decision making in real time. Finally, the third part shows the automatic recognition and indication of tactical deception.

6.1.2 Scene 1 -- Introduction

<screen>	long range view of a 1500 mi. radius.
Narrator	What we are going to do in this videotape is show an example of the use of a command and control system that includes tactical knowledge-based simulation and tactical deception indication. We will be showing an action officer using the system and interacting with others around the CIC and Flag Plots; however, the focus will be through the terminal screen and the narrator who describes the interactions. The carrier *Nimitz* is the flag ship of the bigger of two battle groups that are supposed to be a "presence" in the South Atlantic; the Constellation is the other flag ship. These two groups have the mission of deterring the Soviets from trying to supply a possible communist coup in Buenos Aires with their weapon systems, including ICBM's. This is not just a possibility, because overhead surveillance about 30 hours ago photographed a guided missile cruiser with three destroyers a few hundred miles SSW of Dakar, slowly heading south; behind them by 50 miles were half a dozen merchant ships. This situation is depicted on the screen. The last track update came in from the shore 12 hours ago, and they are estimated to be maintaining course and speed.



There is, in effect, a Soviet group heading for a confrontation with an American group; and the Bermuda listening arrays have picked up four high-speed contacts evaluated as Soviet nuclear submarines. They are proceeding towards the projected CPA in mid-Atlantic. The assumption is that the location of Nimitz' group is not known exactly to the Soviets; there are no subs underneath the group, there are no following Soviet trawlers, and the group observes tight EMCON procedures when they are within Soviet satellite windows.

Our group, consisting of seven ships, is shown on a course of 045 at 20 knots, 400 miles off the big curve that marks the most easterly point of Brazil.

<screen> Change range of display.

Narrator We now join the staff in the Nimitz' CIC. What you are looking at is the actual computer terminal screen displaying the current situation within a range of 200 miles; the situation display is centered on the Nimitz. Lcdr. Stochowski is the flag CIC Watch Officer; the time is just midnight. Lt. Crichton has just walked into the CIC to replace Stochowski as flag CIC watch officer.

<screen> Crichton officially takes over from Stochowski, by the simple process of typing a command.

Narrator The system automatically displays the status of some of the more important, global issues:

- There has been no update on the rules of engagement; the admiral is sending a message to Washington every hour asking for one. The strategy is to stay out of enemy weapons range until ROE and mission objectives are clarified.

- There is a substantial backlog of intelligence to be processed.

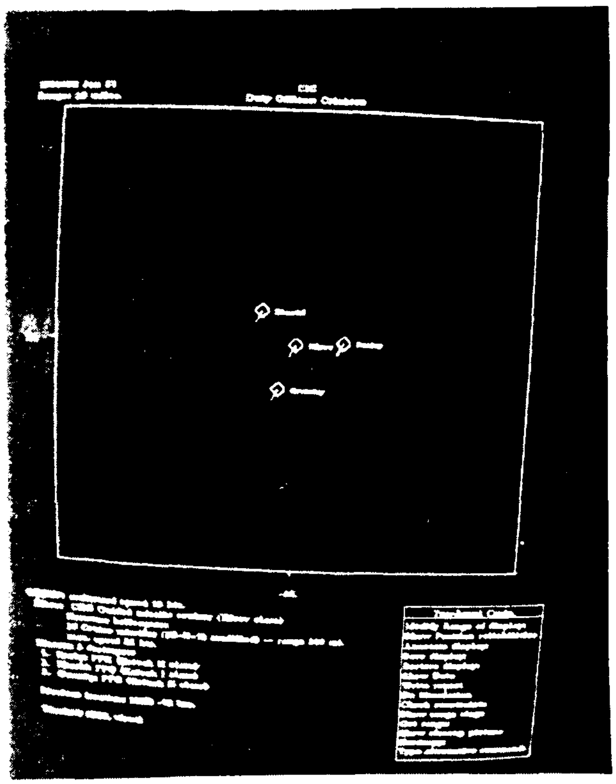
- A lot of ELINT reports have come from ashore. One of Crichton's jobs will be to update the computer databases with this data when he can.

- An E-2C was launched at about tango -20, and it'll be on station at a range of 150 miles in tango +10. His data is coming in now and is being entered into the system.

- One report indicates that there is a Soviet group closing in; the last simulation at tango -45 projected interception with the Soviet group at tango +270 based on a 18 kt. SOA. The group had reported on uncovered tactical voice that mechanical difficulties limited SOA to 18 kts.

The next command typed by Crichton will switch the display back to a longer range, 500 miles. A Soviet group (including a guided missile cruiser), is bearing about 040, about 450 miles away, heading towards our group, again with the Nimitz at the center of the display. The Constellation's group is about 400 miles southwest; they are heading to the same projected CPA in mid-Atlantic.

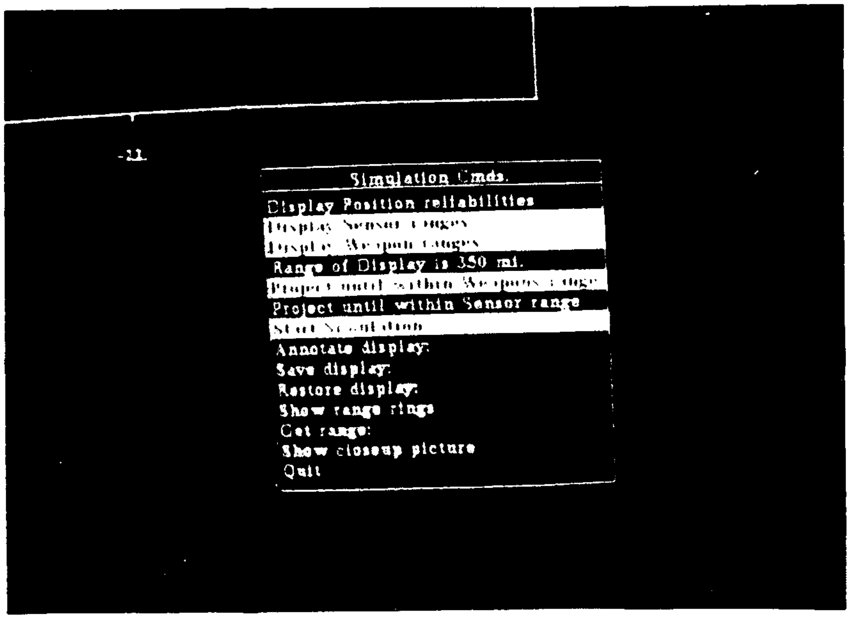
Crichton types a command to check on the identity, location and bearing information on the Soviet group. Various data is typed out indicating a short-term history of where the ships have been, how they have been tracked (OTT, Outlaw Shark, HFDF, sighting reports), and some characteristics of the vessels themselves (identity, threat status, sensor status). In addition to the textual data, their formation is also displayed.



6.1.3 Scene 2 -- Simulation and system interrupts

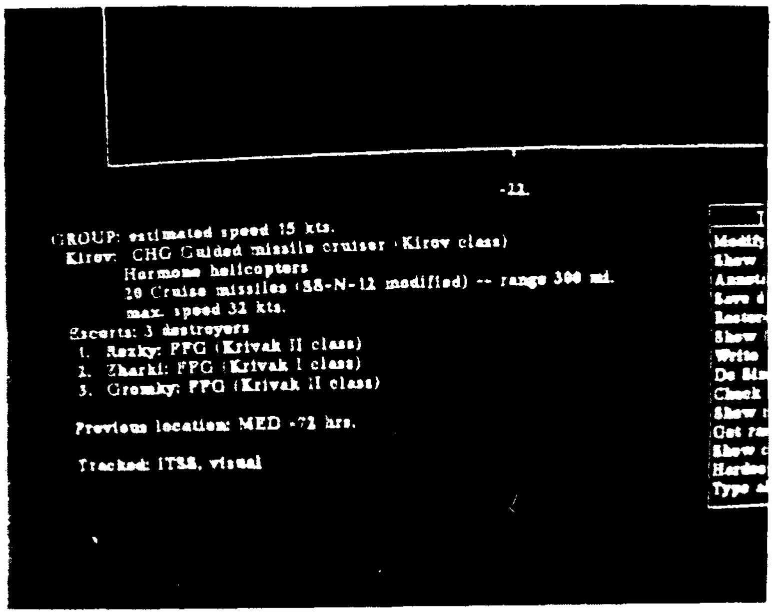
Narrator — In order to project the threat of the Soviet group to the Nimitz' group, Crichton performs some simulations -- that is, to project what the situation will be sometime in the future.

<screen> — Crichton types several commands.



Narrator — What these commands are doing is asking the system to estimate when the battle group will be within enemy weapons range. The simulation display should include all vessels with sensor ranges and weapons ranges indicated. The display is centered around the Nimitz' battle group.

The resultant display shows the group with several dashed circles around it, representing the maximum ranges of their several sensors and weapons, the projections being based on their known characteristics and projected status. This projects the Soviet positions at tango +250. Notice that the largest circle, which corresponds to the weapons range of 300 miles of the Soviet group, is over the battle group at the center. The upper right hand corner of the display shows the simulated time. Some relevant threat information is also displayed, including the kind of weapons and some of their characteristics.

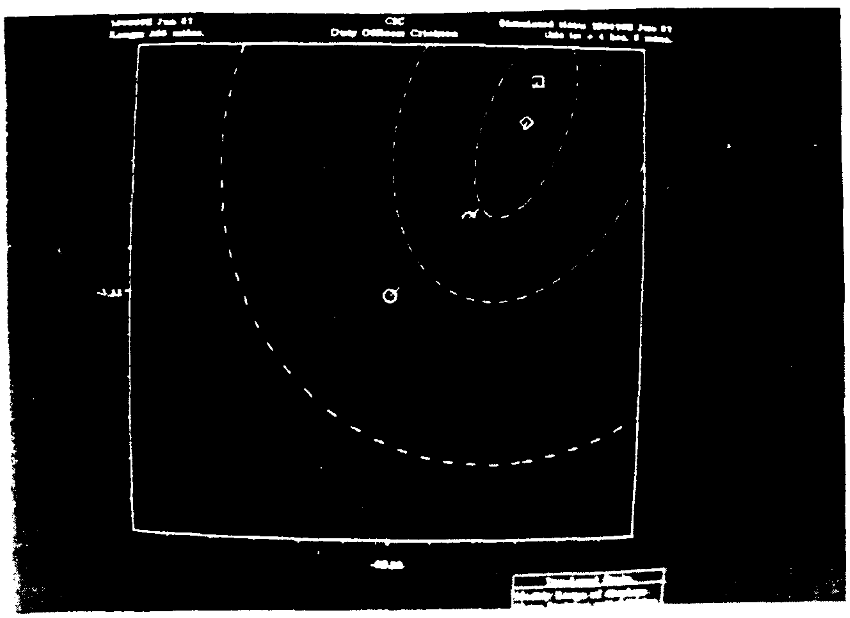


Crichton types another command to save this screen image, or 'frame', for later reference.

Crichton now asks for another simulation: when the group will be within enemy sensor range. The circles showing sensor range capabilities are across the closest battle group vessels at a range of 100 miles. The weapons range of 300 miles from the Soviet projected positions is also shown; the threat tables from the later frame, showing vulnerability to the Soviet missile cruiser, are also displayed. Crichton then stores this frame too.

He leaves the simulation sub-system, recovers the display for the first simulation and asks for the reliabilities of the positions to be displayed on the screen; around the Soviet group appear three ellipses, with the long axes along their tracks. The innermost ellipse is the area of the probable location of the Soviet group; the middle ellipse indicates that groups probable sensor range based on the location probabilities; the outer ellipse shows its weapons range based on the location probabilities.

Then he returns to the display of the present situation and asks for a projection for when the Nimitz' group will be within enemy sensor range, with the displays showing probability ellipses. The outer ellipse is the range of the Soviet weapons, and completely covers the Nimitz' group; the inner ellipse shows the probability for the location of the Soviet group; the middle ellipse, indicating the probability range of the Soviet sensors, shows contact at tango +75.



As before, he stores the frame, and leaves the simulation sub-system. Then he causes the frames that have been saved to be put out in hard copy for the admiral.

Crichton now begins to prepare a report for the admiral by writing short annotations on the frames. He is first interrupted by the intelligence officer, who tells him that Norfolk has updated their estimated Soviet positions from the fusion data -- this one is is from WARF and HFDF. He also indicates that there are two TARFS coming in on the submarines, and that all the data will be added to the database as soon as the rainform traffic is processed.

The system interrupts with the information that ELINT reports from the E-2C show some contact at a range of 300 miles from the Nimitz; the actual data is being added to the database.

Crichton calls Commander Ridley in flag OPS and tells him he should probably wake the admiral; the report that he started will be ready for him soon.

<screen> Warning message from E-2C indicating a slight drop in oil pressure in one engine, and that it needs to return to home plate. It will, however, wait until a new E-2C can be launched, if possible, to maintain some coverage of the Soviet group.

Narrator The next E-2C can be on station in about tango +40; this information is also passed on to Cmdr. Ridley.

Note that the terminal has sounded warning and the screen is showing a notification that the group is within weapons range of the enemy.

<screen>

```
************************************
*  TARGETS UPDATED
*  THREATS EVALUATED
*  CURRENTLY WITHIN WEAPON RANGE
*  ASSUMED TO BE KIROV GROUP
*  ESM SENSORS ONLY
************************************
```

This information is based on the data received from the E-2C.

Narrator — Another message follows: it states the reasons for believing that the group is the Kirov group; the formation is the same and the sensor prints are consistent with the known sensors on board those vessels.

Note that the system has sounded another warning. The Soviets have started jamming the E-2C's radar:

<screen>

```
************************************
*  Soviets Jamming Radar
*  E-2C contact lost
************************************
```

Narrator — Crichton gets a phone call from Ridley. He and the admiral are coming up to the flag plot, and Crichton should join them there immediately. Crichton types a command to transfer control of the system from the CIC to the flag plot, and walks across the passageway.

6.1.4 Scene 3 -- Flag plot, Decision Making, Deception Indication

Narrator — The Nimitz' flag plot has a system which is identical to the one in the CIC.

Immediately after Crichton sits down, Admiral Tordella and Cdr. Ridley come into the room and listen while Crichton brings them up to date, showing them various plots on the screen, by retrieving the frames he had previously stored.

Ten minutes after he entered the flag plot, as Crichton was finishing the report, an enlisted man enters and hands the admiral a message labeled "SPCAT, exclusive for Adm. Tordella."

The admiral tears it open. The NMCC indicates that the White House wants an estimate of the on-scene situation, now.

A message arrives from the E-2C that it is returning to home plate, but will do so as slowly as possible. The replacement E-2C has been launched; there will be a gap of at most a couple of minutes in the sensor coverage. The system indicates that the Soviets are still jamming, based on data coming in from the E-2C.

The Admiral orders the formation to go to condition one, with EMCON plan Alpha modified.

Ridley is ordered by the admiral to draft a noncommittal response to the NMCC message.

Narrator — We rejoin the people in the flag plot some 10 minutes later -- 21 minutes after the Soviets started jamming....

The screen indicates that ESM contact has just been regained; a display will be available shortly.

The display shows that the second E-2C is on station about half way between the two groups of ships. It has a clear radar view of both of groups, and is also picking up ELINT.

The display shows that the Soviet formation has reversed, however, no directional information is available as yet.

The Soviets have started jamming again some 15 seconds after they stopped. There was not enough contact to obtain information about course.

The admiral asks Crichton to check how the time/speed equation works out for the last Soviet maneuver. Also, he requests the E-2C to try to get some lines of bearing on the Soviet group.

<screen> — Crichton types some commands to the system.

Narrator — The result of the analysis is that the Soviet cruiser can do 32 knots, and it looks as if she made 28 to complete the maneuver in the formation. They are well within the range of possibility for reversing course.

The admiral asks Crichton to check for tactical deception before he releases the current SITREP to the NMCC.

In the mean time, a report has come in from CINCLANTFLT Intelligence Support Center in Norfolk suggesting that the whole Soviet force may have turned.

Crichton checks for deception; the system issues a positive TDI; you can see it on the screen.

<screen> — Shows TDI warning...beeping

```
****************************************
*  POSSIBLE DECEPTION
*  CRUISER ONE SHAFT SPEED 18 KNOTS.
*  ONE SHAFT REPORTED OUT, SS R 181724Z.
*  PREVIOUS CONCLUSION IN ERROR:
*  TIME SPEED EQUATION PROBLEM
*  POSSIBLE DECEPTION
****************************************
```

The speed is not commensurate with modifying positions; straight course reversal could have occurred.

Narrator — Notice that the Soviet cruiser casualty report is two days old. It is not on the status board, and apparently no update has been received. The admiral orders the group to reverse course to 225, speed 25.

<screen> — The screen flashed another warning.

```
******************************************
*  POSSIBLE DECEPTION
*  RADAR CHARACTERISTICS ANOMALY: TARGET
*  KIROV ZIG SEARCH RADAR:
*    Freq: 3115   PRI: 127   PRF: 61
*  CURRENT SIGNAL:
*    Freq: 3117   PRI: 129   PRF: 62.5
*  POSSIBLE DECEPTION
******************************************
```

Narrator — After examining the message, the admiral asks CIC about the actual radar blips. Before the Soviet maneuver, they seemed to be much smaller.

As a result of this information, realizing that one way to increase a visual signal in such a fashion was to put up corner reflectors on the escorts, the admiral orders a speed increase to 30 knots, and also orders the carriers to launch aircraft and marshal them overhead.

Realizing that the Soviet commander is employing tactical deception, the admiral then orders Crichton to check the deception history file.

<screen> — Crichton commands the system to check the deception histories against the current situation.

```
******************************************
*  POSSIBLE DECEPTION
*  DECEPTION HISTORY FILE:  poss. match
*  SOVIETS OBSERVED IN EXERCISES
*  WITH CORNER REFLECTORS TO VARY ERP
*  OF VESSELS.  COMMON USAGE IS TO HIDE
*  LARGE SHIP IN FORMATION OF SMALL
*  SHIPS  (see TDHF for examples)
*  POSSIBLE DECEPTION
******************************************
```

And almost immediately afterward...

```
*******************************************
*  POSSIBLE DECEPTION
*  Deception History File: poss. match
*  Soviets Observed in exercises to
*  use jamming and speed changes to
*  obscure possible direction changes.
*  POSSIBLE DECEPTION
*******************************************
```

Narrator — Notice that additional projections about when the groups will be out of range of each other are automatically displayed. In tango +15 the situation will be static, with the groups going in parallel courses southwest, but out of missile range of each other. The situation is still critical, but one confrontation has been averted.

6.1.5 Discussion

Mr. Kahane — The importance of the scenario that you have just seen is that deception efforts can often be detected by an intelligent application of checks and cross checks; we believe that such an application can be handled by advanced computer science techniques as simulated here. In the scenario which you have just witnessed, the purpose of the deception was to place the guided missile cruiser -- as far as our forces knew -- 20 miles behind where it really was. They attempted the deception during EW operations (a common Soviet tactic) in a way that has been observed historically.

During any gap in sensor coverage, maneuvers can be checked to see whether they call for capabilities beyond what targets are known or estimated to possess. Note that the mere calculations of plausibility are not difficult. The more difficult task is to assemble and integrate the data that exist, in different forms and representations, and to decide which of all the data are relevant. Our interest here is not in classifying the possible forms and uses of deception, but in designing and testing prototypes of useful tools that can aid in identifying such C&D operations.

The ability to determine accurately and rapidly when C&D operations are in effect, and to ascertain precise enemy movements, positions, and intentions is vital to effective Naval operations.

6.2 Scenario Description Language

6.2.1 Introduction

A breadboard surface-level implementation of a scenario is based on a language for specifying actions on a Jericho screen, and an interpreter for that language. This section describes, briefly, the language.

The scenario language is based around the notion of a screen, named displays, text, and menus. Named displays are objects to which data can be added or deleted in background, and then displayed. A single screen is assumed to be separated into three separate areas: a display area, an area for menus, and a text area.

6.2.2 Scenario language

An implementation of a specific scenario consists of a list of commands. Each command is also a list, specifying the operation to be performed and the argument(s) for that particular operation. The CAR or first element of the list is the name of the command or operation; the CDR (which is the tail or everything but the first element of the list) contains the parameters for that operation. The following are the available operations:

*	A comment. The command is ignored.
ADDTODISPLAY	Takes two arguments: NAME and DATA. NAME is the name of the display to which the data is to be added; DATA describes that data: (location distance data). The location can be any of LEFT, CENTER, or RIGHT. distance is the number of points from the top of the display stream associated with NAME, and data is the string to be added to the display at that location.
CLEAR	Takes as an argument a specification for the area to be "blanked." The argument can be:

WHOLE	the whole screen
NIL or DISPLAY	the display area of the screen
T or TTY	the text area of the screen
name	a named display

COMMAND	Obtains a command from the user by requesting the command from the text area.
CREATEDISPLAY	Creates a named display, associating with the name a bitmap and a display stream. The arguments are NAME, WIDTH, and HEIGHT, in that order. If HEIGHT is NIL, then it is assumed to be the same as WIDTH. If WIDTH is NIL, then WIDTH and HEIGHT are assumed to be 511.
CREATEMENU	Creates several named menus. This should be invoked exactly once as the first scenario command, especially before the INIT command.

The description information for the menu is to be found on the property list of the name under property MENUSPECS. This property contains two elements. One is the caption, and has the form (CAPTION "caption string"). The other is the contents: (CONTENTS (list of contents)). Each element of the list of contents is a unique atom (item name) for the particular menu and the string to be displayed in the menu corresponding to that item.

DATE — Displays the date in the upper left corner of the named display.

DELAY — The argument is the number of seconds to wait before continuing.

DELETEFROMDISPLAY — The opposite of ADDTODISPLAY. Takes the same arguments.

DELETETYPEAHEAD — Clears the keyboard input buffer.

DISPLAY — Copies the contents of the named display onto the screen. If the first argument is the atom TEXT, then the second argument is the name of a display that should be shown in the text area. Otherwise, the first argument is the name of a display that should be shown in the display area.

EVAL — Just evals its argument.

HELP — Calls the function HELP.

HILITE — Hilites a menu item; that is, displays it in inverse video. It is assumed that the relevant menu has already been displayed with the MENU command. The arguments are MENU-NAME and ITEM.

INIT — The argument is the number of screens available. Presently, only one screen is available. Performs all of the initialization of the size of the areas.

INITVESSELS — Adds a visual image of a set of vessels to a named display. The name of the display is the first argument, the descriptor for the vessels is the second argument. Each vessel has (potentially) several attributes associated with it in the descriptor:

descriptor	meaning or possible values
WHO	OWNSHIP, SURFACE, AIR or SUB
TYPE	FRIEND, FOE or NEUTRAL
NAME	the name of the group, flagship of the group, or specific vessel.
LONGITUDE	in degrees
LATITUDE	in degrees
DIRECTION	in degrees; 0 is to the right, and increases counter-clockwise.
SPEED	in knots
RADAR	range in miles of the radar of the vessel
WEAPONS	range of the vessel's weapons in miles
SEMIMAJORRADIUS	one of the descriptors for a probability ellipse.
SEMIMINORRADIUS	the other probability ellipse descriptor

Draws a map on the display, and shows the vessels in the right places, centered around the OWNSHIP. WHO, TYPE, LATITUDE and LONGITUDE must be present. The others, if present, add data to the display accordingly.

INVERT Inverts the videosense of the named area, clearing it first. The area can be any of the arguments that CLEAR takes.

MENU Displays the named menu.

MOUSE	Goes into an IO wait for the mouse.
NOHILITE	The opposite of HILITE.
NOMENU	Undisplays a menu, redisplaying whatever was underneath it at the time it was originally displayed.
RANGE	Sets the range of the named display. This governs the area covered by the display, the granularity of the legend, and the size of the vessel icons.
READ	Reads an expression from the terminal.
SAVE	Saves either a named display or the text area of the screen. The first arguments are the same as for the DISPLAY command, and specify the source of the save operation. The last argument is the name of the "display" into which the data is to be stored.
SIMDATE	Displays the simulated date (used after a simulation run) in the upper right corner of the named display. The second argument is the date to be displayed, and the third is a descriptor for how far off the simulated date is from the "current" date: (days hours minutes).
SOUND	Makes a warning sound.
TABLE	Clears the text area of the screen, and writes a table into the area. The columns and rows are separated by lines. The width of each column is determined by the maximum length of the contents of the constituents of that column. The arguments to the TABLE command are the data to be displayed. The first argument is the title (caption) of the table. Each of the other arguments is a list corresponding to one row of the table; each element of the list is an element of the row. To leave out an element of a row, NIL should be used.
TEXTDATA	Writes text into the text area of the screen. Its arguments are the data which is to be displayed. T is treated specially to mean "carriage return".

7. RELATIONSHIP OF THE TDP/TDI SYSTEM TO THE NAVY

The intent of the TDP/TDI system, as has been brought out in the preceding chapters, is to identify possible threats, taking into account various deception techniques used by the enemy; the Soviets have a strong doctrine in deception.

One of the major parts of this effort is the fusion of data obtained from many sources, such as OTH (over-the-horizon) radar, HFDF, SOSUS, and other national collection systems, in addition to local battlegroup sensors. This project is associated with several current and projected R&D programs:

- Integrated Ocean Surveillance (IOS). This uses rainform traffic to provide track information. This is primarily used for strategic purposes.
- Ocean Tactical Targeting (OTT). This also provides track information, but is more localized than IOS and is used for tactical purposes. It uses IOS as a baseline, but adds acoustic data as a data source to enhance processing. This is a joint Navy/DARPA program.
- Outlaw Shark. This is very similar in scope to OTT, and is currently being implemented in the fleet, but is limited in scope by technological considerations.
- Integrated Tactical Surveillance System (ITSS). A proposed system which is an integration of many current and evolving technologies, including, possibly, satellite-based sensors. A preliminary conceptual system architecture should be ready by March 1982.

In all of these programs, various databases will be accessible giving the identity, position, track and position reliability information for all known platforms in a given region. Currently, we know of no access protocol to allow retrieval from these databases of information relative to specific localities.

Outlaw Shark is primarily a broadcast system intended to provide targeting information to submarines (although it can be used more generally); information updates are provided automatically or on request, and are relative to specific theatres of interest.

The TDP/TDI system assumes afloat access to multiple sensor data from organic and remote sources, and afloat correlation capabilities which are projected under current ITSS program alternatives. It is intended to provide mechanisms for display of that correlated data, characterize the extent and nature of threats, perform projections about likely futures, and reason about and display the results of possible tactical deception ploys being implemented in a specific situation by the enemy. Therefore, it is more ambitious than any of the other systems with the possible exception of ITSS. At the present, it is primarily a concept study and trial implementation rather than aimed at the production of an operational system to be deployed in the fleet.

Another major part of this effort is to provide commanders with a readily accessible simulation tool. This will aid them in the evaluation of alternative plans and scenarios, in projecting the current situation into the near future in order to test possible responses, in recognizing potential threat situations before they occur, and in gaining practice and experience with C2 situations and decisions. To the best of our knowledge, we are the only Navy program aimed at this problem; one of the aims of the TDP/TDI system is to provide such a tool which is integrated with the aspects of data fusion and threat display. Everything that we have learned in this contract from talking with Navy personnell is that such a tool, if provided, would be very useful both in tactical situations.

8. RECOMMENDATIONS FOR FUTURE WORK

The design study which has been reported here clearly shows the viability of such a system. It is our belief that this TDP/TDI system will, when completed, provide Naval tactical commanders with a needed capability which will help the US Navy have a tactical superiority on the seas.

Ultimately, the development of an effective TDP/TDI system represents a major technical investment. However, systems which manipulate hypotheses and projections seem an inevitable development as we attempt more effective support of the C2 decision maker.

As discussed in the preceding sections, there are three areas that are critical for the successful implementation of a system for Threat Display and Projection including Tactical Deception Indication. The following sections list the short-term (one year) goals for each of those areas. We believe the research area of knowledge-based simulation will investigate new and almost totally unexplored dimensions for interactive decision support capabilities. It will prove to be one of the major developmental directions for C2 systems during the next decade; it promises a new paradigm for Command and Control.

8.1 Knowledge-based simulation

The initial thrust should be to research and develop a limited, prototype interactive Naval tactical simulative query system (SQS). The approach is to take some powerful tools

(Jericho, Interlisp, KL-ONE, and AIPS or VIEW) and apply these to the problem of simulative inferencing. By building a limited prototype simulation system, the issue of what effect these new tools will have in terms of operational inferencing capabilities for Naval C2 will be explored.

The models of the prototype simulation system will be primarily concerned with kinetics -- the simulation of motion. As it becomes possible to do so, these will be augmented with simple detection and weapons models. Accordingly, an initial task will be to construct a KL-ONE taxonomy of tactical objects. This process will start with physical objects (e.g. various types of vehicles, platforms, missiles). Initially, no attempt should be made to build a descriptive hierarchy for tactical plans or abstract tactical situations.

Another initial task should be to develop a KL-ONE representation for events. Among the issues to be explored are the representation of causal interdependencies among events and the separation of events among distinct contexts of hypotheses.

Once a satisfactory representation of events has been achieved, procedural kinetic models can be developed and attached to the taxonomic hierarchy. The emphasis with these models will be on the prediction of kinetic events or conditions. This differs from most existing kinetic tactical models, which are oriented toward incremental updating of kinetic information. Nevertheless, it will often be possible to borrow model algorithms from existing tactical simulations.

Finally, the simulator must be given an interactive graphic user interface.

8.2 Database implementation and integration

The primary emphasis of this task is to design the various databases needed by the TDP/TDI system, provide trial implementations of several of them, and produce a demonstration showing the extraction of data from the databases for display.

There are several forms of databases which need to be integrated into a complete system for TDP/TDI, and over which the models designed and implemented for the simulation system would operate. The data takes two orthogonal directions: long-term versus short-term knowledge, as has been described, and fused versus raw sensor data. Examining the sources for sensor data should be the first task, including the design of the representation and the design of a conversion mechanism, if necessary, from that data to the internal knowledge-based format used for inferencing. The sources of data include raw sensor data, rainform traffic, and the various sources of fused data such as OTT, IOS, ITSS, and Outlaw Shark.

It is clearly a very large task to design representations for all of the sources, so the two or three which are most critical to the TDP/TDI system should be isolated and representations designed. Sample data needs to be used to test the implementation.

A demonstration system should be built showing the extraction of data from the internal KL-ONE representations and displaying that data in a form which is usable by Naval commanders. This will probably take the form of interfacing to AIPS.

8.3 User-interaction integration

There are two mechanisms by which user interaction with the simulation and TDP/TDI systems can be done: VIEW or AIPS. The problems of interfacing to either of them have already been described. In order to adequately assess the nature of the interfaces and to ascertain which system should be used as the interaction module, the protocols for interfacing to both of the systems should be examined. The issue of implementing the protocol with VIEW should also be studied in terms of using a Jericho as the host machine for both the SQS and the TDP/TDI system.

ACKNOWLEDGEMENTS

We would like to offer our thanks to to a number of consultants whose contributions have greatly aided this contract. The most important contributions were made by the late Earl Bowersox, who provided the initial impetus to examine the specific area of tactical deception, as well as lending much energy and knowledge to the project as a whole. We will miss him, both as a fine individual and as an inspirational consultant.

Bill McBride provided much help with the notions of deception and with many of the specifics of the scenario.

Frank Zdybel was responsible for many of the ideas discussed in the sections on knowledge-based simulation, and provided early drafts of those sections.

REFERENCES

[1] R. J. Bechtel and P. H. Morris.
STAMMER: System for Tactical Assessment of Multisource Messages, Even Radar.
Technical Document 252, Naval Ocean Systems Center, May, 1979.

[2] G. Birtwistle, O.J. Dahl, B. Myhrhaug, and K. Nygaard.
SIMULA BEGIN.
Auerbach, Philadelphia, PA, 1973.

[3] R. Bisbey II and D. Hollingworth.
A Distributable, Display-Device-Indepent Vector Graphics System for Command and Control.
Technical Report ISI/RR-80-87, Information Sciences Institute, July, 1980.

[4] R. Bisbey II, D. Hollingworth and B. Britt.
Graphics Language.
Technical Memo ISI/TM-80-18, Information Sciences Institute, July, 1980.

[5] R. Bobrow, J. Vittal and O. Selfridge.
Specification of a System for Tactical Deception Indication.
Technical Report 4599, Bolt Beranek and Newman Inc., February, 1981.

[6] A. H. Borning.
THINGLAB-- A Constraint-Oriented Simulation Laboratory.
PhD thesis, Stanford University, March, 1979.

[7] R. J. Brachman.
A Structural Paradigm for Representing Knowledge.
PhD thesis, Harvard University, May, 1977.
Also, BBN Report No. 3605, Bolt Beranek and Newman Inc., July, 1978.

[8] R. J. Brachman, E. Ciccarelli, N. R. Greenfeld, and M. D. Yonke.
KLONE Reference Manual.
Technical Report 3848, Bolt Beranek and Newman Inc., July, 1978.

[9] R. J. Brachman.
Theoretical Studies in Natural Language Understanding.
Technical Report 3888, Bolt Beranek and Newman Inc., September, 1978.

[10] R. Brandenberg, M. Gallegos, R. Hayes-Roth, and D. Waterman.
RITA Threat Evaluation and Countermeasures Agent.
Working paper, The Rand Corporation, 1977.

[11] J. S. Brown and R. R. Burton.
Multiple Representations of Knowledge for Tutorial Reasoning.
In D. G. Bobrow and A. Collins (editor), Representation and Understanding, pages 311-349. Academic Press, New York, 1975.

[12] K. Forbus.
A CONLAN Primer.
Technical Report 4491, Bolt Beranek and Newman Inc., March, 1981.

[13] A. Goldberg and A. Kay.
SMALLTALK-72 Instruction Manual.
Technical Report SSL 76-6, Xerox Palo Alto Research Center, March, 1976.

[14] S. Goldin and P. Klahr.
Learning and Abstraction in Simulation.
In Proceedings of the Seventh International Joint Conference on Artificial Intelligence, pages 212-214. IJCAI, University of British Columbia, Vancouver, BC, August, 1981.

[15] N. Greenfeld, F. Zdybel, E. Ciccarelli, and M. Yonke.
BMG Reference Manual.
Technical Report 4368, Bolt Beranek and Newman Inc., April, 1980.

[16] N. Greenfeld.
Jericho: A Professional's Personal Computer System.
In 8th International Symposium on Computer Architecture. 8th International Symposium on Computer Architecture, Rochester, MN, May, 1981.

[17] C. F. Herot.
Spatial Management of Data.
Technical Report CCA-79-24, Computer Corporation of America, June 30, 1979.

[18] C. Herot, R. Carling, M. Friedell, D. Kramlich and J. Thompson.
Spatial Data Management System: Semi-Annual Technical Report.
Technical Report CCA-79-25, Computer Corporation of America, June 30, 1979.

[19] C. Herot and G. Wilson.
Semantics Versus Graphics - To Show or Not To Show.
Technical Report CCA-80-09, Computer Corporation of America, March 30, 1980.

[20] D. Ingalls.
The SMALLTALK-76 Programming System Design and Implementation.
In Proceedings of the Fifth Annual ACM Symposium on Principles of Programming Languages. ACM, Tuscon, AZ, January 23-25, 1978.

[21] W. Teitelman.
InterLisp Reference Manual
XEROX Palo Alto Research Center, Palo Alto, CA, October 1978.

[22] P. Klahr and W. Faught.
Knowledge-based Simulation.
In Proceedings, First Annual National Conference on Artificial Intelligence. Stanford University, Stanford, CA, 1980.

[23] The Learning Research Group.
Personal Dynamic Media.
Technical Report SSL 76-1, Xerox Palo Alto Research Center, 1976.

[24] Z. Manna and R. Waldinger.
The Automatic Synthesis of Systems of Recursive Programs.
In International Joint Conference on Artificial Intelligence - 1977, pages 405-411. IJCAI, Cambridge, MA, August, 1977.

[25] D. McArthur and H. Sowizral.
An Object-Oriented Language for Constructing Simulations.
In _Proceedings of the Seventh International Joint Conference on Artificial Intelligence_, pages 809-814. IJCAI, University of British Columbia, Vancouver, BC, August, 1981.

[26] J. McCarthy.
Recursive Functions of Symbolic Expressions and their Computation by Machine.
Communications of the ACM, April, 1960.

[27] J. McCarthy.
History of LISP.
In _Proceedings of the ACM Conference on the History of Programming Languages_. ACM, 1977.

[28] D. A. Moon.
MACLISP Reference Manual
MIT, 1974.

[29] J. S. Moore II.
The Interlisp Virtual Machine Specification.
Technical Report CSL-76-5, Xerox Palo Alto Research Center, September, 1976.

[30] E. Sacerdoti.
The Non-Linear Nature of Plans.
In _Advance Papers of the Fourth IJCAI_, pages 206-214. IJCAI, Tbilisi, Georgia, USSR, September, 1975.

[31] J. Schmolze.
Guidelines for Choosing an Object-Oriented Programming Style in Lisp.
January, 1981.
In preparation. Submitted to the 1981 Computer Faire.

[32] _Byte_ 6(8), August, 1981.
This is a special issue dedicated to Smalltalk.

[33] G. L. Steele, Jr. and G. J. Sussman.
Constraints.
Technical Report 502, MIT-AI, November, 1978.

[34] M. Stonebraker, E. Wong, P. Kreps and G. Held.
The Design and Implementation of INGRES.
ACM Transactions on Database Systems 1(3):189-222, September, 1976.

[35] M. Stonebraker and E. Neuhold.
A Distributed Database Version of INGRES.
Memorandum No. ERL-M612, Electronic Research Laboratory, University of California, September, 1976.

[36] W. Teitelman and L. M. Masinter.
The Interlisp Programming Environment.
Computer 14(4):25-33, April, 1981.

[37] R. Waldinger.
Achieving Several Goals Simultaneously.
In E. W. Elcock and D. Michie (editors), Machine Intelligence 8: Machine Representation of Knowledge, . John Wiley & Sons, New York, NY, 1977.

[38] D. A. Waterman and F. Hayes-Roth.
Pattern-Directed Inference Systems.
Academic Press, New York, NY, 1978.

[39] Daniel Weinreb and David Moon.
Lisp Machine Manual
Second Preliminary Version edition, MIT AI Lab, 1979.

[40] G. Wilson.
Three-Dimensional Knowledge Presentation.
Technical Report CCA-80-11, Computer Corporation of America, December, 1980.

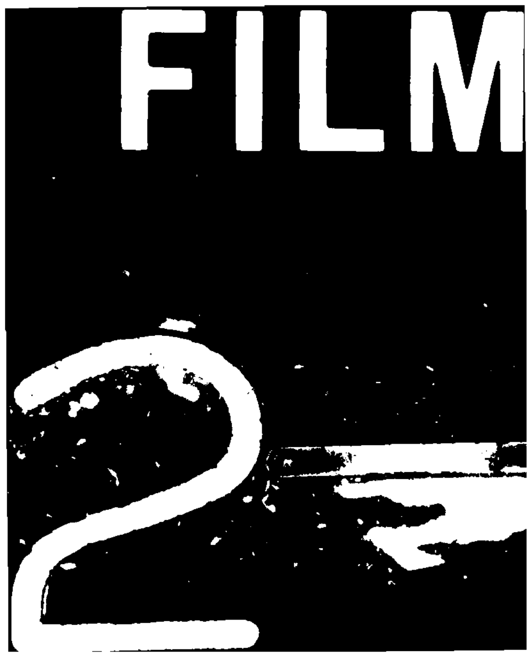
[41] G. Wilson.
Multi-media Cooperative Person/Machine Interaction.
March, 1981.

[42] W. Woods.
Theoretical Studies in Natural Language Understanding.
Technical Report 4332, Bolt Beranek and Newman Inc., April, 1979.
Annual Report.

[43] M. Yonke and N. Greenfeld.
AIPS: An Information Presentation System for Decision Makers.
In Proceedings of the Thirteenth Hawaii International Conference on System Sciences. the University of Hawaii and the Association for Computing Machinery, January, 1980.

[44] F. Zdybel, M. D. Yonke and N. R. Greenfeld.
Application of Symbolic Processsing to Command and Control.
Technical Report 3894, Bolt Beranek and Newman Inc.,
February, 1980.

[45] F. Zdybel, N. Greenfeld, M. Yonke and J. Gibbons.
An Information Presentation System.
In Proceedings of the Seventh International Joint Conference on Artificial Intelligence, pages 978-984.
IJCAI, University of British Columbia, Vancouver, BC,
August, 1981.


FILM
2