DRAFT
Final Report for the Intelligent Multi-Media Interfaces Project

Contract Period: October 27, 1987 to October 26, 1989

J.G. Neal, S.C. Shapiro, C.Y. Thielman, S.A. Glanowski,
J.M. Lammens, D.J. Funke, J.S. Byoun, Z. Dobes,
M.S. Summers, J.R. Gucwa, R. Paul

November 30, 1989

Contents

1 INTRODUCTION

1.1 MOTIVATION

The introduction of improved and advanced processing capabilities into Air Force Command and Control (C2) systems is proceeding at an ever-increasing rate. This has placed great pressure on the human-computer interface of these systems. Large amounts of information must be communicated between the human users and these computer-based systems. Further, this human-computer communication must be accomplished quickly and without error to support time-critical decision-making tasks within the command and control environment.

It is essential that the human-machine interfaces to these information intensive systems not become limiting factors which degrade the larger command and control functions. Too often in the past, the human-machine interface was either overlooked or handled much like a retrofit after the fact. In today's information explosive environment it is critical that human-computer interface technology be developed and applied to meet the demands of modern sophisticated computer-based systems. Martin [Martin73] expressed it well:

> "For man, this is a hostile environment. His mind could no more cope with this deluge of data, than his body could cope with outer space. He needs protection. The computer – in part the cause of the problem – is also the solution to the problem. The computer will insulate man from the raging torrents of information that are descending upon him."

The research conducted during this effort was motivated by the need for more capable and powerful human-computer interface technology. It has attempted to apply artificial intelligence, interactive graphics, speech recognition, and speech generation to build powerful and efficient human-computer interfaces more capable of meeting the demands of modern information intensive systems. The goal has been to integrate and combine various graphic and voice human-computer interface technologies in a manner that enhances human-computer communication. By dynamically selecting output media based on the features and capabilities of that media vis-a-vis human sensing and understanding mechanisms, while also considering the context of the communication and combining multiple output media to achieve increased bandwidth and linguistic redundancy, it was expected that the human-computer interface efficiency could be enhanced. Additionally, by accepting user inputs via combinations of multiple media selected by the user, the resulting human-computer interface was expected to be both natural to use and highly effective.

The development of new user interface technology is driven by both interface requirements and by computer hardware and software capabilities. Not only does computer technology drive human-computer interface requirements as noted above, but it also determines the approaches available to the human-computer interface. Before parallel processing and high resolution integrated displays technology, human-computer interfaces employing integrated windowing environments were not possible. Today these are commonplace. A plethora of human factors research projects have explored, or been undertaken to explore, how to best apply these technologies. This project has applied the emerging technologies of artificial intelligence and voice recognition and synthesis to the human-computer interface. This effort was a first step toward the goal of adding these technologies to the growing war-chest of human-computer interface solutions.

Human-computer interface technology is developed in an empirical fashion. It is necessary to build upon our existing understanding of interface design techniques and our knowledge of past, current, and emerging human-computer interface requirements, to design, evaluate, and build better human-computer interfaces. As new technology provides opportunity for better human-computer interfaces, testbeds for applying these technologies and exploring alternative implementation approaches are needed. CUBRICON was developed as one such testbed, and this project has started the process of determining how best to apply these technologies to achieve the goals of an enhanced human-computer interface.

1.2 FUNCTIONALITY OVERVIEW

The CUBRICON system design is based upon a *unified view of language.* Language is a means of communication, whether verbal, visual, tactile, or gestural. Human beings communicate with each other via written and spoken natural language, frequently supplemented by pictures, diagrams, pointing to objects, and other gestures. It is a unified language, in that these various modalities are integrated and combined to represent and describe a single underlying reality.

The CUBRICON system design provides for the use of a unified multi-media language, defined by an integrated grammar, consisting of textual, graphic, and combined text/graphic symbols. Inputs to, and outputs from CUBRICON, are treated as compound information streams with components corresponding to different media. This approach is intended to take advantage of humans' ability to simultaneously accept information from different sensory channels (such as eyes and ears), and to simultaneously generate communications in different media (such as voice, pointing motions, and drawings).

The CUBRICON system includes: (a) language parsing and generation capabilities to support the understanding and creation of multi-media information streams, (b) knowledge representation and inferencing capabilities to provide for reasoning about the meanings of all communications vis-a-vis the underlying application, (c) knowledge bases and models to provide a basis for decision-making with regard to taking action in response to communications, and (d) automated knowledge-based reasoning models to provide for media selection and the formulation of responses that takes advantage of human sensing and understanding capabilities.

Functionally, CUBRICON is distinct from other human-computer interface systems because it provides intelligent integration of multi-media input and output. This allows CUBRICON to have a unified view of multi-media language, thus providing a powerful potential for accomplishing human-computer interactions. The following unique CUBRICON features are important parts of this capability:

- CUBRICON integrates multiple input and output modalities. Input modalities include voice, pointing via mouse, form-based input, and typed text. Output modalities include voice, pointing/highlighting, forms, tables, and typed text. CUBRICON's unified view of language will allow efficient addition of other input and output modalities if needed.
- CUBRICON accepts inputs from human users in way that is natural and desirable to the user. Specifically CUBRICON can:
  - Coordinate input from different devices and modalities.
  - Accept varying numbers of point gestures within phrases, and allow a variety of object types to be the targets of point gestures.
  - Accept varying numbers of multi-modal phrases within sentences.
  - Use natural language inputs to disambiguate corresponding point gestures (and eventually point gestures will also be able to disambiguate natural language).
  - Handle certain types of ill-formed multi-modal inputs.
- CUBRICON provides for intelligent and automatic management of windows. This includes:
  - A method for determining window importance (used for deciding which windows to remove when display space is needed for other windows).
  - A procedure for automatically managing windows in a dual monitor environment. This procedure considers window importance and type.
- CUBRICON generates multi-media outputs in a manner that enhances understandability. Specific features are:

  - Modality selection is based on the characteristics of the information to be expressed vis-a-vis human sensing and understanding capabilities, as well as task and dialog context.
  - Multiple modalities are combined to: 1) take best advantage of the relative strengths of each; 2) add emphasis or orientation to accompanying modalities; and 3) provide redundancy to ensure understanding and notice of important information.
  - All multi-media outputs are temporally synchronized (e.g., highlighting of graphics is temporally coordinated with related speech).
  - Spoken and written natural language outputs are designed for short-term and long-term reference, respectively. For example, written outputs include specific object referencing rather than using pronouns. This allows correct interpretation of the output over a longer period of time.
  - System outputs maintain format consistency within and across displays, and also provide for contextual orientation across all displays throughout the user-computer dialogue.

- CUBRICON is a knowledge-based system. Input understanding and output composition considers dialog context (i.e., what is currently being displayed and has recently been expressed), task context (i.e., the importance of information relative to the ongoing task), and information context (i.e., the nature of the information vis-a-vis human sensing and understanding capabilities).

1.3 IMPLEMENTATION AND STATUS

The CUBRICON system is implemented on a Symbolics Lisp Machine with a mouse pointing device, a color-graphics monitor, and a monochrome monitor. Speech recognition is handled by a Dragon Systems VoiceScribe 1000. Speech output is produced by a DECtalk speech production system. CUBRICON software is implemented using the SNePS semantic network processing system [Shapiro79a; Shapiro81; Shapiro86], an ATN parser/generator [Shapiro82a], and Common Lisp. SNePS is a fully intentional propositional semantic network and has been used for a variety of purposes and applications [Maida85; Shapiro86; Neal86, Neal87]. SNePS provides: (a) a flexible knowledge representation facility in the semantic network formalism; (b) representation of rules in the network in a declarative form so they can be reasoned about like any other data; (c) a bi-directional inference subsystem [Shapiro82b] which focuses attention towards the active processes and cuts down the fan-out of pure forward or backward chaining; (d) a simulated multi-processing control structure

[McKay80]; (e) special non-standard connectives [Shapiro79b] to model human reasoning processes; and (f) existential, universal, and numerical quantifiers [Shapiro79c].

CUBRICON is a proof-of-concept system. It integrates multiple-media input and output, and provides knowledge-based understanding and generation of human-computer communication including natural language, pointing/highlighting, and form-based interface technologies. As a proof-of-concept system, it performs the functions of multi-media human-computer communication (see Section 1.2), but not at the level of sophistication that would be expected of a final system. For example, speech input uses a discrete voice recognition system rather than a more expensive continuous voice recognition system. The CUBRICON grammar and lexicon have been developed to support the present proof-of-concept application. It does not provide for the processing of all possible English language structures and terms. Finally, CUBRICON has been developed in the lab. It does not use a hardware and software design that would permit extremely fast and efficient processing of inputs and outputs (i.e., CUBRICON dialogue is slower than human to human dialogue). Rather, it was designed for efficient development and evaluation of the technology itself. Improvements in speed and naturalness could be made in future CUBRICON implementations.

Finally, The focus of the CUBRICON implementation has been to develop a proof-of-concept intelligent integration of multiple-media input and output modalities, which can be implemented efficiently as a front-end to a variety of application systems. State-of-the-art technologies have been applied, and in some cases developed, to achieve this end. The emphasis has been on the application of artificial intelligence based technologies to the human-computer interface problem. Only peripheral efforts have been expended in the application of more routine or standard human-computer interface techniques. For example, significant effort has been applied to achieve understanding of simultaneous voice and pointing inputs, while little effort has been made to provide menu-based alternatives. While a final application of CUBRICON will offer both voice/pointing and menu-based approaches to human-computer communication, it currently offers very little in the way of menus. This can be added later.

1.4 ORGANIZATION OF THE REPORT

This report describes the research efforts conducted during this project and presents the results of the evaluation which attempted to measure how well the above goals (see Section 1.1) were achieved. A brief overview of the organization of this report is given below:

Report Section: Summary of Contents:

1	Introduction. Provides an introduction to and functional overview of the Intelligent Integrated Interface Project, and of the CUBRICON system.
2	Overview of System Design. Contains an overview description of the CUBRICON design including a brief description of each major system component.
3-6	These sections describe the design of all major CUBRICON processing and knowledge base components.
7-13	These sections describe major CUBRICON output technologies and modalities.
14	KB Builder Tool. Describes the CUBRICON Knowledge Base Builder Tool which is used for implementing CUBRICON as the human-computer interface for a new application system.
15	Evaluation. Describes the CUBRICON evaluation that was conducted during this effort and summarizes the results (evaluation data is contained in Appendix C).
16	Future Directions. Recommends future directions for the CUBRICON system and related research.
17	Summary. Provides a summary of CUBRICON and the research that was accomplished during this effort.
18	References. Contains a complete list of all references made within this document and its appendices (excepting Appendix H).

Appendixes

A	Example CUBRICON Dialogue. Contains examples of user-CUBRICON dialogue which illustrate important CUBRICON features. These sample dialogues are illustrated with pictures of actual CUBRICON displays.

B Grammar and Lexicon. Contains a complete description of the grammar and lexicon used by CUBRICON.

C Graphic Representation of the Task Domain Knowledge Base. Contains diagrams which depict the structure of the Task Domain Knowledge Base.

D Software Documentation for Primary Functions. Contains descriptions of the software used to accomplish primary functions within CUBRICON.

E Evaluation Training Material and Data. Contains: a complete set of material used to train subjects for the CUBRICON evaluation; all work aids used during the evaluation; and data generated during the evaluation.

F Working Paper on Computer Speech Generation. Contains a working which presents the results of a literature review on human factors issues relating to the use of computer generated speech. This paper was delivered to DARPA and RADC earlier in the program and is included here for completeness.

G Working Paper Describing Locative Referencing for Map-Based Systems. Contains a working paper describing research conducted in in association with CUBRICON. This research resulted in the implementation of a definitive referencing capability within CUBRICON, but was funded by another agency.

H References to Published Technical Papers Describing CUBRICON and the Research Conducted Under the Intelligent Integrated Interfaces Program.

2 OVERVIEW OF SYSTEM DESIGN

The CUBRICON team has designed and implemented an integrated user interface system with the functionality described briefly in Section 1.2. Figure 1 provides an overview of the software system and hardware I/O devices currently supported by CUBRICON.

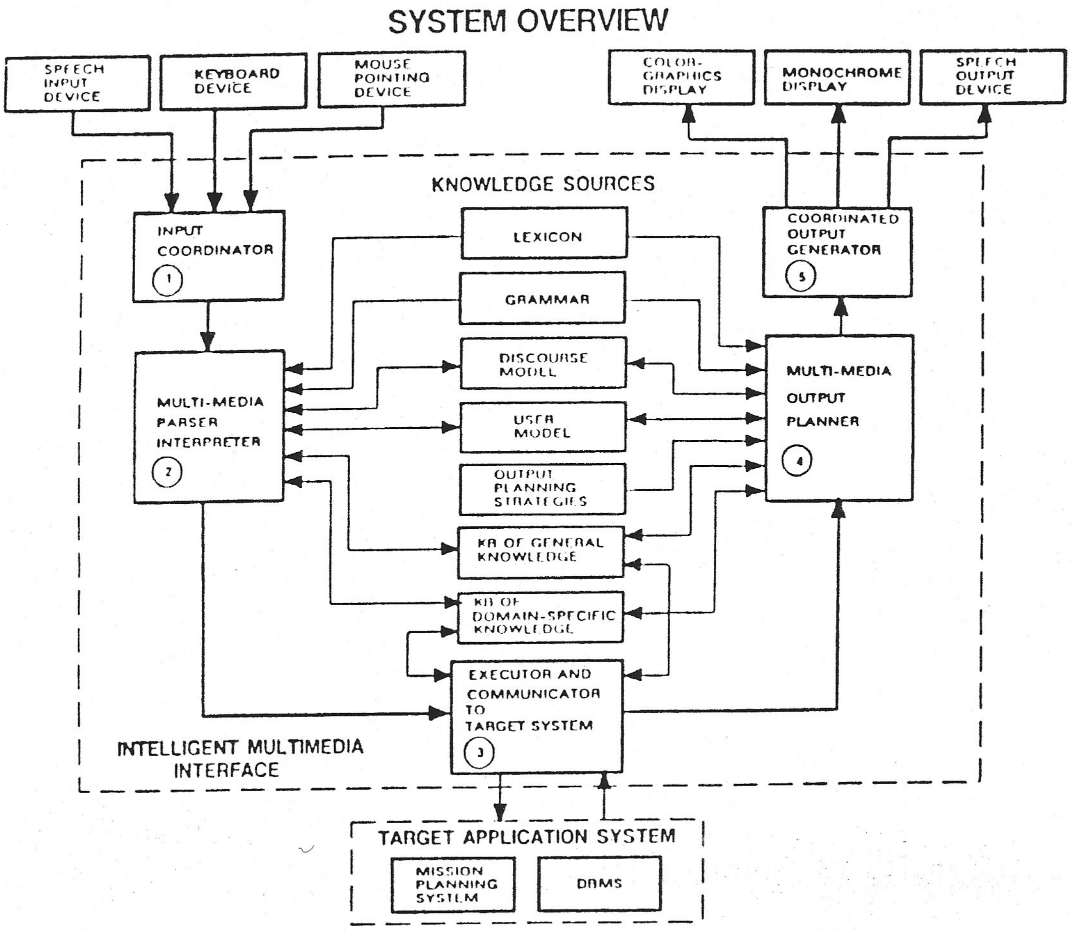


Figure 1: System Overview

CUBRICON accepts input from three input devices: a speech recognition system, a keyboard, and a mouse. CUBRICON produces output via three output devices: a high-resolution color-graphics display, a monochrome display, and a speech output device.

The primary data processing flow through CUBRICON is indicated by the numbered modules in Figure 1. These are: (1) Input Coordinator, (2) Multi-media Parser Interpreter, (3) Executor/Communicator to Target System, (4) Multi-media Output Planner, and (5) the Coordinated Output Generator. Each of these are briefly described in the following paragraphs.

Inputs to CUBRICON are handled by the Input Coordinator and the Multi-Media Parser Interpreter. The Input Coordinator module accepts input from the three input devices and fuses the input streams into a single compound stream, maintaining the temporal order of tokens in the original input stream. The Multi-media Parser/Interpreter is an augmented transition network (ATN) that has been extended to: 1) accept the compound stream produced by the Input Coordinator and 2) produce an interpretation of this compound stream.

Once inputs are received and understood by CUBRICON appropriate action is then taken by the Executor module. This action may be a command or database query to the underlying application (e.g., a mission planning system, a database), or an action that entails participation of the interface system only.

An expression of the results of CUBRICON action (completed by the Executer) are planned by the Multi-Media Output Planner for communication to the user. The Multi-Media Output Planner is a generalized ATN that produces a multi-media output stream representation, with components targeted for different devices (e.g., color-graphics display, speech output device, monochrome display). This output stream representation is translated into visual/auditory output by the Coordinated Output Generator module. This module is responsible for producing the multi-media output in a coordinated manner in real time. For example, the Multi-Media Output Planner module may specify that a certain icon on the color-graphics display must be highlighted when the entity represented by the icon is mentioned in the simultaneous natural language voice output. The Coordinated Output Generator implements this coordinated output.

The CUBRICON system incorporates several knowledge sources that are used during processing. The knowledge sources currently include: (1) a lexicon, (2) a grammar defining the language used by the system for multi-media input and output, (3) a discourse model, (4) a user model, (5) a knowledge base of output planning strategies to govern the composition of multi-media responses to the user, (6) a knowledge base of information about generally shared world knowledge, and (7) a knowledge base of information about the specific task domain of tactical air control. These knowledge sources are used for both understanding input to the system and planning/generating output from the system. They are discussed in more detail in the next section.

In its entirety, the CUBRICON system provides an integrated multi-media human-computer interface system which can be implemented as a front-end to a target application system. Inputs are accepted via a combination of input modalities. Outputs are accepted via a combination of output modalities. CUBRICON is designed in a way that allows it to be applied to a variety of application systems with only minimal programming efforts. It also is configured to accept the incorporation of additional input and output modalities to support future interface needs.

3 KNOWLEDGE SOURCES

The CUBRICON system includes several knowledge sources for use in multi-media language understanding and production. These knowledge sources are: a lexicon; grammar; discourse model; user model; a knowledge base of output planning strategies to govern the composition of multi-media responses to the user; and a knowledge base of information about the task domain of tactical air control.

This section contains: a description of the knowledge base relating to the task domain of tactical air control and related display information; the discourse model; and the user model. Descriptions of the other knowledge bases are contained within other sections of this report. Specifically, the lexicon and grammar are described in Sections 4 and 12, and in Appendix A. The knowledge base of output planning strategies is discussed in Section 6.

3.1 TASK DOMAIN KNOWLEDGE BASE

The Task Domain Knowledge Base contains domain-specific information relating to the particular task domain of the application system to which CUBRICON is serving as the human interface. It includes knowledge of the entities and concepts known to the underlying system, as well as information concerning the presentation or expression of those entities or concepts. For example, this knowledge base typically will include information about how an entity should be expressed via a unified verbal/graphic output. This includes words and symbols that can be used to express the entity or concept, along with conditions that determine the definition of specific expressions containing them (e.g., when and how to abbreviate, color codes to apply, symbology definition). This knowledge base is defined within a semantic network which encodes its structure and the meanings of the objects it contains.

The current CUBRICON implementation contains a knowledge base relating to the general task domain of tactical air control. It relates specifically to an early version of the AMPS data base developed by The MITRE Corporation for RADC. The AMPS data base was designed to support the planning Air Force Air Tasking Orders. A graphical representation of this knowledge base is contained in Appendix C.

(more detailed description forthcoming)

3.2 DISCOURSE MODEL

Continuity and relevance are key factors in discourse. Without these factors, people find discourse disconcerting and unnatural. The attentional discourse focus space representation [Grosz78, Grosz86; Sidner83; Grosz85] is a key knowledge structure that supports continuity and relevance in dialogue. It is used for determining the interpretation of anaphoric references [Sidner83] and definitive references [Grosz81] expressed by the user in natural language. CUBRICON tracks the attentional discourse focus space of the dialogue carried out in multi-media language. This is accomplished with four structures: (1) the Main Focus List; (2) the Display Model; (3) the Presentation Object Data Structure; and (4) the Form Model. Each of these Discourse Model components are discussed in this section.

3.2.1 Main Focus List

Could describe the list more fully

The Main Focus List is CUBRICON's primary means of tracking the attentional discourse focus space. It consists of a continually updated list of those entities and propositions that have been explicitly expressed (by the user or by CUBRICON) via natural language, pointing, highlighting, or blinking. The Main Focus List maintains a temporal record of when references were made, and is used by CUBRICON in determining pronoun or definite referents, and objects or locations to be used within locative references.

3.2.2 Display Model

The display model represents all the objects that are "in focus" because they are visible on one of the monitors. Graphics are an integral part of CUBRICON's language along with natural language and other forms of text and pointing. The CUBRICON system treats objects presented on the graphics displays as having been intentionally "expressed" or "mentioned". All objects on the graphics display are therefore "in focus" and CUBRICON maintains a representation of all these objects in the form of a display model. The display model is defined at two levels: (1) a list of all displayed windows on each monitor and, (2) for each window, a list of all the objects that are visible within it.

The Display Model is used by CUBRICON in the determination of how to express new outputs to the user. All display updates are generated based on the pre-existing display context, represented by the Display Model. Display updates are designed to build upon the pre-existing display context in a way that minimizes display (and dialogue) disruption and maximizes display (and dialogue) continuity. For example, expressions involving entities already displayed are accomplished through diectic dual-media expressions rather than the generation of new displays (i.e., windows). The diectic dual-media expression consists of a phrase such as "this airbase" with simultaneous blinking/highlighting of the airbase icon as its means of pointing to it. If the entity is the most salient of its gender according to the main focus list, CUBRICON may use a pronoun as the verbal part of the expression. The Display Model plays a central role in this process, since it is the source of minute-to-minute knowledge of what is currently being displayed.

The Display Model is also used in a manner that is analogous to the use of the Main Focus List. That is, it supports the identification of pronoun, definite, or locative references. In the case of locative references, the role of the Display Model is somewhat obvious. Entities to be used for a locative reference must already be on the display. In the case of determining pronoun or definite references, the Display Model is consulted when the referent is not found in the Main Focus List. For example, when a person expresses a definite reference such as "the airbase" with just one such object in view (as on a graphics display), and when none have been previously discussed, CUBRICON assumes that this airbase is the one that was meant, even though several others may be contained in the knowledge base. If many airbases are currently displayed in this situation, CUBRICON might select the airbase most relevant to the user's task (e.g., only friendly airbases would be selected as an origin for a strike mission), or if no disambiguating information at all were available, it might respond with the question, which airbase do you mean? In any event, the Display Model is consulted to ascertain what is currently being displayed, and this information is used in determining appropriate outputs.

3.2.3 Presentation Object Data Structure

CUBRICON records the current presentation objects in a tree structure referred to as the presentation object data structure or PODS. A presentation object represents an output mode of expression, such as a highlighted icon or a dynamic window. The PODS serves two functions. First, it is used to determine which presentation objects to regenerate when a map is redisplayed by zooming in or zooming out. Secondly, it is used to determine when and how to remove a presentation objects from a display.

The PODS organizes the presentation objects by the function they serve within the CUBRICON system. This is necessary since the same presentation object is treated differently within CUBRICON depending on why it was created.

3.2.3.1 Presentation Objects

CUBRICON does not record every output mode of expression in the PODS. Presentation objects which are dynamic in nature, appearing and disappearing based on recency of creation, are recorded. Table entries, for example, are static in nature. They are never removed once they are included in a table. Therefore, these presentation objects are not included in the PODS. The following is a list of the presentation objects included in the PODS: icons which trace the location of mouse points, string labels, highlighted windows, highlighted table entries, highlighted icons, pointing text windows, dynamic windows, flight paths, icons created during a flight path presentation, and context boxes which show the relationship between the previous and currently displayed map area.

3.2.3.2 Functionality Types

The PODS is organized primarily by the function of the presentation object within the system. Presentation objects serving different functions within the system are handled differently by CUBRICON. For example, the same type of label, which is placed on a map, is used to identify the order in which point gestures occurred as well as to identify the property and value of an entity. These presentations are treated differently, labels associated with a pointing gestures are removed prior to output generation, whereas labels identifying the property and value of an entity are removed after fifteen requests.

The functions a presentation object serves can be one of the following:

- Mouse Gesture Pointing

  The presentation objects which trace input pointing gestures via a mouse. This includes a map icon indicating the location of the mouse point, a pointing arrow indicating the icon referred to if the mouse point was ambiguous, and a label indicating the order in which the mouse points occurred.

- Map Icon Pointing

  The diectic gestures which point to a map. This includes icon highlighting, table entry highlighting, mission planning form highlighting, and a pointing text box which points to the map icon.

- Window Pointing

  The diectic gesture which point to a window. This includes highlighting the window frame of the window being pointed to.

This is the kind of detail missing in focus list section

A

- Map Context Box

  The presentation object which shows the context between the previous and current map. This includes an orange rectangle outlining a region on a map.

- Property Labels

  The label which display a property and value of an entity, currently a map icon.

- Mission Presentation

  The presentation objects which were created during a mission presentation. This includes map icons, highlighted map icons, highlighted table entries, highlighted form entries, dynamic windows, and flight paths.

3.2.3.3 Data Structure Format

The PODS is a tree structure including the following information:

- Functionality

  The functionality of the presentation object as described in Section 3.2.3.2.

- Recency

  Time the presentation object was created. This item is used along with the functionality of the presentation object to determine when to remove the presentation.

- Key

  Key used to differentiate presentation objects with the same functionality. For example, presentation objects created during different mission presentations.

- Window Identifier

  The identifier of the window instance containing the presentation object.

- PO Type

  The type of presentation object, used to determine the function call and arguments needed to remove or generate the presentation object. Section3.2.3.1 identifies these presentation objects.

- Node List

  List of SNePS node(s) for which the presentation was performed. This field is needed to remove and generate presentation objects which refer to a map icon, map icon highlighting for example.

```
((<functionality> <recency> <key>
  ((<window identifier>
    ((<PO type> <node list> <optional argument>)* )
   )* )
 )* )
```

Figure 2: PODS Structure Diagram

- Optional Arguments

  Any additional arguments needed to generate the presentation object. For example, the contents of a string label is stored as an optional argument.

The structure of the PODS is shown in Figure 2. The PODS includes three association lists with keys of functionality, window identifier, and PO type respectively. These lists will be referred to as the functionality list, the window identifier list and the PO type list. A sample PODS is shown in Figure 3. The first list, which contains the functionality keyword :PROPERTY-LABEL, was added to the PODS as a result of a request for the mobility of three sam systems. One of the presentation objects generated is a label associated with an icon represented by the SNePS node identifier B25. This label contains the string "Mobility High". Two additional labels were generated during this request. These labels are associated with the icons represented by the SNePS node identifiers B15 and B18, containing the string "Mobility Low".

The second list in the PODS, which contains the functionality keyword :MISSION, was added to the PODS as a result of a request to generate the OCA345 mission plan. There are numerous presentation objects generated when presenting a mission plan. First, the origin airbase is pointed to by highlighting and labeling, resulting in the addition of the first two PODS lists with keywords :HIGHLIGHTED-ICON and :STRING-LABEL. The origin airbase is represented in the knowledge base by the the SNePS node identifier B40. The flight path generated during a mission plan consists of waypoints connected by arrows, indicating the direction of the aircraft, and labels, indicating the time of arrival at each waypoint. The presentation objects comprising a flight path are represented in the lists containing the keywords :FLIGHT-PATH and :STRING-LABEL. The explosion of the target airbase added the list containing the keyword :HIGHLIGHTED-ICON to the PODS. An orbit occurring during the mission presentation added an icon to the map window and consequently added a list containing the keyword :ICON to the PODS. In addition, a flight path presentation generates a dynamic window containing text describing the mission plan (:DYNAMIC-WINDOW), the

```
((:PROPERTY-LABEL 3 NIL
  ((#<GUIDE-WINDOW Guide Window 5 11010157 exposed>
    ((:STRING-LABEL (B25) ("mobility: high"))
     (:STRING-LABEL (B15 B18) ("mobility: low"))))
   (#<GUIDE-WINDOW Guide Window 6 11010665 exposed>
    ((:STRING-LABEL (B25) ("mobility: high"))
     (:STRING-LABEL (B15 B18) ("mobility: low"))))))
 (:MISSION 5 "OCA345"
  ((#<GUIDE-WINDOW Guide Window 5 11010157 exposed>
    ((:HIGHLIGHTED-ICON (B40) :CIRCLE)
     (:STRING-LABEL (B40 B171) ("Origin Air Base."))
     (:FLIGHT-PATH (B40 B172) NIL)
     (:STRING-LABEL (B172) ("5:55")) (:FLIGHT-PATH (B172 B173) NIL)
     (:STRING-LABEL (B173) ("6:15")) (:FLIGHT-PATH (B173 B174) NIL)
     (:STRING-LABEL (B174) ("6:30")) (:FLIGHT-PATH (B174 B175) NIL)
     (:STRING-LABEL (B175) ("6:45")) (:FLIGHT-PATH (B175 B176) NIL)
     (:STRING-LABEL (B176) ("6:50")) (:HIGHLIGHTED-ICON (B50) :EXPLODE)
     (:FLIGHT-PATH (B176 B177) NIL)  (:STRING-LABEL (B177)("7:00"))
     (:FLIGHT-PATH (B177 B178) NIL)  (:STRING-LABEL (B178) ("7:05"))
     (:FLIGHT-PATH (B178 B179) NIL)  (:STRING-LABEL (B179) ("7:15"))
     (:FLIGHT-PATH (B179 B180) NIL)  (:STRING-LABEL (B180) ("7:25"))
     (:ICON (B227) NIL)              (:FLIGHT-PATH (B180 B181) NIL)
     (:STRING-LABEL (B181) ("7:40")) (:FLIGHT-PATH (B181 B182) NIL)
     (:STRING-LABEL (B182) ("8:00")) (:FLIGHT-PATH (B182 B183) NIL)
     (:STRING-LABEL (B183) ("8:10")) (:FLIGHT-PATH (B183 B40) NIL)
     (:STRING-LABEL (B40 B184) ("Mission completed."))))
   (#<MISSION-WINDOW Form Window 11000744 deexposed>
    ((:HIGHLIGHTED-FORM (M1571!) NIL) (:HIGHLIGHTED-FORM (M1542!) NIL)
     (:HIGHLIGHTED-FORM (M1573!) NIL) (:HIGHLIGHTED-FORM (M1721!) NIL)
     (:HIGHLIGHTED-FORM (M1660! M1661!) NIL) (:HIGHLIGHTED-FORM (M1723!) NIL)
     (:HIGHLIGHTED-FORM (M1660! M1661!) NIL)
     (:HIGHLIGHTED-FORM (M1660! M1661!) NIL)))
   (#<TEXT TEXT WINDOW 11011373 deactivated> ((:DYNAMIC-WINDOW B221 NIL)))
   (#<TEXT-PRESENTATION-WINDOW Text Presentation Window 8 11010431 deactivated>
    ((:HIGHLIGHTED-TABLE-ENTRY (B40) NIL)))))
 (:POINT-AT 6 NIL
  ((#<TEXT-PRESENTATION-WINDOW Text Presentation Window 8 11010431 deactivated>
    ((:HIGHLIGHTED-TABLE-ENTRY (B47) NIL)))
   (#<GUIDE-WINDOW Guide Window 5 11010157 exposed>
    ((:TEXT-WINDOW (B47) ("dresden air7base"))
     (:HIGHLIGHTED-ICON (B47) :CIRCLE))))))
```

Figure 3: Sample PODS

highlighting of relevant information on a mission planning form (:HIGHLIGHTED-FORM), the highlighting of relevant information on tables (:HIGHLIGHTED-TABLE-ENTRY), and a label indicating the mission presentation is complete (:STRING-LABEL).

The last list in the PODS was added as a result of the request for the location of the Dresden airbase. The result was the the highlighting of the table entry identifying the properties of the Dresden airbase (:HIGHLIGHTED-TABLE-ENTRY), a text-box containing the string "dresden air base" pointing to the icon representing the Dresden airbase (:TEXT-WINDOW) and the highlighting of the icon representing the Dresden airbase (:HIGHLIGHTED-ICON). The Dresden airbase is represented in the knowledge base by the SNePS node identifier B47.

3.2.3.4 PODS Accessor Functions

too detailed - belongs in a manual

Two functions exist to construct the PODS. One function, **add-to-pods-arg-list**, constructs the PO type list. This list contains all of the information needed to remove or generate a presentation object. The fields included in this list are; PO type, node list, and optional argument. The second function, **add-to-pods**, constructs the PODS given the functionality, recency, key, and the PO type list, which is returned from the add-to-pods-arg-list function call. A description of the input arguments to add-to-pods and add-to-pods-arg-list is in Section 3.2.3.3, Data Structure Format. The calling arguments for the functions follow: (**add-to-pods** *functionality key PO-type-list*) and (**add-to-pods-arg-list** *PO-type-list PO-type node-list window-identifier optional-argument-list*).

good, but not about functions

Lists are removed from the PODS whenever presentation objects are removed from a display or whenever the user deletes a window containing presentation objects represented in the PODS. Presentation objects are removed from a display based on the functionality and recency of the presentation object. Deleted windows are windows which are completely removed from the display and are not replaced by an icon. The deletion of a window is performed solely upon user request.

When a presentation object is removed from a display the PO-type list containing PO-type, node-list, and optional-argument is removed from the PODS. If all sub-lists contained in the window identifier list are removed, then the window identifier list is also removed. Similarly, if all sub-lists contained in the functionality list are removed, then the functionality list is removed. The removal of lists from the PODS when a presentation object is deleted is performed by the function **select-modalities-for-removal**. The calling arguments for this function are (**select-modalities-for-removal** *x-y time-of-removal key*). The first argument, x-y, is optional. It is used to delete mouse point gestures at a particular x/y location. The argument time-of-removal is used to indicate under what conditions a presentation object

where are?

is removed from the PODS. There are three values for time-of-removal indicating the three possible times or conditions under which a presentation object can be removed. The first case indicates a request to remove mouse point gestures. This occurs after the input has been parsed, but before any output generation. The second case is a request to remove all presentation objects generated during a mission presentation. The third case is a request to remove all remaining presentation objects. These presentation objects are removed based on the functionality and recency of the presentation object.

When a window is deleted by the user the function **remove-from-pods** removes all lists containing the identifier of the window being deleted from the PODS. If all of the window identifier lists contained in the functionality list are removed, then the functionality list is removed. The calling argument for this function is (**remove-from-pods** *window-identifier*). A description of window identifier is in Section 3.2.3.3.

3.2.4 Form Model

The Form Model consists of data and knowledge bases which enable CUBRICON to accept inputs to, and display outputs via the form display (see Section 10). It keeps track of information contained in the form, and coordinates the information on the form with the larger CUBRICON Knowledge Bases. The Form Model includes the following data structures: (1) the Mission Template; (2) the Form Display Model; and (3) The Informational Data Structure. Each of these is discussed below.

The Mission Template is a semantic network containing information about slots in a mission plan. The information required for defining a mission of a given type, and the relationships among those entities (e.g., a refueling orbit location and a refueling waypoint must be colocated), are defined in the Mission Template. The Form Model contains one Mission Template for each type of mission that might be planned.

The Form Display Model contains a representation of the forms themselves. It defines how forms are displayed. All form slot sizes and locations are specified. This enables CUBRICON to accept inputs to the form via diectic gestures, and allows outputs to be displayed in their proper location. Representations for many forms may be included within the Form Display Model, however in the current CUBRICON implementation, only one form is represented. Slots in the form are related to corresponding slots in the larger Mission Template.

The Informational Data Structure (IDS) keeps track of the information entered or displayed via the form, and relates that information to the CUBRICON Domain Knowledge Base and the form itself, in accordance with the structures of the Mission Template. When information is entered on a form, for example, data structures are created within the IDS defining the specific missions. These are created in accordance with the data relationships defined in the Mission Template and using domain knowledge contained in the Domain Knowledge Base.

(more detailed description forthcoming)

3.3 USER MODEL

Many aspects of a user are highly relevant to interface technology. These aspects include level of expertise in the current task, perspective based on his role, his value system, degree and nature of impairedness due to fatigue or illness, and preferences concerning mode of communication. Carberry [Carberry87] provides a brief summary of recent research on user modeling. To address all of these aspects of user modeling is, of course, beyond the scope of this project. The aspects of the user that are most relevant in the CUBRICON system are (1) the importance rating that the user attaches to the different entity types that are relevant to each given task, which we call the user's *entity rating system*; and (2) the task on which the user is currently engaged.

CUBRICON includes a representation of the user's entity rating system as a function of the task being addressed by the user. For a given task in the process being carried out by the user, the entity rating system representation includes a numerical importance rating (on a scale from zero to one) assigned to each of the entity types used in the application task domain. The numerical rating assigned to a given entity type represents the degree of importance of the entity to the user.

Associated with the entity rating system is a *critical threshold* value: Those entities with a rating above the critical threshold are critical to the current task and those with ratings below the threshold are not. This critical threshold is used by CUBRICON for determining which entities to display in response to user requests. This is accomplished as follows: (1) it is used in determining what information is relevant in answering questions or responding to commands from the user. (2) it is used in selecting ancillary information to enhance or embellish the main concept being expressed and to prevent the user from making false inferences that he might otherwise make. (3) it is used in organizing the form in which information is presented.

As an example of (1) above, if the user instructs the system to "Display the Fulda Gap Region", CUBRICON uses the entity rating system representation to determine what objects within the Region should be displayed. If the user is a military mission planner, then displaying all the country cottages in the region, for example, is irrelevant. The objects to display are those that are relevant to the job of the mission planner. Thus the objects that the system selects from its data base for display are airbases, missile sites, targets, etc. Section 8 discusses examples of the use of this entity rating system representation in interactive dialogue between a user and the CUBRICON system.

The CUBRICON design provides for the entity rating system representation to change automatically under program control in the following manner: (1) when the user's task changes the system replaces the current entity rating list with the standard initial rating list for the new task; and (2) when the user mentions an entity whose rating is lower than the critical threshold, then its rating is reset to be equal to the critical threshold to reflect the user's interest in the entity and its seeming relevance to the current task from the perspective of the user. In the current implementation, CUBRICON performs the only second function listed above. The implementation of the first function is not complete.

CUBRICON does, however, include a simple representation of the current task in which the user is engaged. CUBRICON's mode of response to the user is affected by whether or not the user's task has just changed. The CUBRICON team is developing a task hierarchy: a decomposition of the user's main tasks into subtasks. This a priori task knowledge can be used by CUBRICON to help track the discourse focus, manage the displays, and anticipate the needs of the user.

4 MULTI-MODAL LANGUAGE UNDERSTANDING

CUBRICON allows users to express themselves in a multi-modal way, using a variety of input media, much like persons talking to each other use every means at their disposal to get the message across. Multi-modal language understanding refers to the system's ability to accept input from all available input devices and interpret it in a consistent and coordinated way. The underlying viewpoint is that the different input devices should not be seen as separate sources of information but as parts of a single multi-modal input stream. Users are therefore free to mix different modalities and substitute expressions in one modality for equivalent expressions in another. This section explains how the different input modalities are read and interpreted in accordance with these principles.

4.1 MULTI-MEDIA INPUT COORDINATION

The system has to deal with three different input sources: spoken natural language as provided by a separate discrete speech recognition system, written natural language as provided by keyboard input, and pointing gestures as provided by mouse input. The three input sources are integrated into a single multi-modal input stream before any parsing and interpretation takes place. This process of integrating the input sources into a single stream is referred to as input coordination. We first discuss the input sources separately and then take a look at the integration process.

4.1.1 Speech

Speech input is provided by a separate Dragon Systems discrete speech recognition system running on a PC with additional hardware. It has a context free grammar in BNF-like notation that describes acceptable input sentences. Since this grammar is separate from the one used in the system's multi-modal language parsing and interpretation (see Section 4.2) there may be slight variations in the type of spoken versus written sentences that are accepted. This could, for example, include shorter spoken forms of long words. The main limitations of the speech recognition system are its discrete and speaker dependent nature.

Words recognized by the speech recognition system are read from a serial input stream and put into the input buffer of the natural language interaction window (see Section 4.1.4).

4.1.2 Written Language

The user can type in sentences in natural language using a keyboard. The standard Symbolics input editing facilities are available. The end of a sentence is detected when a period, question mark or exclamation mark is read, making the use of the return key unnecessary. Typed input is also sent to the input buffer of the natural language interaction window.

4.1.3 Gestures

Gestures are made using a standard three-button mouse, but the system never distinguishes between the buttons. The user can point to any object visible on either the monochrome or the color-graphics screen and click any mouse button to indicate the object being talked about. There are many different types of windows on the displays at any given time, for example tables, maps and forms, each containing several objects of different types. Pointing gestures may therefore be ambiguous with regard to the intended referent, and special routines are required to interpret them (see Section 4.2.2.4).

Apart from being used as a simple pointing device, the mouse can also be used to input graphical data. The only implemented instance of this is the entering of a flight path by the user. In this case the mouse is used to input a closed, directed polygonal path (see Section 4.3). The user indicates the vertices of the polygon on a map window on the color graphics screen (see Section 11), and the path is traced with directed line segments as each vertex is entered.

Just like speech and written language input, all mouse input is redirected to the input buffer of the natural language interaction window.

4.1.4 Coordination

The three available input sources are combined in a single input buffer associated with the natural language interaction window. This window echoes all input in a suitable printed representation (Figure 4). It also serves as a user feedback window where other parts of the system can display messages in natural language (see Sections 6 and 12). Words entered through the keyboard or speech recognition device are represented in their usual printed form, and mouse clicks are represented symbolically. The symbolic representation consists of a down arrow followed by a number (the ordinal number of the mouse click for the sentence being entered), and a list with X and Y coordinates, window name and window identification number. The printed representations of mouse clicks are provided for user feedback and to

That's the printed rep,
just what info is contained in a mouse click?
should be told in 4.1.3

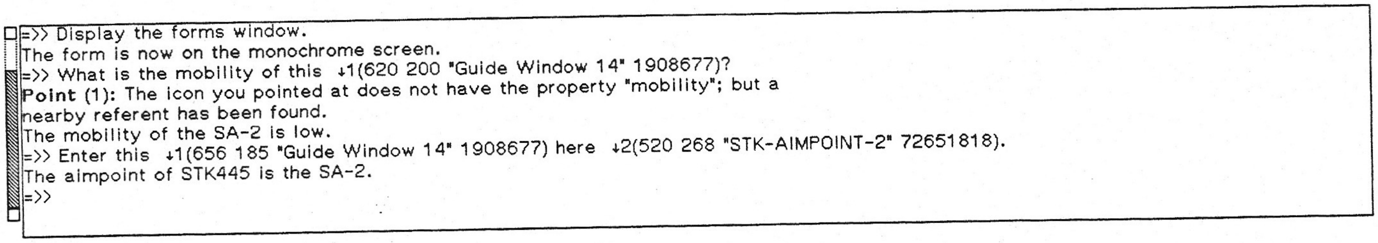


Figure 4: The natural language interaction window echoes all input in a suitable printed representation. It also displays natural language output produced by the system.

allow clicks to be edited out in case of mistakes.

Conceptually, the input sources may be seen as parallel and independent data streams before they are integrated. The combination of these streams proceeds in a linear way, inserting whatever is available from any source at the current point in the input buffer. The input buffer therefore reflects the order in which multi-modal tokens (spoken or written words or mouse clicks) were entered. The delays in processing input speech are insignificant in this respect, given a fast enough PC.

The integrated multi-modal input stream is processed by the parsing and interpretation component (see Section 4.2) on a per sentence basis. Before a sentence becomes available for further processing however, all words are checked for presence in the lexicon (see Section 4.2.1.2). The user is informed about any unknown words, is presented with a list of completions for incomplete multi-word expressions in the input, and asked to re-enter the sentence if necessary.

4.2 MULTI-MODAL LANGUAGE PARSING AND INTERPRETATION

After a Multi-Modal sentence has been assembled from the available input sources it is passed to the parsing and interpretation component. This component's task is to analyze the syntactic structure of the sentence (parsing) and assign a meaning to it (interpretation). Although we discuss parsing and interpretation separately they are really interweaved in time. The guiding principle is that syntactic representations serve only as intermediate structures until an interpretation (semantic structure) can be determined. As soon as possible, the syntactic representation will be abandoned in favor of a semantic one.

4.2.1 Parsing

The parser checks that the input represents a valid Multi-Modal sentence. This includes checking for valid English syntax that the system can understand, and checking that the mouse clicks occur in appropriate places in the sentence. Mouse clicks may occur anywhere within a noun phrase or a locative adverbial phrase, or can alternatively replace an NP completely. The parser builds parse trees for phrases and complete sentences, and it calls the appropriate interpretation functions that build the semantic representation of those phrases and sentences.

CUBRICON is using the parser that comes standard with the SNePS knowledge representation system, as described in [Shapiro89], with a few minor adaptations to handle Multi-Modal sentences.

4.2.1.1 The Grammar.

The grammar used by the parser is written in the form of a Generalized Augmented Transition Network, [Shapiro82]. For a general introduction to ATN grammars, [Bates78]. The same type of GATN grammar is also used in the system's natural language generation component (see Section 12). A description of the input grammar appears in Appendix B.

4.2.1.2 The Lexicon.

The lexicon is a dictionary of words that the system understands. Each word is associated with one or more lists of syntactic, morphological and semantic features. Multiple lists are associated with lexically ambiguous words (e.g. like "start" which can be a noun or a verb). The syntactic and morphological features are used by the parser to determine sentence structure [Shapiro89]. They include things like syntactic category, multi-word lexeme status, number and root form. Semantic features specify meaning-related attributes of words, for example, the key or function to be used in accessing the knowledge base for information about the concept the word represents (see Section 3.1), case frames associated with verbs (see Section 4.2.2.1) or special media-related attributes of the corresponding concept. Examples of the latter are words like "monitor" or "map" that are classified in the lexicon as special kinds of "display objects", or form slot names (see Section 10) that are listed with a feature indicating how to retrieve information from the corresponding slot.

4.2.2 Interpretation

The interpretation routines take the parse trees produced by the syntactic parser and fill in the semantic slots in those trees. There are interpretation routines that operate on parts of the sentence (single nouns, mouse clicks or phrases), and separate ones that attempt an interpretation of the entire sentence, given a successful interpretation of the parts. An interpretation of a phrase is attempted as soon as it has been parsed, although that is not always possible. Interpretations for phrases are added as a separate "interpretation slot" to their parse trees. Sometimes interpretation of, for instance, a noun phrase or a mouse click, must be delayed until more contextual information is available from the rest of the sentence. We now discuss some more specific interpretation issues.

4.2.2.1 Verb Case Frames.

A case frame is associated with each verb in the lexicon. It serves as a skeleton for the semantic structure that eventually results from parsing and interpreting a sentence. A case frame has slots that are associated with a specific semantic role. Possible slots are *agent, object, action, value, recipient, location* and *context*. The representation of the case frames in the lexicon includes either a value for the slots (e.g. the value *display* for the *action* slot associated with the verb "display") or a directive for the interpretation routines to come up with a value (e.g. *match direct-object* for the *object* slot associated with the verb "display", meaning that the syntactic direct object of the sentence corresponds to the semantic case frame object).

The *agent* slot refers to the agent of an action, and is usually interpreted as *system*, i.e., CUBRICON is to perform the specified action. The *object* slot refers to the object that the action is being performed on, i.e., some domain knowledge base object or a more interface-related object like a window. The *action* slot refers to the kind of action to be performed, for example *display, enter, present* (see Section 5). The *value* slot may be used to refer to a value being assigned to some object, e.g the value "current mission" to a mission plan (see Section 5). The *recipient* slot refers to the recipient of an object, for example a mission plan for which a flight path is to be planned. The *location* slot refers to a location in a broad sense, for example a table on which an object is to be highlighted. The *context* slot, finally, may refer to some kinds of contextual information such as the mission plan associated with another object mentioned in an input sentence.

The final interpretation of an input sentence is always an instantiated case frame; i.e. a case frame with some or all of its slots filled in. This case frame is passed to the executor (see Section 5) as the meaning of the sentence.

4.2.2.2 Mouse Gestures.

Special interpretation functions take care of the mouse clicks that appear in noun phrases or prepositional phrases. They use various kinds of information stored in the parse tree or partial parse trees that have been produced by the parser, to determine a referent for the point. Section 4.2.2.4 discusses multi-modal referent determination in more detail.

A referent for a mouse click in a noun phrase is always a knowledge base object, representing an air base for example. Contextual information can be used to determine a referent, such as when the user asks about a property of an object, like mobility. If the user clicks on an icon representing an object that doesn't have the property in question, the system will try to find a nearby visible object that does, using an incremental bounded search. The same happens when the mouse click did not hit any icon at all, or multiple overlapping icons. When this incremental bounded search finds a visible object of the right kind within a maximum distance of the clicked location, it will be returned as the referent for the click. The user will be informed about the "near miss" on the mouse point and the object found through the natural language window. If no referent can be found, an error message is sent to the same window.

redundant w/ 4.2.2.4

Mouse clicks that appear in locative adverbial phrases are not interpreted as referring to an object but as referring to a location. In this case no further interpretation is needed than to return the coordinates of the point and the window it occurred in. Examples of this kind can be found in interactions with the form window (see Section 10), in sentences like *Enter this <click> here <click>*. The first click refers to an object on one of the displays, the second to a location on the form. Note that the first click may also be entered on the form, in which case it will be taken to refer to the object mentioned in that particular form location.

4.2.2.3 Noun Phrases and Prepositional Phrases.

Noun phrases are interpreted as referring to knowledge base objects. We discussed the case of mouse clicks within phrases already. To determine the interpretation of a single noun, its semantic type is looked up in the lexicon. This semantic feature indicates the type of word being dealt with, and is used to direct the interpretation functions in their search for a suitable knowledge base object. The most important type values we use and their meaning are summarized in Table 2. One word can have multiple semantic types, and other words besides nouns have semantic types too.

The semantic type values associated with prepositions are used to direct the interpretation of prepositional phrases. A preposition like *around*, for instance, is interpreted as having a particular spatial meaning. Therefore prepositional phrases like *around the Dresden air base* are interpreted in a two step process: first the referent for the noun phrase *the Dresden air base* is determined, resulting in a specific knowledge base object, and then the knowledge base is searched for objects that are physically close to this object, as determined by computing the real-world distance between them from their geographical coordinates. The type of objects being considered depends on context. In a noun phrase like *the threats around the Dresden air base*, only objects that are considered to be threats are taken into account, like other air bases and SAM systems.

4.2.2.4 Referent Determination for Multi-Modal Phrases.

As mentioned before, the mouse point reference determination functions can handle typical problems like ambiguous pointing gestures that touch two or more icons or gestures that miss the intended icons completely. The dereferencing process depends on a number of factors such as the types of objects being referenced, the properties of those objects, the sentential context and the constraints on the fillers of the semantic case frame slots for the main verb of the sentence.

We mentioned a few cases of referent determination for multi-modal phrases. Mouse clicks can be substituted for single nouns or for entire noun phrases, so a noun phrase like *this air base* is equivalent with the multi-modal noun phrase *this <click on an air base icon>* or even the minimal form *<click on an air base icon>*. A noun phrase can also contain or consist of multiple mouse clicks, referring to multiple objects. But apart from these rather straightforward cases, the system is also able to interpret more difficult ones. In some cases the multi-modality of the phrases is used explicitly to determine a referent when that would not be possible for an equivalent single-modal phrase. We consider this cross-modal dereferencing capability a unique feature of our multi-modal system architecture, which shows that the multi-modal total is bigger than the sum of its parts.

In some cases the natural language content of a phrase is used to disambiguate an otherwise ambiguous pointing gesture. Consider for instance a phrase like *this air base <click>*, where the pointing gesture touches an object of a different kind, say a factory. The immediate interpretation of the gesture is the factory corresponding to the touched icon, but since the

SEMANTIC TYPE	MEANING	EXAMPLE
agent	refers to an agent	*I*
attribute	refers to an attribute of an object	*disposition*
class-property-value	refers to the value of a property of a class of objects	*mobile*
component	refers to a component of an object	*aircraft unit* (component of an air base)
field	refers to a form slot	*AC pool*
find-or-create	refers to an object that may be created if it doesn't exist yet	*PKG0026* (a set of related mission plans)
instance-property-value	refers to the value of a property of an instance of a class of objects	*friendly* (value of the *disposition* property of a SAM system)
literal	passed on to the executor (see Section 5) without being interpreted	*current package*
location	a preposition flagging a locative phrase	*in*
name	a proper name	*Allstedt*
object	refers to an object	*air base*
part	refers to a part of an object	*radar* (part of an air base)
property	refers to a property of an object	*damaged*
screen-object	refers to a screen object	*forms window* (see Section 10)

Table 2: Values of the semantic type feature associated with words in the lexicon, with their meanings and some examples. See Section 3.1 for more information on the semantic structure of the domain knowledge.

natural language part of the phrase indicates an air base, this interpretation is rejected. In stead an incremental bounded search is performed to find an air base close to the indicated location. Another example is a pointing gesture that touches two overlapping icons, for instance representing a city and an air base. Without further information the system will not be able to infer which is meant and will return both as intended referents. However, if the accompanying natural language mentions either a city or an air base, only the corresponding referent is returned. Circumstances like these can easily occur in a graphics window that contains a lot of icons representing different objects. As a final example, consider the phrase *the mobility of these <click 1> ... <click n>*. If any of the mouse points do not touch an object with the specified attribute *mobility*, or do not touch any objects at all, an incremental bounded search will search for possible referents having that attribute, in the neighborhood of the indicated points.

As an example of disambiguation in the other direction, viz. graphical gestures disambiguating natural language, we may consider the previously discussed sentence *Enter this <click> here <click>*. Without the graphical gestures this sentence is obviously not interpretable, since both *this* and *here* can refer to many different objects and locations, respectively. But given an interpretation for the graphical gestures, an object visible as an icon and a form slot, the sentence is perfectly interpretable and unambiguous. In fact the disambiguation process is bidirectional in this case. The adverb *here* signifies a locative phrase, thereby directing the interpretation functions to search for something corresponding to a location in the given context (the kind of window that the point is entered on), viz. a form slot. In contrast with the second point, the first one is interpreted to refer to a domain object and not to a location on the form (an interface object) because of the syntactic context and the pronoun *this*. The syntactic context alone will suffice in this case, so even if the pronoun is omitted the sentence will still be interpreted correctly. In fact if both mouse points are entered on the same form slot, and if that slot already contains a (description of) a domain object, the net result is that the content of the slot will be overwritten with what was already there before, since the first point is taken to refer to the content and the second to the slot itself. Although not very useful, this example is a good indication of the dereferencing capabilities of the system.

4.3 FLIGHT PATH DEFINITION

The interactive definition of a flight path by the user is a special case of multi-modal language understanding. It is not driven by the input parsing and interpretation routines, but called as a separate function by the executor (see Section 5) in response to a user request like *Plan a flight path for OCA111*. As such, it is more part of the domain-specific component of

CUBRICON, although it uses the same kind of graphical gestures and interactive techniques as other forms of input. The system will aid the user in entering the path by highlighting the appropriate information on the map window and through a combination of the available modalities, and deciding when the path is complete based on its domain knowledge.

When locations on a color graphics map window are being clicked by the user as part of the flight path definition, they are interpreted as referring either to objects visible as icons (air bases and the like) or to geographical locations. Each point is treated as a vertex in a closed polygonal path defining the flight path.

A number of constraints must be satisfied for a flight path to be accepted as valid: it must include the origin air base and target of the mission, for instance. Additional information such as the time passed since take-off is computed as the path is being entered.

5 EXECUTOR

After a unique interpretation has been determined for a multi-modal input sentence, that interpretation is handed to the executor. It is the responsibility of this component to take the appropriate action. The type of action is partly dependent on the target application, and in our case can consist of a command to the mission planning system, a database query, or an interface manipulation request. The results of these actions are handed to the multi-modal output planner (see Section 6) which controls communication of the results to the user.

5.1 TYPES OF ACTIONS

A number of actions are concerned with manipulations of the interface itself. Although the user rarely manipulates the interface directly, many user requests will result in this type of action, for instance when the system is being asked to display some object. Some of these actions are *display, present, highlight, zoom in* and *blink.* They are discussed more fully in Section 6. It is the executor's task to extract the appropriate information from the interpretation of the input sentence, and hand it to the output planner along with the specific request.

Other types of actions are more concerned with the application domain itself. For these actions, either a knowledge base search or some other domain-specific action is executed, and the results are handed off to the multi-modal output planner to be presented to the user. Actions that belong in this category are summarized below.

Identity query.

This type of action results from a user question about the identity of an object, typically with sentences like *What is this <click>?*. The knowledge base description is passed to the output planner to be presented.

Yes-no query.

This action results from a user's yes-no question (truth question), for example *Is this <click> a SAM?*. The knowledge base is queried for the class of the object, and the result is passed to the output planner.

Location query.

A sentence like *Where is the Nuernberg air base?* results in a knowledge base search for the location property of the object referred to. The node representing this information is passed to the output planner.

Property query.

This action is taken when the system is queried about a property of an object, typically with questions like *What is the mobility of this <click>?*. The knowledge base is searched for the value of the requested property of the indicated object, and the result is passed to the output planner.

Component query.

A component query is executed in response to questions like *What units are at the Dresden air base?*, where the requested objects (units) are in a component-of relation to the specified object (Dresden air base). The knowledge base is searched for all components of the object, and a list of them is passed to the output planner.

Characteristic and part queries.

These actions result from questions like *What is the STN mission plan for SVC1002?* and *What radars are at the Dresden air base?*, respectively. They search for objects that are in a characteristic-of and part-of relation to the object referred to, and pass them as a list to the output planner. The characteristic-of relation is subdivided into a number of named characteristics. They are comparable to properties of objects, but are defined only for non-tangible objects such as mission plans.

Specification of property values.

This action is taken in response to declarative sentences like *The target of this mission is Dresden runway?*. The knowledge base is updated to reflect that the object referred to (the mission) has the specified value (Dresden runway) for the specified property (target).

Assigning values to variables.

Some application-domain variables are maintained by the system and can be assigned values through this type of action. There are a few different cases to be distinguished, which we will do by giving some examples. One type of assignment occurs in response to input sentences like *Make PKG0026 the current package*, where *PKG0026* refers to a knowledge base object (which may be pointed to on a table or referred to in some other multi-modal way, as always) and *the current package* is returned as an uninterpreted string from the interpretation functions. The latter is used to identify the particular variable to be assigned a value. Another type of assignment action is taken in response to sentences like *Assign SVC001 to OCA123*, where the former is assigned as a sub-mission to the latter, and both are existing knowledge base objects. The third type of assignment action is used to assign a value to a newly created property of an object, for example with a sentence like *Assign SVC001 a duration of 20 minutes.* The result is that a new property "duration" is created for the object referred to by *SVC001*, and it is assigned the value "20 minutes". This is not a variable assignment is the traditional sense, but rather a creation of a new piece of knowledge base structure.

Starting a new mission.

A new mission of the appropriate class is started in response to sentences like *I am starting an OCA mission plan.* This means that the appropriate knowledge base object is created as a member of the specified class.

Naming objects.

This action is taken to give a name to a knowledge base object. It is typically executed in response to a sentence like *Call it OCA555*, which will usually be preceded by one like *I am starting an OCA mission plan.* The result is that the (anaphorically) referred to knowledge base object is given the specified name.

Changing the current task.

The system maintains a notion of the current task (see Section 3.2.1), which can be changed in response to sentences like *I am working on the OCA123 mission plan.* This results in a system variable for the current task being given a new value. The current task is changed implicitly when a new package is selected, as described above.

Planning a flight path.

The flight planning action is taken for input sentences like *Plan a flight path for OCA123*. The executor invokes the flight path definition function (see Section 4.3) which accepts user input and builds the structures corresponding to the flight path into the knowledge base.

Listing class members.

Sentences like *List the packages* or *List the SA-3s* results in a knowledge base search for all members of the specified class. The found objects are passed to the output planner as a list.

Saving the current package.

An input sentence like *Save the package as PKG2222* will cause the current package to be saved to disk under the given name. Retrieving it is done with a sentence like *Make PKG2222 the current package*, as indicated before.

Entering information on the mission form.

This action enters a piece of information in the knowledge base structures corresponding to the mission form, which is a graphical representation of the information associated with the current mission plan package. It is taken in response to sentences like *Enter this <mouse click> here <click>* or *Enter the Nuernberg air base as the origin of OCA111.*

6 MULTI-MODAL OUTPUT PLANNER

The Multi-Media Output Planner composes the response that is to be produced to the user by the Output Generator in coordinated multi-modalities. The Output Planner determines the media and modalities for expressing the response information to the user, but then must determine whether the resources are available in order to do so. If they are not, then the Planner must take appropriate action to modify the state of the resources, modify the information to be expressed, and/or select different modalities for expressing the information before the composition of the output can be accomplished.

The top level output planning process is summarized below. This planning process presupposes that the primary relevant information has been obtained to respond to the user.

1. Assess the availability of the monochrome and color graphics devices. If none of the window positions on the monochrome device are available and there are window positions available on the color graphics device, then the color graphics device is the preferred device. This would supersede the monochrome device as the preferred media for the table modality.

2. For each information item or cluster, determine the modality in which it should ideally be expressed. Graphic/pictorial presentation is always desirable. Natural language can always be used, as a last resort if no other modality is available.

3. Determine whether the resources are available to express the information as desired. Resources: (1) Color graphics display: Are the items to be expressed graphically already on the color display (e.g., objects of interest in a geographical domain may already be displayed on a map)? If so, no additions are necessary. If not, is there room to add them in their "natural" position? (e.g., can the desired objects be inserted in the area already on the graphics display without changing the area shown or does the displayed area need to be extended or changed totally?) (2) Monochrome display: Similar to the color graphics display. (3) Speech output device: Always available.

4. If the desired resources are not available, modify the state of the resources. The desired resources would be "not available" if the device (e.g., a display) already contains critical information that cannot be disrupted nor covered by a window. For the graphics displays, if not all the items to be expressed graphically are on the graphics display, then the system must compose a new display. Borrowing terminology from the geographical situation, the possible cases are:

- *Zoom out* with intelligent addition of relevant ancillary objects to fill in the new area to maintain consistency throughout the display.
- *Zoom in* with intelligent addition of relevant objects to create an intelligible display.
- *Pan* to a different area maintaining consistency in the types of objects displayed.
- Combination of the above.
- Display a different disjoint area. (i) Completely replace display with new "area" or (ii) Open a window on the monitor to show new information.

A detailed explanation of the methodology used to dynamically compose geographic maps is in Section 11.1.

5. If the desired resources are still not available to accommodate the information to be expressed, try modifying the information to be expressed: trim the amount of information by filtering on the basis of relevance with regard to user model and/or discourse model.

6. If the information can still not be expressed in the given modality due to insufficient resources for the selected modality, then select another modality and go back to step 3.

7. Compose the output, having resolved resource constraints.

8. Repeat the modality selection and generation process until all modalities have been evaluated.

6.1 MODALITY SELECTION

Selection of the most appropriate modalities for expressing information in the CUBRICON system is based on the nature and characteristics of the information. Our system design is based on the premise that graphic/pictorial presentation is always desirable. The following is a brief summary of the selection criteria.

1. *Map*: Selected whenever CUBRICON knows how to represent the information pictorially.

2. *Table*: Selected when the values of common attribute(s) of several entities must be expressed.

3. *Form*: A predefined form is selected when the task engaged in by the user requires the form.

4. *Pointing Gesture*: Pointing gestures are selected whenever an object or the property of an object is requested, so that the attention of the user will be drawn to the object. Three types of pointing gesture modalities exist; map, table and form.

5. *Text Window*: Text windows are selected whenever textual information is desired.

6. *Natural Language Prose*: Selected for the expression of a proposition, relation, event, or combination thereof, when the knowledge structures being expressed are heterogeneous. Natural language can be presented in either spoken or written form.

The selection of the media and modalities in which to express the response information to the user is based primarily on SNePS nodes and/or a command which results from the parsing and interpretation of the user's request. In addition, the selection of some modalities depends on the modalities previously generated. The selection of modalities is done sequentially, evaluating modalities in order of importance. The following sections describe the types of modalities, described in order of preference.

6.1.1 Direct Window and Modality Manipulation

Although CUBRICON attempts to manipulate widows and modalities without direct user intervention, it does allow the user to directly manipulation in a limited sense. The selection of these modalities is based solely on the command input to the Output Planner. Too low level

The window manipulation modality, expose window, is chosen whenever the input command argument is set to :expose. This modality is selected based on the user input "Expose this window <point>.". The result of this modality is that the window pointed to is brought to the foreground, overlaying any overlapping windows.

Another window manipulation modality, remove window, is chosen whenever the input command argument is set to :remove. This modality is selected based on the user's request "Remove this window <point>.". The result of this request is that the window pointed to is removed and transformed into a map icon. low level

CUBRICON allows the user to request that the modalities presented during a flight path presentation be removed. The modality remove flight path is selected whenever the input command to the Output Planner is :remove-all-mission which is generated as a result of the request "Remove the flight paths.".

6.1.1 could be reduced to one para.

The user may save information entered into a mission planning form with the request "Save the form.". The input command to the Output Planner for this request is :save which is the criteria for the save form modality.

6.1.2 Map Modality Selection Criteria

The map modality is selected whenever CUBRICON can express the information pictorially. Two types of map modalities exist; the presentation of a geographic area containing one or more map icons and the presentation of a geographic region. The criteria for selecting these modalities based on the SNePS node(s) and command input to the modality selection function as follows:

what's the difference?

- Zoom In

  low level

  The zoom in modality is selected whenever the command :zoom-in is input to the modality selector. This command represents a user request to zoom in on a geographic area by stating "Zoomin on this point <point>.".

- Map Icon

  - The input nodes represent the assertion that the objects are a part of an airbase, and there does not exist a map which contains all of the objects represented in the input nodelist. A user request which would generate a call the modality selection function with these nodes is "What are the aimpoints within the Merseberg airbase?". The output generated as a result of selecting this modality is a part-whole decomposition map containing the icons representing the objects which are parts of the Merseberg airbase.
  - The input node(s) represent an object or class instance whose superclass can be represented as a map icon and at least one of the objects being requested are not on an active map. One user request which generates this case is "Blink the heliports." which inputs a node representing the class of heliports to the Modality Selector. The result of this request is to add all map icons which represent heliports and are located within the boundary of an active geographic map to the map.
  - The input nodes represent the assertion that an object is located at a particular latitude and longitude and at least one of the objects whose location was requested are not on an active map. One user request generating this case is "What is the location of the Merseberg airbase?". The resulting output generated by the map icon modality is a map containing an icon representing the Merseberg airbase.

- Region

  The region modality is selected whenever the input node represents the instance of a region and there does not exist a map which contains the area defined by the region. A region node has a latitudinal and longitudinal boundary defined for it. A user request which results in the selection of the region modality is "Display the Fulda Gap region." which generates a map containing the area included in Fulda Gap region boundary.

6.1.3 Map Pointing Modality Selection Criteria

The pointing gesture modalities are selected whenever an object or the property of an object is requested, so that the attention of the user will be drawn to the object. Several types of objects can be referenced by the Output Planning system; windows, icons, table rows and form panes. The modalities representing gestures which point to geographic objects, windows and icons, are selected and generated following the Map Modality. The remaining pointing gesture modalities are selected following other modalities. A detailed explanation of the pointing modality is presented in Section 8, Deictic Gestures.

- Window Pointing

  Window pointing occurs whenever a geographic map is requested and it is contained in an active map. One case which generates the window pointing modality is whenever the input node represents the instance of a region and a map exists which contains the regional boundary. In this case the input node is identical to the node input in the region modality described above. The second case which generates the window pointing modality is whenever the input nodes represent the assertion that the objects are a part of an airbase, and a map exists containing all of these objects. Once again, these input nodes are identical to the nodes input in one of the map icon modality cases described above.

- Map Icon Pointing

  Map icon pointing occurs whenever the Cubricon system can express information graphically, without modifying the display, and a subset of the icons on a map are being pointed to. The input nodes which meet the map icon pointing criteria are identical to the nodes input in two of the map icon modality cases described above. The input node represents either an object whose superclass can be represented as a map icon, or the assertion that an object is located at a particular latitude and longitude.

- Highlight

The highlight modality highlights map icon(s) and/or table entries based on user request. The highlight modality is selected whenever a highlight command is input to the output planner. The object to be highlighted is represented by the input node(s). The highlighting occurs on every window containing the object, unless a specific window has been requested by the user. In this case the window to be highlighted is passed to the output-planner in the destination-window parameter. A user request which generates the highlight modality is "Highlight this <point at the Nuernberg airbase> on the table.". The output planning system is passed a command highlight, a node representing the Nuernberg airbase, and a destination window which represents the table which is related to the map on which the input point gesture occurred. Based on this information the highlight modality is chosen and the table entry containing the Nuernberg airbase is highlighted on the destination window.

6.1.4 Table Modality Selection Criteria

Generally, the table modality is selected when the values of common attribute(s) of several entities must be expressed. There are two conditions which determine whether the table modality is appropriate. The first condition is whenever a map was created during this response, to display the important attributes of the entities displayed on the map. The table modality is selected since all of the icons on the map have common attribute values and it is not necessary to evaluate each attribute being displayed. The second condition is whenever the input command has a value of list, present or nil, the SNePS nodes input have at least one common attribute and there are more than four nodes whose attribute values are to be expressed. When determining if common attributes exist not all properties are considered, the applicable properties are those defined in the user model for each node. low level?

Two types of table modalities exist monochrome table and color-graphics table. The monochrome table modality generates tables which are placed on the monochrome device, whereas the color-graphics table modality generates tables which are placed on the color-graphics device. Within the hierarchy of modalities monochrome tables are the preferred, unless the Modality Selection system has determined that the color-graphics display has window positions available and the monochrome display does not. It is possible for the Window Management system to reject a request to create a monochrome table, due to the monochrome device being unavailable. If the Modality Selection system attempts to create a monochrome table and the Window Management system rejects this request, then the Modality Selector will choose the color-graphics table as an alternate modality. The same table, however, would not be presented in multiple modalities. A detailed explanation of the table window placement algorithm is in Section 7.3.

6.1.5 Table Entry Pointing Modality Selection Criteria

As previously mentioned one of the types of pointing gesture modalities is table entry pointing. Pointing to a table entry occurs whenever the information to be expressed is on an active table. The objects represented in the input node list are compared with the content list of each active table. If the object is contained in the table, then the corresponding entry is pointed to. The type of nodes which meet the table entry pointing criteria are identical to the nodes which generate the map icon pointing modality, which is described above.

6.1.6 Form Modality Selection Criteria

A predefined mission planning form is selected when a form is requested by the user, the SNePS node input to the modality selection function is the node representing the instance of a window and the command :display. This node is then used as a key value to get the window identifier from the monochrome window list data structure. If the type slot in this window is set to form, then the mission planning form associated with the current task is displayed. Section 10 provides a detailed description of the form modality. low level

6.1.7 Form Pane Pointing Modality Selection Criteria

An additional type of point gesture modality is form pane pointing. Form pointing occurs whenever the information to be expressed is on an active mission planning form. If the object or property being expressed is contained in the form, then the corresponding entry is pointed to. The node input represents the assertion that the property of a mission is assigned a value. One user request generating the form pain pointing modality is "Enter the Nuernberg airbase as the origin.". A network diagram showing the assertion that the Nuernberg airbase is the origin of the OCA0999 mission plan is *Sample OCA Mission Plan.*

6.1.8 Selection Criteria for the Text Window Modality

The text window modality is selected whenever textual information is desired. Text windows are utilized as part of integrated multi-media/multi-modal modalities when composing the point at modality and when generating a mission presentation. The text window modality, however, is selected in only one case, whenever the input nodes represent the assertion that an object has a property which is assigned a value, and there are less than four nodes input. An example network diagram showing the structure of the input node is *SA-2 Class.* The property name and value associated with the input node are placed on a map next to the icon representing the object. The threshold of five input nodes is used to minimize map clutter. If more than four nodes are input a table modality is chosen. An example user request which selects the text window modality is "What is the mobility of these <point1> <point2>?". The response generated by the text window modality is a text window containing a string such as "mobility: low" placed next to the icon whose property was requested, where low represents the concept that the value of the property mobility is low.

6.1.9 Natural Language Prose Modality Selection Criteria

Natural language prose is selected for the expression of a proposition, relation, event, or combination thereof, when the knowledge structures being expressed are heterogeneous. Natural language can be presented in either spoken or written form. The following summarizes the selection criteria for spoken versus written language

- Spoken Natural Language
  - Dialogue descriptions to assist the user in comprehending the presented information. These include explanations of graphic displays or display changes and verbal highlighting of objects on the displays (e.g., "The enemy airbases are highlighted in red").
  - Informing the user about the system's activity (e.g., "I'm still working" when the user must wait for output from the system).
  - Short expressions of relatively non-technical information that can be remembered when presented serially (e.g., a "yes"/"no" answer to a user's question).
- Written Natural Language Selected for longer technical responses that would strain the user's short term memory if speech were used (see [Miller56]).

6.2 OUTPUT COMPOSITION

Most frequently, multiple modalities are desirable to express a body of information to the user. For example, to inform the user about the movements of a certain tank battalion, a desirable presentation might be an explanation delivered in combined spoken speech and coordinated drawing on a graphic map display showing movements of the battalion, as well as a printed textual summary with ancillary information on the monochrome display. The multiple modalities should be selected to complement and enhance one another. Andriole [Andriole86] has used "graphic equivalence" effectively using dual displays or split screens to present the same material in different forms to aid user comprehension and problem solving performance. We are not restricting the system to presenting the *same* material in different forms, but, instead, our system presents related material or different aspects of a given event or concept in different forms/modalities (as appropriate based on the nature and characteristics of the information).

The CUBRICON system rarely restricts output to one modality typically multiple media and modalities are selected. Written and Spoken Natural Language, for example, are utilized in nearly every output presentation. In general, if CUBRICON represents and object graphically (eg. the location of an airbase is requested) output generation combines Map Icon and Icon Pointing modalities on the color graphics display, the Table and Table Pointing modalities on either the monochrome or color graphics display, Written Natural Language on the monochrome display, and Spoken Natural Language via the speech output system. In this example, the tabular presentation was selected because there are important attributes associated with the entities displayed on the map display. Specific examples illustrating the composition of multiple media and modalities are presented in the next section, Multi-Media and Multi-Modal Output Examples.

6.3 MULTI-MEDIA AND MULTI-MODAL OUTPUT EXAMPLES

In order to further illustrate CUBRICON's modality selection and output composition process, consider the next user input. The user queries the system about the location of the Nuernberg airbase in a manner that provides no instruction to the system as to how to present the information (e.g., map, natural language only, etc).

USER: "Where is the Nuernberg airbase?"

DEVICE CONFIGURATION:

> The color graphics display contains a map displaying the Fulda Gap region and a table showing the important attributes of the objects is displayed on the map. The monochrome display contains a mission planning form.

CUBRICON: (Refer to Figure 5.)

Speech output:

- Statements to direct the user's attention to the appropriate monitor when a major window is presented. As the map is expanded on the color monitor: "The map on the color graphics screen is being expanded to include the Nuernberg airbase."

Color Graphics Display:

- Map of the Fulda Gap Region with added area that includes the Nuernberg airbase.
- Main roads, major cities, waterways, and national boundaries (as before but across the whole map, old and new areas).
- Icons representing entities within the new map area displayed that are above the critical threshold on the entity rating system for the user's task.
- An airbase icon representing the Nuernberg Airbase.

Speech Output with coordinated Color Graphics:

- After the map is expanded, statement to direct the user's attention to the Nuernberg airbase on the map: "Its location is here <point> 50 miles southeast of the East-West Germany border." The word "here" is accompanied by a visual point gesture in the form of blinking the airbase icon and the addition of a pointing text box.

Written Natural Language:

- A written an more detailed version of the previously spoken response is "Its location is 50 miles southeast of the East-West Germany border.".

Speech Output:

- As the table is presented on the monochrome monitor: "The corresponding table is being generated" and "The corresponding table is now on color graphics screen."

Monochrome Graphics Display:

- Table of relevant entity attributes. Same table as before, but expanded to include the new entities added to the map covering the extended area.
- The table entry containing the attributes for the Nuernberg airbase is highlighted.

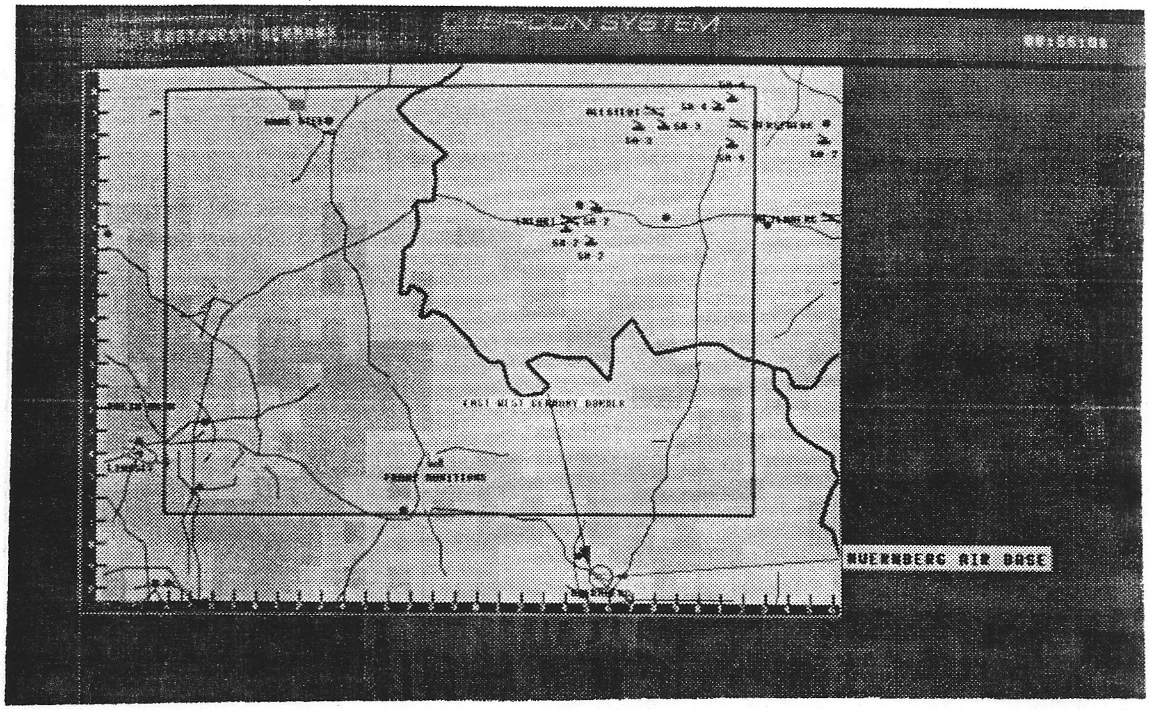
DISCUSSION:

As previously discussed, whenever possible the CUBRICON system prefers to present information to the user graphically with ancillary information presented simultaneously in an another modality. Since CUBRICON knows how to display an airbase graphically (it has an icon associated with the class in the knowledge base), and since each particular airbase in the knowledge base has an associated geographical location the Map Icon modality is selected. Then the system will display the airbase on the color-graphics map with additional information displayed in another modality. If the Nuernberg airbase is already displayed on the color map display, then the system would choose to blink the particular airbase icon as its way of pointing to the object and accompany this pointing action with a spoken response. If the Nuernberg airbase could be added to the current map, it would do so and direct the user's attention to the airbase icon as mentioned above. However, the Nuernberg airbase is outside of the region shown in the map display currently on the color CRT. Therefore the resources needed to present the Nuernberg airbase graphically are unavailable. The system must now decide how to modify the state of the resources to show the airbase. What map should be displayed?

In composing a new map on which to display the Nuernberg airbase, the system has some choices. These choices include: open a window on the color graphics display showing the area around the Nuernberg airbase, replace the old map on the CRT with a new area around the Nuernberg airbase, or compose a new map including both the old map and the region around the Nuernberg airbase.

An important guideline to which the CUBRICON system tries to adhere is to maintain the context of the user-computer dialogue. With regard to the graphic displays, this means that the system tries to retain the most recently discussed or mentioned objects on the displays so as to maintain continuity in the dialogue. The discourse focus space representations, discussed in Section 3.2.1 are the key knowledge sources in this process. The system composes a new map containing the objects that are on the old map as well as the Nuernberg airbase. The algorithm that the system uses to determine the boundary for a new map of this type is to determine the smallest rectangle that encloses the old objects on the current map as well as the new objects to be displayed and then add a small "border" area around all sides. This essentially extends the area shown to include both the old and new objects.

Another important guideline to which the CUBRICON system adheres is to maintain consistency throughout a display so as to prevent the user from making false inferences about what is or is not located within the region. In the case of our map display, this means that there should be consistency in the types of objects shown across the entire map. If SAMs are displayed in the old region, then they should be displayed in the newly added map area. Similarly for other types of objects. If this is not done, then the user would probably infer that there were no SAMs in the new area since he sees none on the display in the new area, when in reality there are SAMs in the new area. Figure 5 shows the new map display composed by CUBRICON in response to the user's input "Where is the Nuernberg airbase?" The rectangular outline within the map is used to indicate the previously displayed area. This provides graphic context: the new entities in the context of the previously displayed area.



Entities in the Expanded Region

Item	Disposition	Latitude	Longitude	Name	Mobility
air base	friendly	50.300N	8.390E	Rhein Main	--
air base	friendly	50.050N	8.330E	Lindsey	--
air base	enemy	51.400N	11.460E	Allstedt	--
air base	enemy	50.970N	10.960E	Erfurt	--
air base	enemy	51.350N	11.960E	Merseberg	--
air base	enemy	50.980N	12.510E	Altenberg	--
air base	friendly	49.550N	11.150E	Nuernberg	--
SA-2	enemy	50.933N	10.933E	--	low
SA-2	enemy	51.016N	11.116E	--	low
SA-2	enemy	50.883N	11.083E	--	low
SA-2	enemy	51.283N	12.466E	--	low
SA-3	enemy	51.338N	11.516E	--	high
SA-3	enemy	51.335N	11.366E	--	high
SA-4	enemy	51.266N	11.921E	--	low
SA-4	enemy	51.416N	11.838E	--	low
SA-4	enemy	51.450N	11.921E	--	low
plant	friendly	51.421N	9.344E	Hans Steel	--
factory	friendly	49.991N	10.152E	Franz Munitions	--

Its location is 30 miles southwest of the East-West Germany border .
=>> Where is the Nuernberg airbase?
The map on the color graphics screen is being expanded to include the Nuernberg air base.
Its location is 50 miles southeast of the East-West Germany border .

The corresponding table is being generated.

The corresponding table is now on the monochrome screen.
=>>

Figure 5: Map and Table Maintaining Context and Consistency

Based on the information provided by the user/task model, CUBRICON knows the important attributes of each object. The table modality is selected to present this information. Guided by the consistency principle, the system also modifies the tabular presentation that is on the monochrome display to include the additional objects and their relevant attributes. The map and table displays are shown in Figure 5.

In this example, the CUBRICON system distinguishes between spoken and written (to a CRT display) NL. CUBRICON used graphic and deictic gestures with spoken NL only (not with written NL), since a pointing or graphic gesture needs to be temporally synchronized with the corresponding verbal phrase, allowing for multiple graphic gestures within any individual sentence. The coordination between a graphic gesture and its co-referring verbal phrase is lost if printed text is used instead of speech. Written NL was used however, when deictic/graphic expressions are not used, but, instead, definite descriptions are generated as noun phrased with sufficient specificity to hopefully avoid ambiguous references.

The user now asks the system a question phrased exactly like the previous question for purposes of comparison.

USER: "Where is the Stargard airbase?"

DEVICE CONFIGURATION:

The color graphics display contains a map displaying the Fulda Gap region and a table showing the important attributes of the objects is displayed on the map. The monochrome display contains a mission planning form.

CUBRICON:

Monochrome Display:

- No change

Speech Output with coordinated Color Graphics:

- The sentence "Its location is 120 miles east of the Fulda Gap region." is accompanied by the visual point gesture which blinks the window containing the Fulda Gap region.

This example illustrates the flexibility CUBRICON has in selecting from alternative presentation modalities and its ability to measure the relevance. Although this question is phrased exactly the same as the previous question, the CUBRICON response is totally different. The Stargard Airbase is well outside of the user's area of responsibility as represented in the CUBRICON knowledge base (ie. the task model). Therefore, CUBRICON judges that the Stargard airbase is less relevant than the current display and does not modify the color graphics display to present the information graphically. Instead, the Natural Language Prose modality is chosen and the system responds verbally without changing the current display.

7 INTELLIGENT WINDOW MANAGER

The Intelligent Window Manager automatically performs all window placement and manipulation functions within the CUBRICON system. The decision to automate window management functions was based on the premise that this would reduce the user efforts required for window management, and thus free user mental and temporal resources for task domain activities. The goal was to automatically perform window management functions well enough so that the user would not need to manipulate the windows directly. The window management functions performed by the CUBRICON window manager include window: creation; placement; sizing; moving; and removal.

The fact that the time spent manipulating windows in a windowing system consumes a significant portion of overall problem solving time has been demonstrated experimentally [Davies85; Bly86], at least for certain types of tasks. Davies et al. found that for tasks requiring supplemental information relative to a primary task, the windowing environment allowed more error-free performance but took significantly longer. Their study indicates that the additional time spent, was due to window management operations (e.g., displaying and positioning windows, scrolling to desired locations within windows). Their data also indicates that the reduction in errors was not simply the result of having spent more time on the task. The time differential was evident even when all errors had to be corrected. Apparently, the overhead of window management adds a significant time burden.

Bly and Rosenberg [Bly86] studied the relative tradeoffs between overlapping and tiled windowing systems (see Section 7.2). They found that overlapping systems are good at optimally sizing windows (to contain the desired information), but are more difficult to manage. The CUBRICON window management methodology is a hybrid of tiled and overlapping approaches. The default configuration is tiled, but windows can overlap when necessary to avoid overly cluttered windows. Four pre-defined tiled window positions are available on each display. These overlap adjacent windows when necessary. If more than four windows are requested, the least important window is iconized and removed (i.e., removed and displayed displayed as an icon). Although redisplay of iconized windows is not implemented at this time, this feature will eventually allow the recall of windows that were recently removed.

7.1 WINDOW TYPES

The types of windows managed by the intelligent window management system are map, table, mission planning form, text and dynamic text windows. Figure 6 shows a color graphics display containing map, table, text and dynamic text window types.

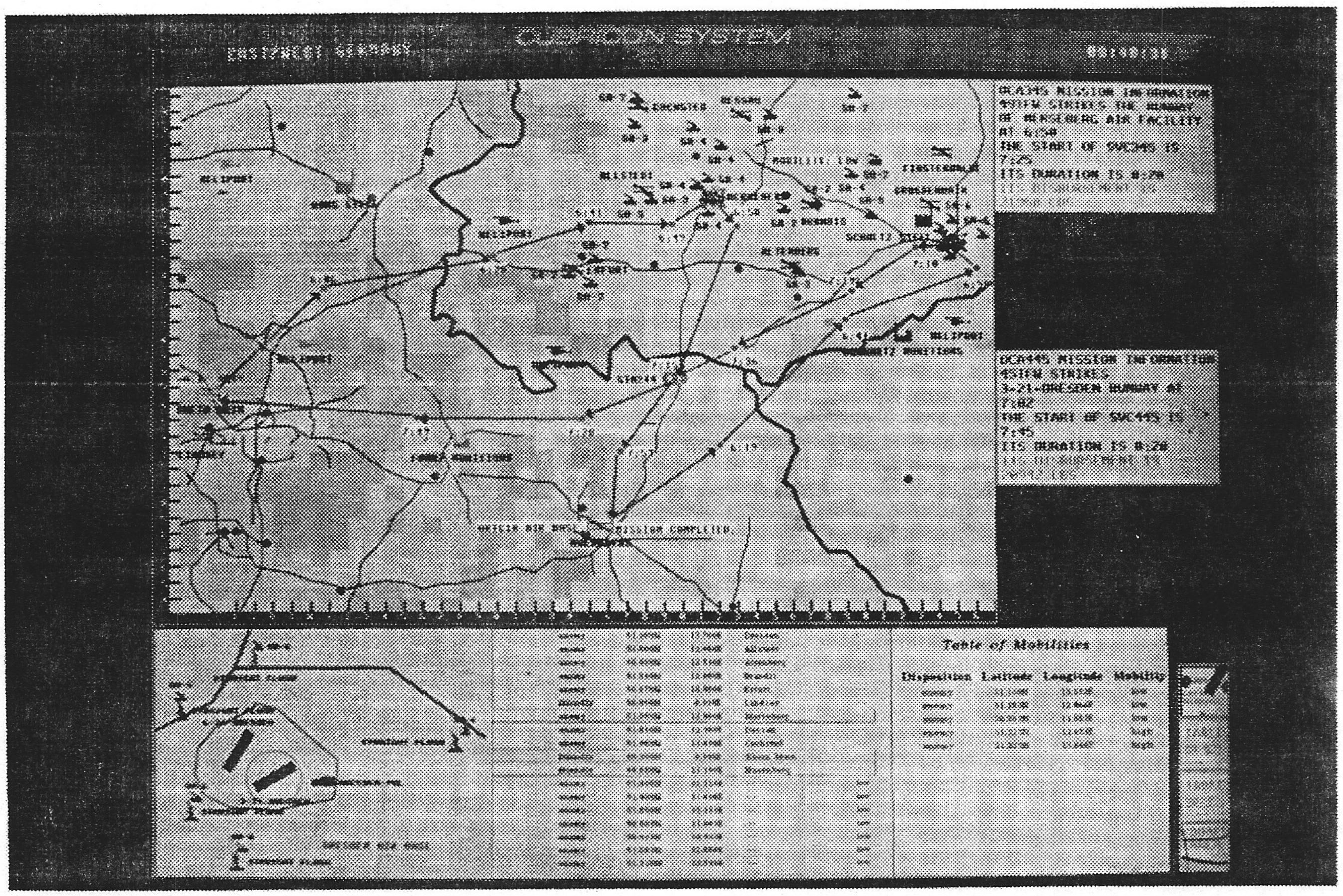


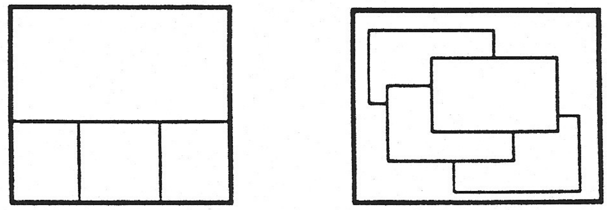
Figure 6: Types of Windows

Map windows show geographic information. All of the window manager functions: creation; placement; sizing; removal and moving can be performed on map windows. Map windows are placed only on the color graphics display. There are two types of maps; geographic maps and part-whole decomposition maps. Geographic maps are composed of background maps with relevant application domain information overlaid. The background maps include objects such as national borders, roads, rivers and cities which are displayed using a map system called the MAP Display System[Hilton87]. CUBRICON uses these maps as background, displaying domain application icons on them. The part-whole decomposition map window displays objects which are parts of an entity in the knowledge base. For example, several radars, SAM systems and runways are defined as parts of an airbase. The map displaying the parts of the Dresden airbase is shown in Figure 6. This map contains the boundary of the airbase and an icon representing each object which is a part of the airbase. The part-whole decomposition map does not utilize a background map.

Tables are used to display voluminous homogeneous information. The functions performed on table windows are creation, placement, sizing and removal. Tables can be temporary or permanent. Generally, if the information to be presented is on an existing table, the appropriate table entries are highlighted. However, if numerous table entries (more than four) are highlighted at various positions in the table, the user may find it difficult to compare information particularly if the information is not visible on one screen. Therefore, a temporary table is created which contains the information requested by the user, making it easy to view contiguously only the requested information. This table is referred to as temporary since it is visible for one user interaction. Permanent windows remain on the display until they are removed, due to space constraints. Permanent table windows can be placed either on the monochrome display or the color graphics display whereas temporary table windows are placed only on the monochrome display. In addition, permanent tables can be related to another display, such as a map. A table which is related to a map identifies the important attributes of the objects contained in the map. The window placement and importance algorithms are different for related and unrelated table windows as described in Section 7.3.

Text windows contain information pertaining to one or more icons on a map. Currently, the information contained in the window is text identifying the name or type of the icon(s). If the text window is not placed next to the icon, then a pointing arrow is used to correlate the appropriate icon(s) with the text window. Figure 6 shows a text window identifying the Nuernberg airbase as the origin of an OCA mission plan.

A dynamic text window is used to display natural language text on the color graphics display. It is dynamic in that text can be added to it. The window height is determined by the amount of text being presented. Figure 6 shows two dynamic text windows containing natural language text describing the highlights of an OCA mission plan presentation.



	TILED	OVERLAPPED
ABILITY TO OPTIMALLY SIZE WINDOWS	FAIR	☆ GOOD
NUMBER OF WINDOWS ON SCREEN	LOW	HIGH
WINDOW MANIPULATION REQUIRED OF THE USER	☆ LOW	HIGH
DISPLAY APPEARANCE	☆ ORGANIZED UNCLUTTERED	DISORGANIZED CLUTTERED
SUPPORTS MULTI-TASKING ON THE PART OF THE USER	☆ HIGH	LOW
CAN STANDARDIZE WINDOW LOCATIONS	☆ YES	TO A LIMITED EXTENT

Table 3: The pros and cons of the tiled and overlapping window layouts.

7.2 WINDOW LAYOUT

The window manager places a window in one of four pre-defined window positions on the display. Figure 7 shows window placement layout. Positions 1-4 are for the tiled windows. Positions 5-9 are for iconized windows. Iconized windows are windows that have been removed from the main part of display and redisplayed (symbolically) in the form of a small icon in the lower right hand corner.

The configuration of the widows is a hybrid window layout combining tiled and overlapping approaches. Table 3 shows the pros and cons of tiled and overlapping window layouts. This table lists various characteristics of window layouts and identifies the type of window layout which is superior in each category with a star. The strengths of a tiled window layout is that it requires little window manipulation by the user [Bly 86], displays windows in an organized and uncluttered manner, supports multi-tasking on the part of the user and allows

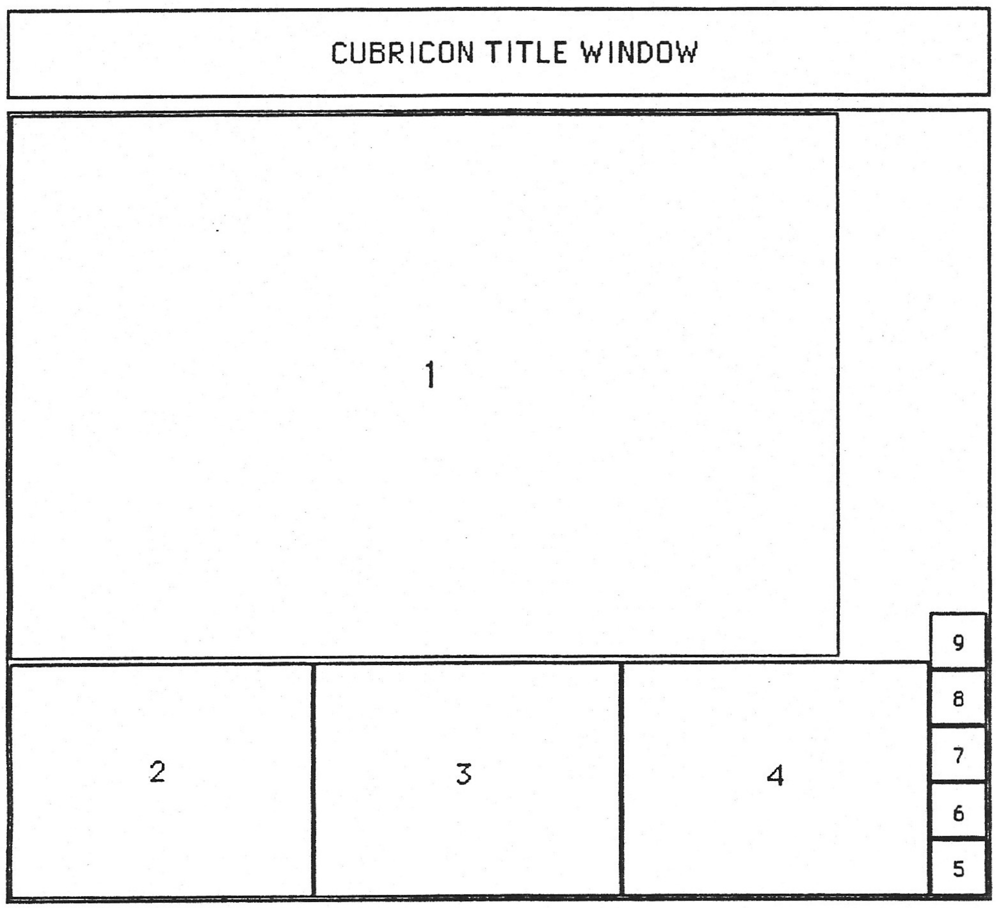


Figure 7: The Basic Tiled Window Layout.

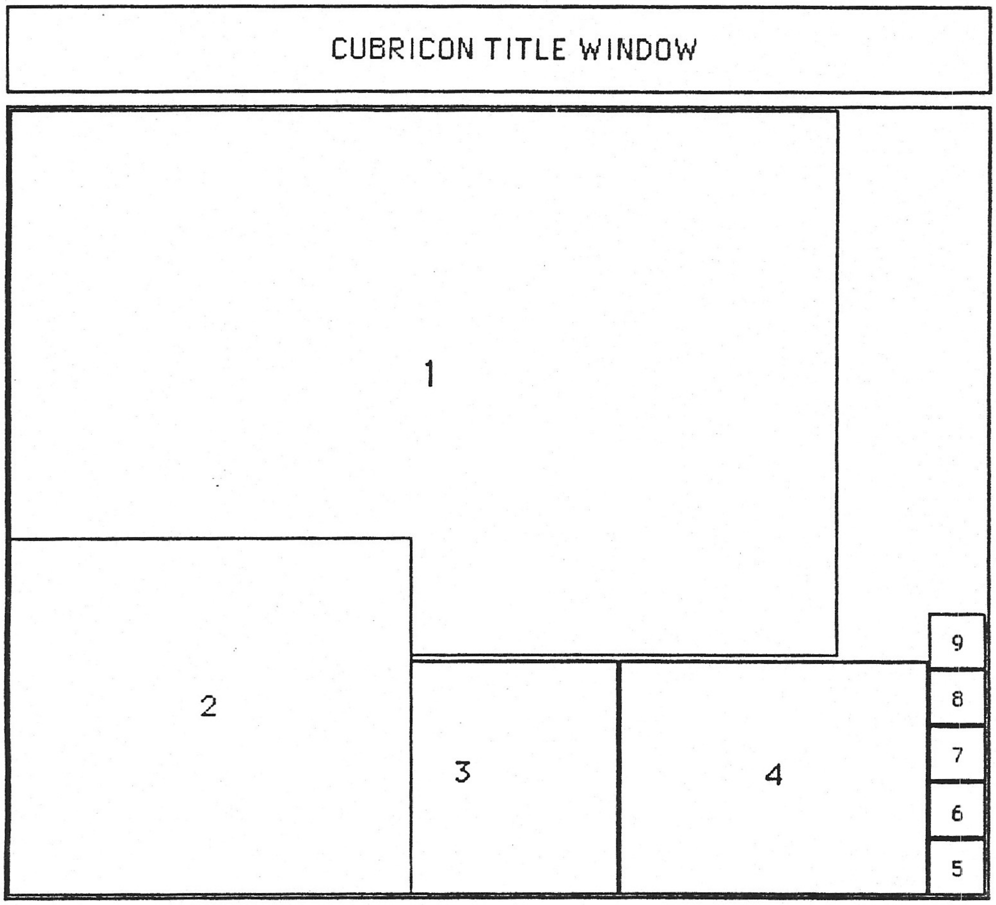


Figure 8: The Overlapping Window Layout. The window in position 2 is overlapping the windows in positions 1 and 3 because its contents could not fit otherwise.

standardized window locations. The strength of an overlapping window configuration is the ability of windows to conform to their contents, maximizing the visibility of these contents [Bly 86]. CUBRICON uses a tiled windowing approach as a default, but allows the "tiled" windows to overlap adjacent windows when necessary based on window contents. This allows CUBRICON to realize the advantages of both types of windowing systems. An overlapping window configuration is shown in Figure 8.

One type of window in CUBRICON often requiring an overlapping configuration is the map window. The size of a map window is determined by an algorithm described in Section 7.4. This algorithm computes the proper size of a map window based on a clutter factor which considers the density of icons and labels. Using this algorithm the size of a window may exceed the size of the pre-defined window position, requiring it to overlap adjacent windows. Another type of window often requiring an overlapping configuration is the table window.

If the columns of a table exceed the default width of the pre-defined window position, then the table overlaps horizontally.

7.3 INTELLIGENT WINDOW PLACEMANT

The two window types currently placed in one of the four pre-defined window positions are maps and tables. Maps are placed only on the color graphics display, while permanent table windows are placed either on the color graphics or monochrome display. The monochrome display is preferred for permanent table placement. However, if there is not an available window position on the monochrome display the table is placed on the color graphics display.

As previously stated, the placement of windows in one of the four pre-defined window positions is based primarily on the window positions available and the relative importance of the window being placed compared with the windows currently on the display. If there is a window position vacant, then the window will be placed in an available window position. Generally, window positions are filled in the following order; top, lower left, lower middle, then lower right. If a window would normally be put on the monochrome display, but all of the pre-defined window positions are filled and there is space for the window on the color graphics display, then the window manager places the window on the color graphics display. If a window is to be placed on the color graphics display, but all of the pre-defined window positions are filled, then the window manager removes the least important window to make space for the new window.

(more detailed description forthcoming)

7.4 SIZING MAP WINDOWS

One of the functions of the Window Manager is to decide the minimum acceptable size for a map window. The minimal size of a map is based on the density of display entities (e.g., icons, labels). This is important because when display entities are packed too closely together the map becomes difficult to read (i.e., the ease of extracting information from the map is reduced). For example, it becomes difficult to tell which labels go with which icons and some icons may overlap making them difficult to recognize. This reduction in readability can be measured as increased time or decreased accuracy for tasks using the map. These measures of reduced usability are said to indicate clutter [Potash77].

CUBRICON defines too much clutter as occuring when:

- A small number of display entities (e.g., three icons with labels) are crowded in a very

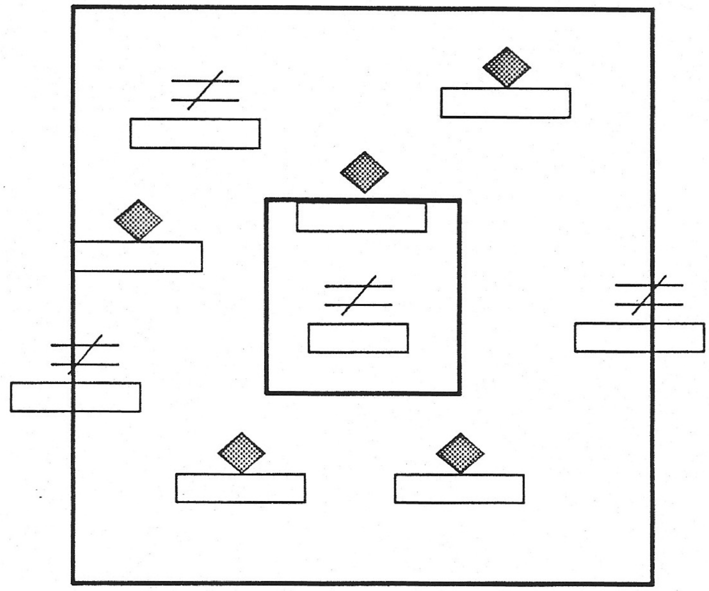


Figure 9: Two Critical Unit Areas. Only the larger area indicates a cluttered map.

small area.

- A medium number of display entities (e.g., five icons with labels) are crowded in a moderately sized area.
- A large number of display entities (e.g., seven or more icons with labels) are crowded in a large area.

The CUBRICON clutter (i.e., window sizing) algorithm defines "critical unit areas" based on these three conditions (see Figure 9). A criteria for saying that an area is too cluttered has been set for each of these three critical unit areas. The critical unit areas, and the clutter criteria for each, were empirically derived using the judgement of the human factors engineers assigned to the project. These values can be adjusted if this appears to be warranted based on experience in using CUBRICON, or as a result of research designed for this purpose. The evaluation conducted during this effort (see Section 15) indicates that the present values, and the algorithm is currently performing pretty well. what are they?

CUBRICON decides which critical unit areas to apply to a map display on a quadrant-by-quadrant basis. This is done based on "screen density" which is based on the number of display entities present in the quadrant. If there are only a few display entities in a map quadrant, for example, there is no need to apply the criteria (i.e., the critical unit areas) that relate to large numbers of display entities. If many display entities are present, then all critical unit areas must be applied. This is the approach CUBRICON uses to determine which critical unit areas to apply within each quadrant.

As noted above, this algorithm is applied on a quadrant-by-quadrant basis. This decision was based on the judgement of the human factors engineers on the project. It is intended to minimize the number of unnecessary calculations performed without sacrificing functionality. It reconizes the fact that icon density is not likely to be evenly distributed across an entire map display.

Once the critical unit areas to be applied for each quadrant are determined, they are applied on an icon-by-icon basis. That is, the degree of clutter in the immediate area (i.e., the critical unit area) around each icon is assessed. When the most crowded (i.e., cluttered) icon is found for each critical unit area, it is compared to the predetermined criteria. If the criteria are not exceeded for these worse-case icons, then the map is not too cluttered. If the criteria are exceeded, the map size (i.e., the window size) is increased to a size that reduces clutter factor for the worse-case icon to within the acceptable range. Since this resizing is based on worse-case icons, it ensures that all other icons will also be brought within the acceptable range. The amount of resizing needed is calculated directly from the clutter factor calculation.

7.4.1 The Map Sizing Algorithm

The first step in determining map size requirements is to determine whether the planned map is too cluttered. As described in the preceding section, this determination is made by analyzing clutter within each map quadrant. An appropriate criteria to be used in the analysis is selected for each quadrant based on the number of display entities contained within the quadrant. One or more "critical unit areas" may be applied. If a quadrant is found to be too cluttered, the entire map size is increased to a level where clutter is within acceptable levels. The following algorithm accomplishes this: redundant

Calculate four equal quadrants for the map
FOR each quadrant
    Calculate the screen density for this quadrant
    Table-Lookup the critical unit areas, if any, to use based on screen density
    FOR each critical unit area
        Initialize the maximum clutter factor to the acceptable clutter factor
        FOR each icon in the quadrant
            Calculate the clutter factor around this icon
            IF the clutter factor is the greatest thus far **THEN**
                Assign the maximum clutter factor for this critical unit area to this clutter factor

END IF (greatest clutter factor)
END FOR (each icon)
END FOR (each critical unit area)
END FOR (each quadrant)
Initialize the maximum percent increase to 1
FOR each critical unit area
Calculate the percentage to increase the map area by
IF the percentage is greater than the maximum percentage **THEN**
Assign the maximum percentage to this percentage
END IF (greatest percentage)
END FOR (each critical unit area)
Calculate the new map length and width

The map boundary to be used for clutter analysis is the boundary of window in which the new map will be placed. If a zoom out operation is being performed, then the new map replaces an existing map and the boundary of the window containing the existing map is used for clutter analysis. If a new map is being created, a new window is created and a default window boundary will be used for clutter analysis. The default window boundary is dependent on the position in which the map will be placed. Section 7.2, Window Layout, describes the available window positions in detail.

The screen density value approximates the overall clutter of a window. This value is used solely to determine the number of critical unit areas to be analyzed for clutter. Screen density is calculated as follows:

$$ScreenDensity = NumberOfIcons + .5NumberOfLabels$$

The variables NumberOfIcons and NumberOfLabels represents the total number of icons and the total number of labels on the map. In this calculation labels have less weight than icons. This is due to the difference in icon and label placement methodologies. CUBRICON utilizes an intelligent label placement algorithm which labels an icon only if space is available. Therefore, a label will not overlap an icon or another label. An icon, however, is placed at a particular location and may overlap another icon. Therefore, it is assumed that labels contribute to clutter half as much as icons contribute to clutter. Table 4 relates screen density values to their corresponding critical unit areas.

The clutter algorithm determines the actual clutter of the critical unit area. The formula for clutter follows:

Range for the Screen Density Value	Use the Following Critical Area Sizes
0 to 5	Not required
5.5 to 10	.25in.
10.5 to 15	.25in., .75in.
15.5 and higher	.25in., .75in., 1.25in.

Table 4: The selection of the critical unit area sizes for a quadrant is based on the screen density in that quadrant.

Critical Area Size	*Acceptable Clutter*
0.25	4.00
0.75	10.00
1.25	16.00

Table 5: The three critical unit area sizes and their respective acceptable clutter factor values.

$$Clutter = IconsInCA + 1.5LabelsInCA$$

The variable IconsInCA represents the number of icons in the critical unit area whose center is the location of a particular icon. Similarly the variable LabelsInCA represents the number of labels in the critical unit area relative to the same icon. An icon is considered within a critical unit area if the critical unit area square and the smallest rectangle enclosing an icon intersect. A label is considered within a critical unit area if the critical unit area square and any point on the label intersect. In this algorithm labels are weighted more than icons, since the size of a label is approximately 1.5 times the size of an icon.

The optimal map size is based on the maximum clutter value and the acceptable clutter value for each critical unit area. These values are used in calculating the percentage of increase to the pre-defined map area. The acceptable clutter values are defined in Table 5. This percentage is calculated as follows:

$$PercentIncrease = 1 + \frac{MaxClutter - AcceptableClutter}{AcceptableClutter}$$

The maximum value of PercentIncrease is used to calculate the final boundary values. The length and width of the map are increased as follows.

$$Length = Length\sqrt{PercentIncrease}$$
$$Width = Width\sqrt{PercentIncrease}$$

8 DEICTIC GESTURES

(section forthcoming)

9 TABLE MODALITY

(section forthcoming)

10 FORM MODALITY

(section forthcoming)

11 MAP MODALITY

The map modality is the modality preferred by CUBRICON and is selected whenever the information to be represented includes spacial relationships. This is accomplished by looking for regional, coordinate, or component references and the existence of iconic symbols associated with information to be presented. A detailed explanation of the criteria for selecting the map modality is in Section6.1.2, Map Modality Selection.

The map modality results in the creation or transformation of one of two types of maps; geographic maps or part-whole decomposition maps. A geographic map displays regional and coordinate information, whereas a part-whole decomposition map is a schematic diagram depicting the components of an object. These map types are described in detain in Section 7.1. Several operations are performed on geographic maps; map creation, zoom out, zoom in, and pan. However, part-whole decomposition maps are created and not modified afterwards.

11.1 GEOGRAPHIC MAPS

An important aspect of CUBRICON's processing capability is its decision-making logic for deciding when and how to create and transform geographic map displays. CUBRICON dynamically composes geographic map displays including the determination of the boundary of the region to be displayed and selection of the relevant entities to display in the region.

11.1.1 Map Composition

After selecting a map modality CUBRICON must decide when and how to compose map displays. The steps involved in composing a map display are:

1. Determine what type of map transformation to perform.

2. Determine the objects to display on a map.

3. Determine the boundary of the region to be displayed.

4. Extend the boundary of the region vertically or horizontally so that one vertical kilometer is displayed as the same distance as one horizontal kilometer.

5. Display the map in the appropriate window with the appropriate icons, colors, and labels.

Relevancy is a critical factor for an HCI to consider when determining what information to present and how to present it to the user. In CUBRICON's map manipulation process, relevancy plays an important role determining both what type of map transformation to perform and what objects to display on a map.

11.1.1.1 Determining the Map Transformation

One aspect of relevancy in the CUBRICON system pertains to the user's task. CUBRICON keeps track of the task that the user is working on and registers the user's transition from one task to another. CUBRICON's process for deciding on an appropriate map transformation takes into consideration whether or not the user's task has just changed. If the user's task has just changed then CUBRICON will move the map from the main window to a secondary window and "repaint" the main window with a new appropriate map area. If the user's task has not changed, however, then CUBRICON assumes that the current main map configuration is still relevant and tries to keep it in an existing window, subject to some expansion or contraction. When an existing map is being expanded to include additional objects requested by the user CUBRICON must decide which of these maps to modify to display the area of interest. The criteria used to determine which of the existing maps to transform are listed below in ranked order.

1. Expand the window containing the greatest number of objects requested by the user.
2. Expand the map which is closet to the smallest rectangle enclosing the objects requested by the user.
3. Expand the map with the greatest geographic area.

11.1.1.2 Determining the Relevant Objects

Relevancy is also important in selecting the objects to display in a map region. Frequently, sophisticated application systems include one or more massive databases and, indeed, the databases may be shared by more than one application system. When a system such as CUBRICON selects objects from the database for display on a map, it should be discriminating in its selection. Not all the available objects should be selected from the database for display, since this could result in an unnecessarily cluttered and confusing map. Instead, only the relevant objects should be displayed. Relevant objects are objects which are (1) specifically requested by the user, (2) relevant to the dialogue and support dialogue continuity and (3) relevant to the user's task.

Continuity and relevance are key factors in discourse. Without these factors, people find discourse disconcerting and unnatural. The attentional discourse space representation [Grosz78; Grosz86; Sidner83; Grosz85] is the key knowledge structure used in determining which objects to display in order to maintain dialogue continuity and relevance. The representation of the discourse focus space is in two structures (1) a main focus list and (2) a display model (discussed in Sections 3.2.1 and 3.2).

The technique used in the CUBRICON system to determine the objects relevant to the user's task relies on the use of the *entity rating system* of the user model (discussed in Section 3.3). When composing maps, CUBRICON displays only those objects above the critical importance threshold for the user's current task. Thus, for an Offensive Counter Air (OCA) planning task, CUBRICON would display all airbases, SAM sites, critical factories and plants, but not objects such as schools or minor industry.

11.1.1.3 Determining the Region to Display

After determining the appropriate map transformation and objects to display in a map region, based on relevancy, CUBRICON must delimit the boundary of the region to be displayed. The coordinates of the boundary are determined by the smallest rectangle enclosing both the objects to be displayed and the existing map to be expanded, if expansion of an existing map is relevant. This boundary is then enlarged to include a small border area.

what size?

11.1.1.4 Scaling the Region

One of the human factors guidelines incorporated into CUBRICON is to maintain consistency across displays [Smith 86]. An object presented on more than one display should have the same shape. To accomplish this a geographic map must display one vertical kilometer as the same distance as one horizontal kilometer. Therefore, the the boundary of the region is extended vertically or horizontally (if necessary), so that when the region is displayed in the window provided by the Intelligent Window Management system, a vertical kilometer and a horizontal kilometer are the same distance.

11.1.1.5 Displaying the Map

Having determined the type of map transformation, the objects to display on the map, the appropriate map boundary and properly scaled the map, the map is displayed in the appropriate window with the appropriate icons, colors, labels, etc. The composition and presentation of the map in the appropriate window is performed by the Color Graphics system.

11.1.2 Map Operations

The operations on geographic maps are listed below. In general, CUBRICON's decision-making process has been designed with the goal of maintaining context for the user and helping the user understand the transition from one map to another. For each of the map transformations, CUBRICON presents the new map in the context of the previously displayed map. In communicating the map transitions, CUBRICON uses a "region boundary box" to outline or highlight a region that is a sub-region of another. Objects to be displayed on each new map are selected according to their importance to the user's task, as discussed above.

- **Map Creation**

  A new map is created and displayed in a window on the color-graphics screen.

- **Zoom Out**

  The area shown in a map window is extended to include appropriate additional area of interest to the user. The criteria used to determine which map to extended was described in the previous section. A "region boundary box" is superimposed on the new map to show the boundary of the map that was previously displayed. This helps the user understand the transformation from previous map to new map display. Figure 10 shows the CUBRICON color-graphics screen after a zoom out operation has been performed.

- **Major Zoom In**

  A sub-region (specified by either the user or the system) of the current main map is enlarged. The map transition is performed as follows: CUBRICON first moves the map currently displayed in the main window to a secondary window and adds a "region boundary box" to this secondary window showing the sub-region that is to be enlarged. The enlarged version of the sub-region is then displayed in the main window. Figure 11 shows the CUBRICON color-graphics screen after a major zoom in operation has been performed.

- **Minor Zoom In**

  A sub-region (specified by either the user or the system) of the current main map is enlarged. The map transition is performed as follows: CUBRICON superimposes

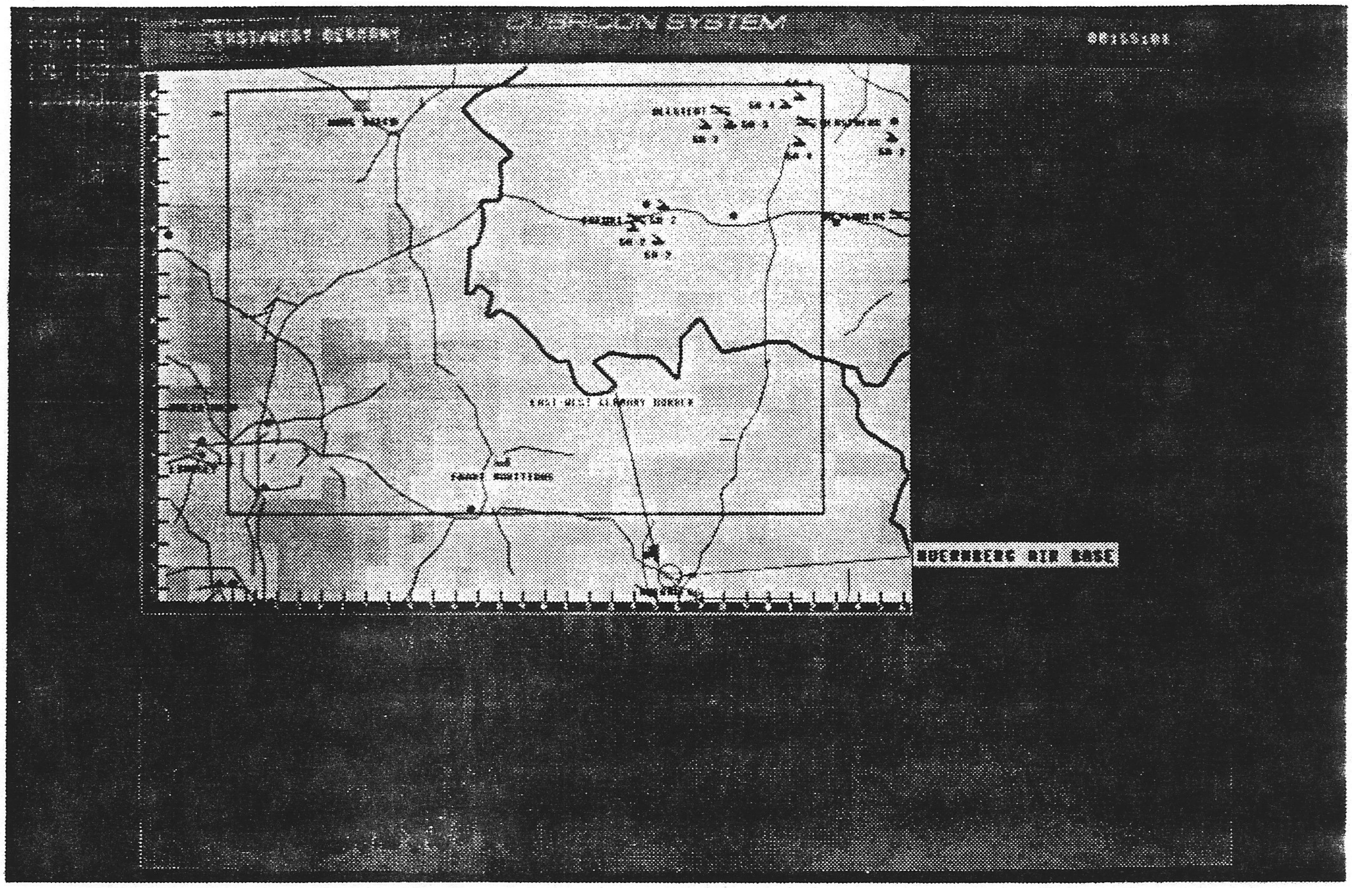


Figure 10: Color-Graphics Screen After Zoom Out Operation

a "region boundary box" on the main map showing the outline of the region to be enlarged. An enlarged version of the designated region is then displayed in a secondary window of appropriate size. Figure 12 shows the CUBRICON color-graphics screen after a minor zoom in operation has been performed.

- **Pan**

  Pan to a new region. The map transition is performed as follows: in one of the secondary windows, CUBRICON displays a map region whose boundary is the smallest rectangle enclosing the old map in the main window and the new region to be displayed; CUBRICON then shows region boundary boxes designating (1) the old region that was in the main window and (2) the new region to be displayed. The new region is displayed in the main map window. Figure 13 shows the CUBRICON color-graphics screen after a pan operation has been performed.

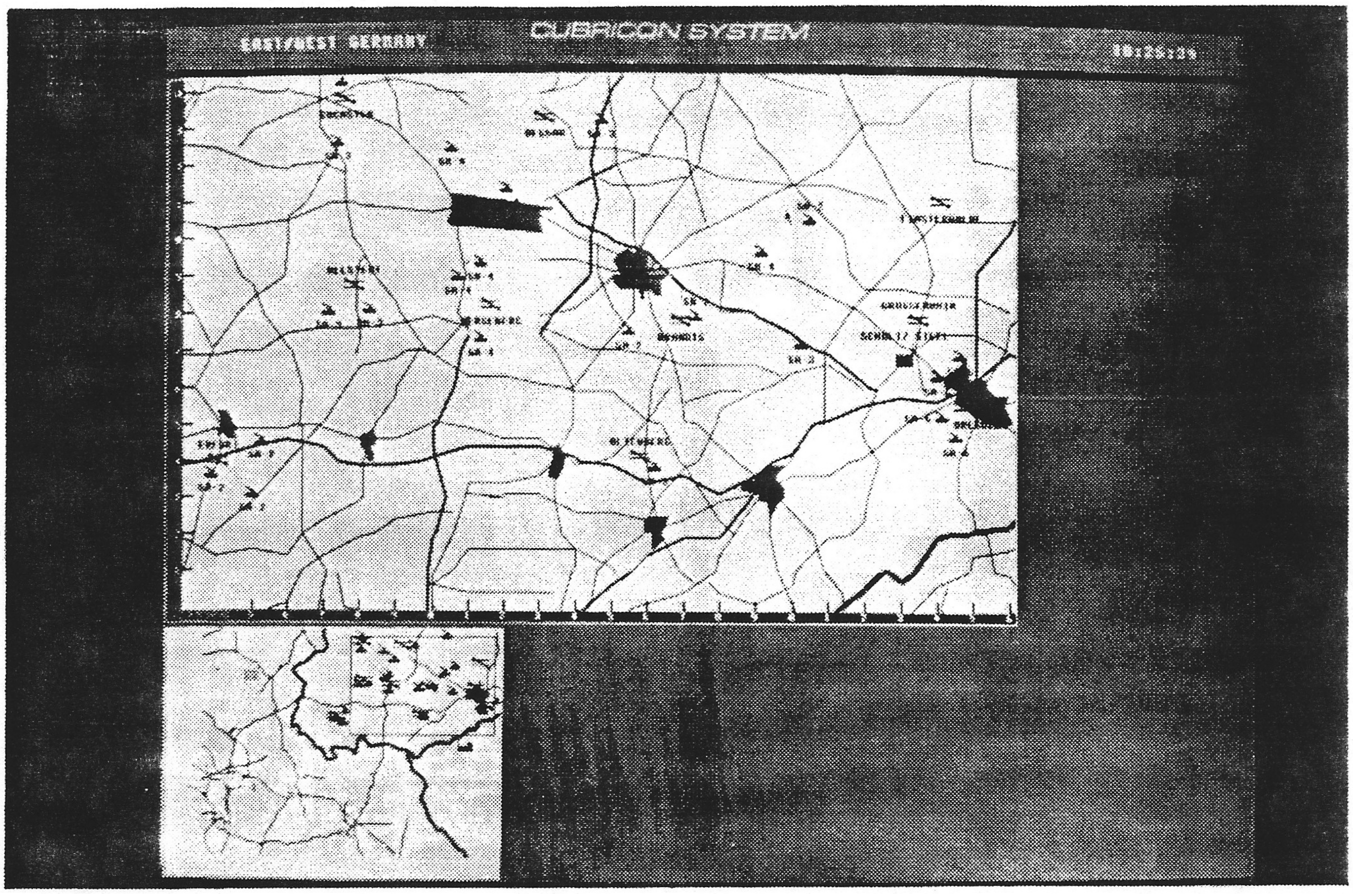


Figure 11: Color-Graphics Screen After Major Zoom In Operation

11.2 PART-WHOLE DECOMPOSITION MAPS

The map modlaity is the preferred modality used to represent the parts of an object, such as an airbase. An airbase's parts might be objects such as runways, radars and sams. The type of map generated is a part-whole decomposition map (see Section 7.1) which is a schematic diagram depicting the parts of the object and their relative locations.

In selecting the objects to display on a part-whole decomposition map, all parts of an object are relevant in the current CUBRICON system. When representing parts of an object the existing knowledge base contains only those objects which highly relevant to the OCA planning tasks defined. Therefore, the determination of the type of map transformation and objects to display in the map do not apply to part-whole decomposition maps. As one or more large databases are accessed by CUBRICON there will be a need to discriminate among the parts, displaying the objects which are most relevant. The descision-making logic used to select the relevant information to display on geographic maps, described above, will be used to select relevant information to display on part-whole decomposition maps.

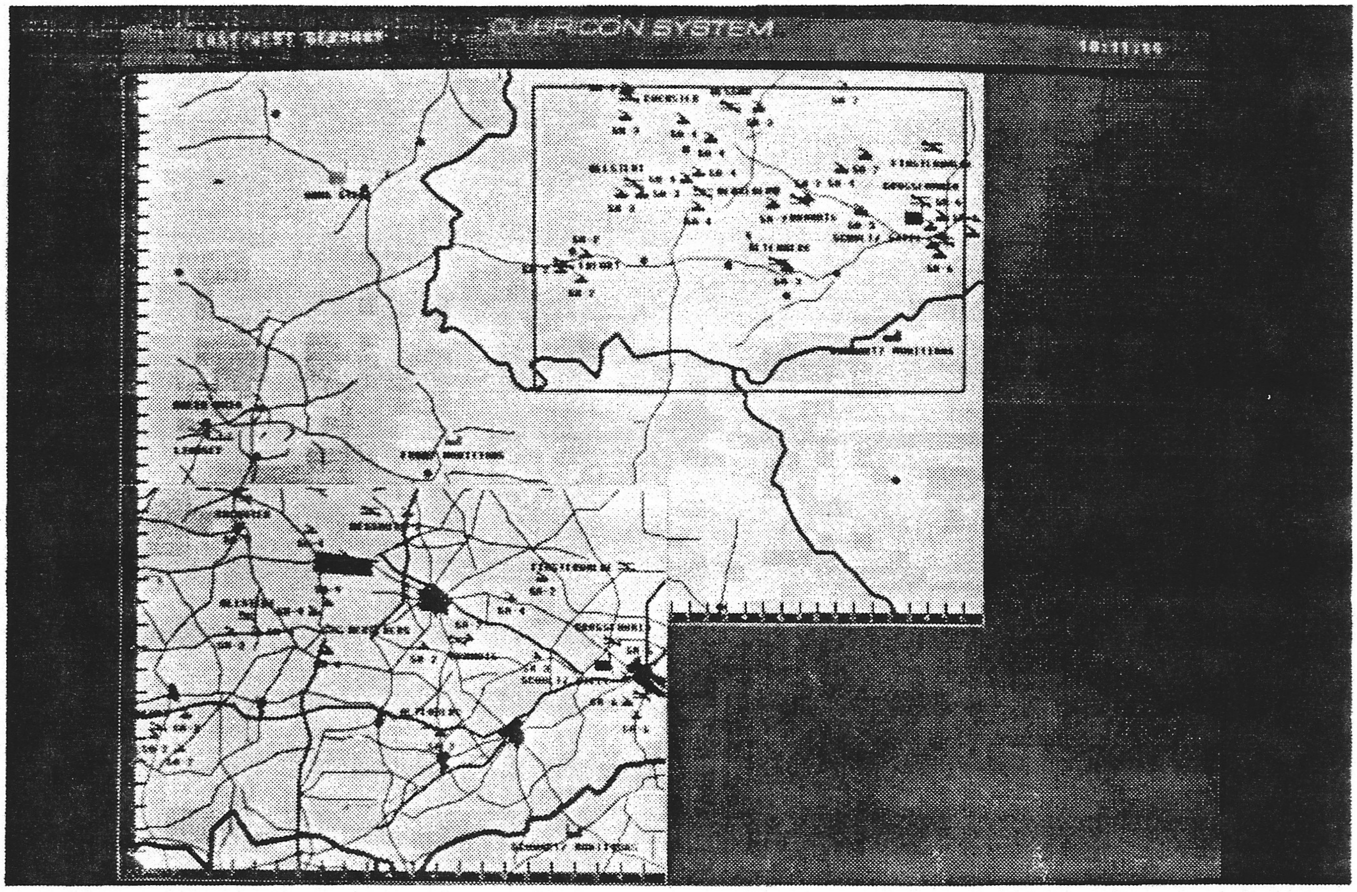


Figure 12: Color-Graphics Screen After Minor Zoom In Operation

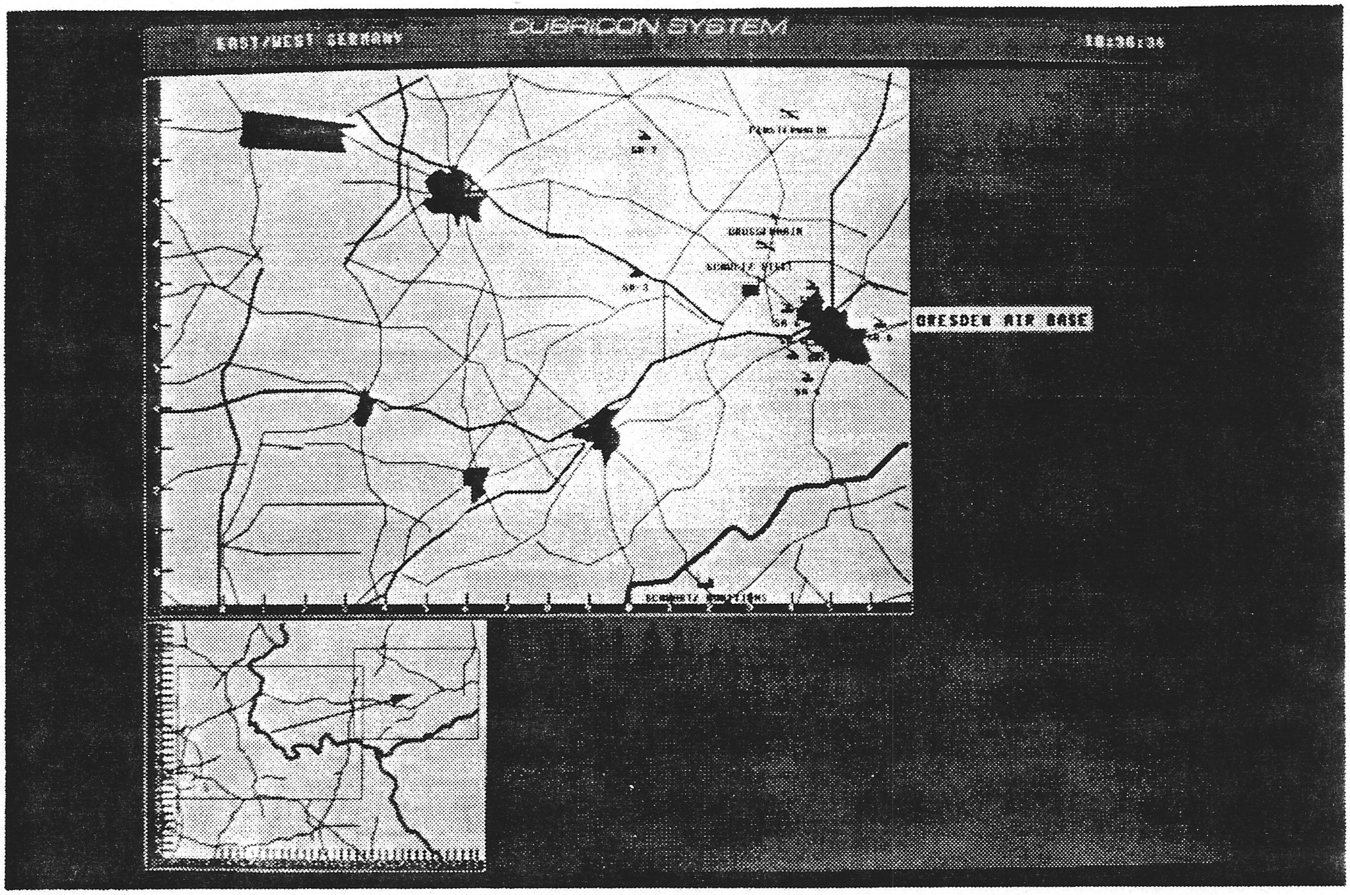


Figure 13: Color-Graphics Screen After Pan Operation

12 MULTI-MODAL LANGUAGE GENERATION

Just as the CUBRICON system allows users to express themselves in a multi-modal way, it will use a similar language in its responses. The system has essentially the same modalities at its disposal as the user has: spoken and written language, and pointing and other gestures. Multi-modal language generation refers to the system's ability to generate output using all available modalities in a consistent and coordinated way. The different modalities are not seen as separate, but as part of a single multi-modal output stream. This section discusses the use of the available output devices from the multi-modal point of view. We first take a look at the output modalities separately and then discuss their integrated use.

12.1 AVAILABLE OUTPUT MODALITIES

There are four different output modalities available for multi-modal language generation: spoken natural language as generated by a separate speech synthesis device, written natural language appearing on the monochrome and/or the color screen, and deictic gestures and graphic expressions appearing on the monochrome and/or the color screen.

12.1.1 Speech

Speech output is produced by a stand-alone DECtalk speech synthesis device, controlled through a serial port on the lisp machine. Since the DECtalk system includes a full grapheme-to-phoneme conversion system for English it accepts normal orthography almost without modifications. The only exceptions are a number of domain-specific words, abbreviations and acronyms and some other words that otherwise would be mispronounced mispronunciations (like the lexical stress in "subMISsion" in stead of "SUBmission") that need a special phonemic dictionary entry. The exception dictionary is stored in the synthesis device itself and is uploaded at system initialization time. We use DECtalk's standard male voice for all speech output.

12.1.2 Written Language

Written natural language output is typed mainly to the Natural Language Interaction Window on the monochrome screen. This is the same window that is used to echo the user's multi-modal input to the system. In addition, it can appear on some color windows.

12.1.3 Deictic Gestures

There are a number of deictic gestures available to the system (Section 8), dependent on the type of object being pointed to and the modality the object is represented in. On a map window, the system can highlight individual icons by drawing a circle around them or flashing them. They may also be provided with a descriptive label. An icons can be pointed to by drawing a textbox and an arrow pointing to it. Regions of a displayed area can be pointed to by drawing a box around them or flashing the window border containing them. Pointing to items on a table is done by drawing a box around the line(s) containing them. Pointing to items on the form is accomplished by flashing the box around the appropriate field and putting its content in boldface type. More information on deictic gestures employed by the system can be found in Section 8).

12.1.4 Graphic Expressions

The system can also produce graphic expressions that appear on the color graphics display. There are two implemented instances: locative information (see Appendix G) and path traversal information.

Locative information can be expressed as a visual representation of the spatial relation between two objects visible as icons on a map window. One object is considered the figure, the other the ground [Herskovits85]. An arrow is drawn from the ground icon to the figure icon, and labels are added to state the real-world distance between them and their identity, if appropriate (Section 12.2).

Path traversal information can be expressed in a graphical way by drawing consecutive segments of a path, represented as directed line segments on a map window, in the order of traversal. This presentation is accompanied by labels indicating the type or name of objects encountered along the way, and ancillary information such as an estimate of the time since departure, special icons representing actions to be taken, etc. There are provisions for presenting multiple paths in a synchronized way.

12.2 COORDINATION OF OUTPUT MODALITIES

The output modalities discussed above are controlled by a single multi-modal language generator which integrates and synchronizes them in real time. Just as multi-modality of the input stream allows for cross-modal disambiguation of otherwise ambiguous sentences (Section 4.2.2.4), so can typically terse spoken phrases like "this SAM" be disambiguated by a simultaneous deictic gesture to the intended referent (e.g. an icon on a map window or an element of a table).

12.2.1 Use of Written Natural Language

When written natural language is generated, deictic gestures are not used. This form of output is considered to be more permanent in nature than the transient spoken variety, and must therefore be more self-contained. Since the Natural Language Interaction window has a history mechanism, previous written natural language output can always be retrieved for reference purposes, and in this context there are no disambiguating deictic gestures available. Written natural language thus uses definite descriptions for noun phrases and locative phrases, so the intended referent can be determined from the language alone. In stead of saying "this SAM" and pointing to it, the system will write e.g. "the SA-2", which is the most specific description of the object it can come up with.

12.2.2 Use of Spoken Natural Language, Deictic Gestures and Graphic Expressions

Deictic gestures are combined with appropriate natural language during output to guide the user's visual focus of attention. During language generation, in order to compose a reference for an object,

1. if the object is represented by an icon on the display, then the system generates a natural language expression for the object and a simultaneous coordinated graphic gesture that points to the icon.

   If the object has an individual name or identifier, then the system uses its name or identifier (e.g., "the Merseberg airbase") as the natural language expression

   else the system generates an expression consisting of a demonstrative pronoun followed by the name of an appropriate class to which the object belongs (e.g., "this SAM", "these SAMs") as the natural language expression.

2. if the object (call it X) is not represented by an icon on the display, but is a component of such a visible object (call it Y), then the system generates a phrase that expresses object X as a component of object Y and uses a combined deictic-verbal expression for object Y as described in the above case. For example, if the system is generating a reference for the runway of an air base called Merseberg and an icon for the air base is visible on the map (the air base as a whole is represented visibly, but not its parts),

then the system generates the phrase "the runway of the Merseberg Airbase" with a simultaneous point gesture that is directed at the Merseberg air base icon on the map.

It is frequently the case that an object to which the system wants to point has a visible representation in more than one window on the CRTs. Therefore the system must select the visual representation(s) of the object (e.g., an icon, table entry, form slot entry) that it will use in its point gesture(s) from among the several candidates. The current system methodology is to point out all the object's visible representations, but to use a strong pointing gesture (e.g., blink the icon to attract the user's attention and add a pointing text-box) for the most significant or relevant representations and weak non-distracting gestures (e.g., just highlight the visible representation) for the less significant ones. In order to select the most relevant visible representations from among all the candidates, the system:

1. selects all the windows which contain a visible representation of the object.

2. filters out any windows which are not active or not exposed.

3. if there are exposed windows containing a visible representation of the object, then the system uses all of these representations as objects of weak deictic gestures and selects the visible representation in the most important or salient window (Section 3.2) as the target of a strong deictic gesture.

4. if there are no exposed windows displaying the object's visible representation, then the system determines the most important active de-exposed window (Section 3.2) displaying the object. The system exposes this window and uses the representation of the object in this window in a strong deictic gesture.

The system combines *graphic expressions* with natural language output when the information to be expressed is, at least partially, amenable to graphic presentation.

When generating locative information about some object (the figure object [Herskovits85]), the system selects an appropriate landmark as the ground object [Herskovits85], determines a spatial relationship between the figure and ground object, and generates a multi-modal expression for the locative information including the spatial relationship. When selecting the ground object, the system selects a landmark such as a city, border, or region, that is within the current map display (i.e., does not require a map transformation). If possible, the system uses a landmark that is in focus by virtue of its having been already used recently as a ground object. The system's discourse model (Section 3.2) includes a representation of the attentional focus space of the dialogue, including a main focus list of entities and propositions that have been expressed by the system or by the user via multi-modal language. If a new landmark must be used as a ground object, then the system selects the landmark that is nearest the figure object. The system derives a spatial relation between the ground object and figure object that it represents in its knowledge base. This relation includes (1) the direction from the ground object to the figure object and (2) the distance, if the distance is greater than 0.04 times the window width. If the distance is less than 0.04 times the window width, then the figure object appears to be right next to the ground object. This criterion for deciding whether to include distance as part of the relation reflects the tendency for people to omit a distance measure when the distance is small relative to the geographic area under discussion and to say something like "just northeast of" instead of stating a distance explicitly.

As an illustrative example, the user may ask about the location of a particular object, such as the Fritz Steel plant. The system then uses the steel plant as the figure object, selects a ground object, and derives a spatial relation between ground object and figure object as discussed above. The multi-modal response is given below.

USER: "Where is the Fritz Steel plant?"

CUBRICON: "The Fritz Steel plant is located here <point>, 45 miles southwest of Dresden <graphic-expression>."

The <point> consists of a gesture that points out the Fritz Steel plant icon to the user via a gesture that uses a combination of blinking, highlighting, circling the icon and the attachment of a pointing label-box that identifies the icon. The <graphic-expression> is a visual representation of the spatial relation between the figure object (Fritz steel plant) and the ground object (Dresden city), consisting of an arrow drawn from the Dresden city icon to the steel plant icon, a label stating the distance, and a label identifying the city (the steel plant should already be labeled).

The second implemented type of graphic expression is used to present path traversal information, more specifically for the presentation of a mission flight path (see Section 4.3 about defining flight paths). This type of presentation is fairly lengthy and consists of multiple multi-modal sentences, summarized below.

1. as an introduction to the mission presentation, the OCA (Offensive Counter Air) mission number, its package number (a package is a set of related missions), the origin (departure) air base, and the OCA's submissions (strike and refueling missions) are summarized in speech and written language (on the Natural Language Interaction Window), accompanied by pointing gestures to the corresponding items on the mission form, which is on the monochrome display. A mission information window is

initialized on the color graphics display, next to the relevant map window. It will be used to summarize important information in a written form.

2. the origin air base is highlighted on the map window, accompanied by a label reading "origin airbase" and speech saying the same.

3. one by one, the segments making up the (polygonal) flight path are displayed on the map window as directed line segments, with a label indicating the time computed for each vertex.

4. when the target air base of the mission is reached, the corresponding icon is highlighted on the map window. Simultaneoulsy, pointing gestures are generated to the aimpoint within the target air base (e.g. the runway). These pointing gestures go to the form, aimpoints window (if any) and any table that shows the aimpoint. The target information is summarized in the mission information window and in speech.

5. the presentation continues with flight path segments as before.

6. when the refueling location is reached on the map window, information about the refueling mission is pointed to on the form, and summarized in the mission information window and in speech.

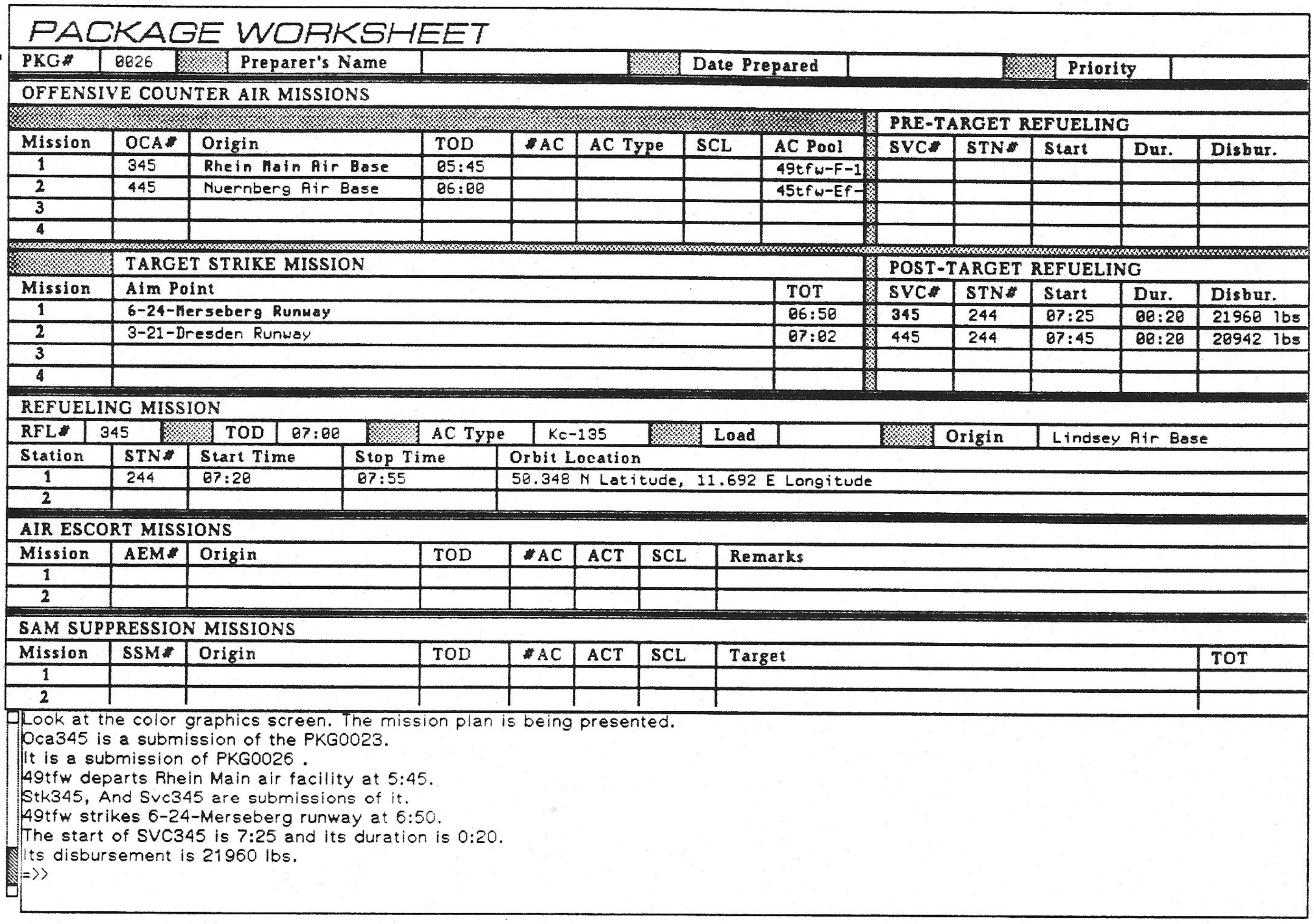
7. the presentation continues with flight path segments as before.

8. when the origin air base is reached again, which closes the polygonal path, the completion of the presentation is announced in speech.

The data that is written to the mission information window is also echoed in the Natural Language information window (figure 6). Two missions can be presented simultaneously. In that case the presentations are synchronized by the time determined for the vertices of the paths; the presentation reflects the real time relation between flight path segments and events for both. only two?

12.3 THE GENERATOR

The generator used to produce these multi-modal expressions is the one that comes standard with the SNePS knowledge representation system, as described in [Shapiro89], with a few minor adaptations to handle Multi-Modal sentences.



PACKAGE WORKSHEET

PKG#	0026	Preparer's Name		Date Prepared		Priority	

OFFENSIVE COUNTER AIR MISSIONS

Mission	OCA#	Origin	TOD	#AC	AC Type	SCL	AC Pool
1	345	**Rhein Main Air Base**	05:45				49tfw-F-1
2	445	Nuernberg Air Base	06:00				45tfw-Ef-
3							
4							

PRE-TARGET REFUELING

SVC#	STN#	Start	Dur.	Disbur.

TARGET STRIKE MISSION

Mission	Aim Point	TOT
1	**6-24-Merseberg Runway**	06:50
2	3-21-Dresden Runway	07:02
3		
4		

POST-TARGET REFUELING

SVC#	STN#	Start	Dur.	Disbur.
345	244	07:25	00:20	21960 lbs
445	244	07:45	00:20	20942 lbs

REFUELING MISSION

RFL#	345	TOD	07:00	AC Type	Kc-135	Load		Origin	Lindsey Air Base

Station	STN#	Start Time	Stop Time	Orbit Location
1	244	07:20	07:55	50.348 N Latitude, 11.692 E Longitude
2				

AIR ESCORT MISSIONS

Mission	AEM#	Origin	TOD	#AC	ACT	SCL	Remarks
1							
2							

SAM SUPPRESSION MISSIONS

Mission	SSM#	Origin	TOD	#AC	ACT	SCL	Target	TOT
1								
2								

Look at the color graphics screen. The mission plan is being presented.
Oca345 is a submission of the PKG0023.
It is a submission of PKG0026 .
49tfw departs Rhein Main air facility at 5:45.
Stk345, And Svc345 are submissions of it.
49tfw strikes 6-24-Merseberg runway at 6:50.
The start of SVC345 is 7:25 and its duration is 0:20.
Its disbursement is 21960 lbs.
=>>

Figure 14: The data written to the mission information window on the color screen is echoed in the Natural Language Interaction Window on the monochrome screen. Also notice that the origin air base and target of the relevant mission are boldfaced on the form.

12.3.1 Input for Generation

The input to the generator consists of a knowledge base node or a list of such nodes, representing the (propositional, see [Shapiro87]) information that has to be described. Optionally, a specific start state and some register value initializations can be specified (see [Shapiro89]).

12.3.2 The Grammar.

The generation grammar is written as a GATN, see [Shapiro89] and [Shapiro82] for more details. A description of the grammar is provided in Appendix B.

12.3.3 The Lexicon.

The lexicon used in natural language output generation is the same one as used in natural language input parsing and interpretation, see Section 4.2.1.2.

13 COLOR GRAPHICS MODULE

(section forthcoming)

14 KNOWLEDGE BASE BUILDER TOOL

Relational Database Management Systems (RDBMS) have become widespread in recent years, and many large relational data base (RDB) systems have been built. These systems, however, usually do not provide advanced intelligent interfaces and their use requires training in a data base (DB) language. The interfaces usually provided with such RDB systems generally consist of computer forms with simple record search and updating capabilities. Recent advances in human computer interface (HCI) technology provide the potential to overcome such limitations and make RDBs more readily accessible to personnel without special training. However such advanced HCI systems are often built to work with KBs and not RDBMSs.

This section discusses a tool, the Knowledge Base Builder, that was constructed to support the integration of the RDB underlying the AMPS Mission Planning system, and the KB underlying an intelligent multi-media interface (IMMI) system, CUBRICON. It is a hybrid tool in the sense that it operates on a knowledge base *and* a database system.

The KB used by the CUBRICON system is implemented in a Semantic Network Processing System (SNePS). The construction of KB's using this system involves the use of an editor (ZMACS in our case) to write files which contain semantic network building statements written in the SNePS user language (SNePSUL). This is a labor intensive process and does not easily support changes to the KB.

Without the aid of a tool such as the KB Builder here the process of building a KB in SNePS that corresponds to the data in AMPS would be a large effort in itself. It involves working from listings of the AMPS tables. For each piece of data in the listings the person doing the translation would type the corresponding SNePSUL statements. This transformation of the AMPS data into SNePSUL statements is not straight forward and requires thought on the part of the translator. The translation process does not condense the amount of text required but increases it. This means that is the AMPS DB were to be manually translated in this way the resulting text file of SNePSUL statements would be much larger than the original listings of the tables of the AMPS DB.

The KB Builder supports the integration of the KB and the RDB by providing the ability to construct KBs which are *linked* with the AMPS RDB. The tool does this by providing four main capabilities, a RDB browsing capabilities, a link construction capability, semi-automatic KB generation capabilities, and an interactive KB editing capability.

These capabilities are provided by the tool through a direct manipulation interface (windows, icons, menus, and pointing). This type of interface was chosen because of its easy of use *and* its ease of construction. Ultimately tools of this sort may have IMMI interfaces such as the

CUBRICON system. However, while the development of such an interface would be a worth while research project, it would not be an appropriate part of the CUBRICON project.

The link construction capability enables the definition of links between CUBRICON concepts (implemented in SNePS) and data in the AMPS RDBMS. Once these links are defined they can be used to build concepts in the KB from RDB data. This is the KB generation capability. In addition to the linkage and KB generation capabilities an interactive KB editing capability and a RDB browsing capability have also been provided to support the generation of skeletal class structures and the specification of KB structures for which no data exists in the KB.

The process by which the capabilities of the KB Builder are used to develop a KB and link it to a RDB is as follows:

1. Build a skeletal class structure in the SNePS KB using the interactive editing capabilities of the tool.
2. Build KB structures called *links* and place them in the skeletal class structure. Using the tool this is done in a interactive semi-automatic manner or by using the KB editing capabilities.
3. Identify instances of the skeleton classes from data in the RDB and build KB nodes representing them.
4. Build KB structures associating RDB information with each instance node.
5. Build additional KB structures manually using the KB editing capabilities.

An understanding of the capabilities provided by the tool requires an understanding of the concepts used in RDBM's and the SNePS system. Additionally a specialized concept of a RDB-SNePS link must also be discussed. This concept forms the basis for understanding the linkage definition and KB generation capabilities of the KB Builder.

14.1 SYSTEM CONCEPTS

The concepts that are involved in the use of the KB Builder are presented in Figure 15. The SNePS based concepts are discussed in detail in Section 3.1 and the RDB concepts are described in the literature. In this section we will examine two concepts which are essential to an understanding of the KB Builder's capabilities, the link concept, and the case frame type concept. Later sections discussing the tools capabilities and displays will rely on an understanding of these concepts.

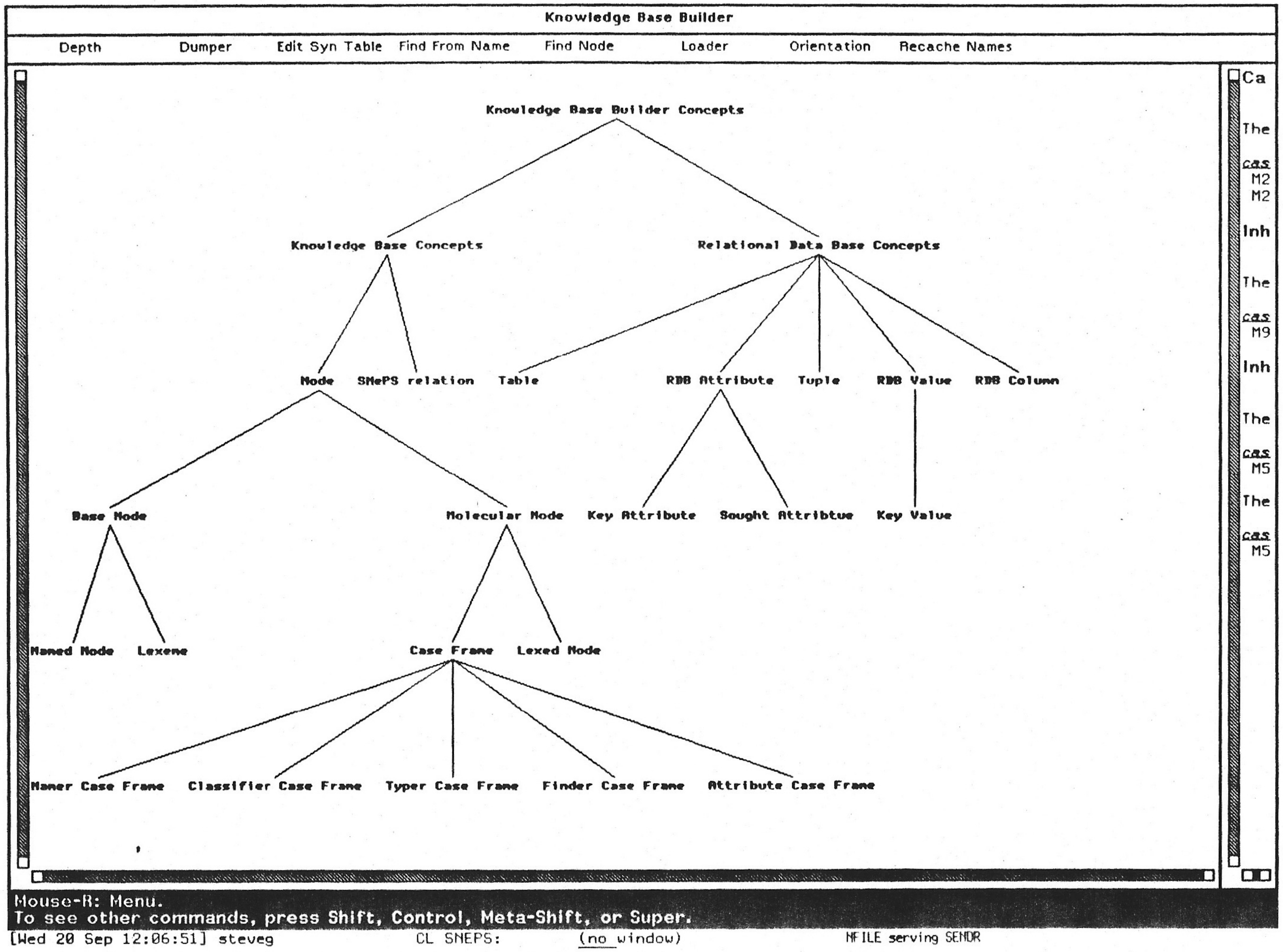


Figure 15: Heirarchy of Knowledge Base Builder Concepts

14.1.1 The Link

The link associates information from a RDB table with nodes in a semantic network that represent instances of a given class. The relationship between the instance and the node containing the linked information is accomplished through a case frame consisting of molecular node and a set of arcs (cf. 3.1). Figure 16 illustrates an example of such a linkage.

Please note that the figures of this section show abbreviated versions of the semantic networks. Nodes such as the Dresden node of Figure 16 do not really have the name "Dresden" but rather a node containing the name "Dresden" is related to the node representing dresden through a special case frame called the namer case frame. Such KB structures are provided so that the KB can distinguish between the concept of the dresden air base itself and the concept that the air base's name is Dresden. Section 3.1 provides many examples of the complete form of such networks. However, there is not enough room in our figures for all of the nodes and arcs that would result if all these naming structures were included. Such naming structures will be understood when a name in a node found in a figure is underlined.

The link itself is stored in the semantic network as a case frame (cf. Figure 17) which relates four pieces of information together.

- the name of a RDB table from which to extract information,
- the name of a RDB table attribute which is to be used as a key,
- the name of another attribute which is to be used to extract values from the table, and
- an identification of the manner in which the key values are to be generated.

The link is related to a class node in the network through another case frame such as in Figure 18. In this figure the relating case frame is represented by molecular node M2 which has three arcs descending from it, object, property, and value. The form of the case frame that links the class node to the node representing the link has a special purpose. The same case frame is used to relate the information found in the RDB to instances of the given class. This is illustrated in Figure 18 by the case frame represented by M3 which is of the same form as the case frame represented by node M2.

The use of links to generate KB structures from the contents of the RDB proceeds as follows.

- the RDB table is identified using information stored in the link.
- the name of a key attribute (stored in the link) is used with the name of the instance (Dresden in Figure 18) to identify the relevant tuples from the table.

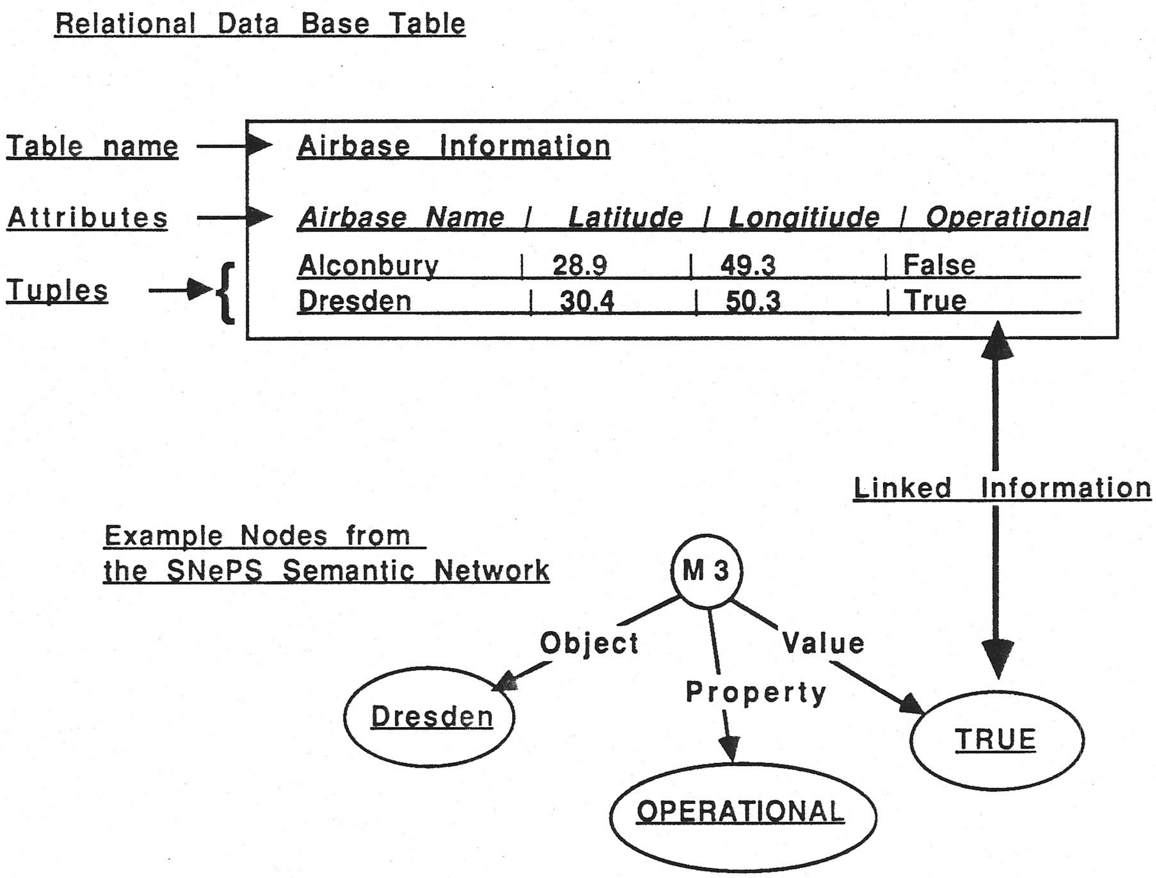


Figure 16: An Example Linkage

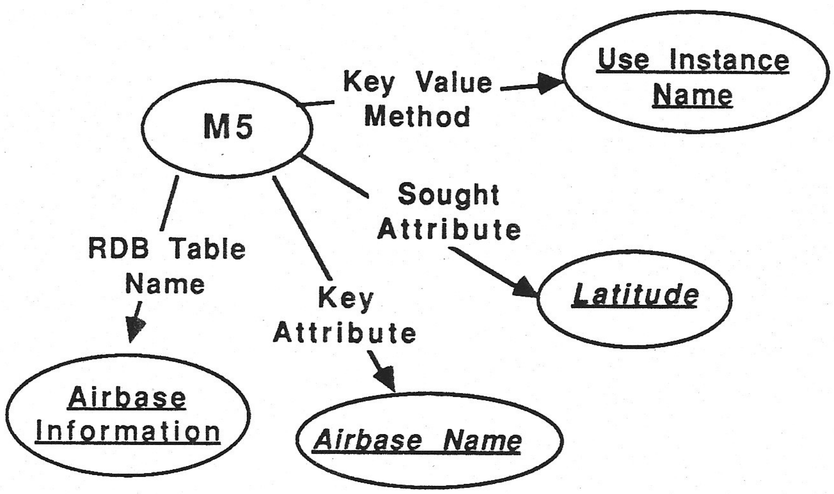


Figure 17: An Example Link

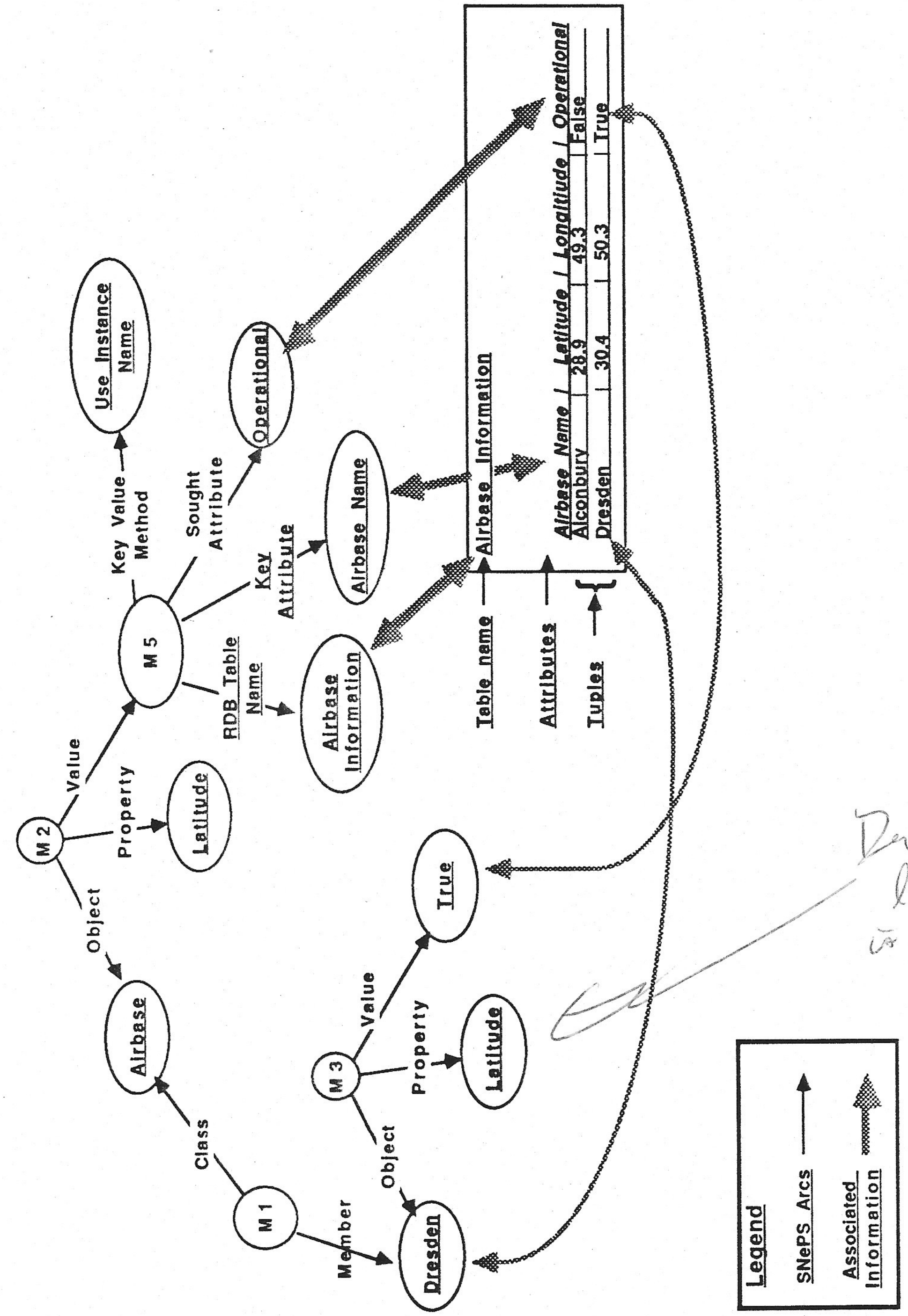
- the name of the sought attribute (Operational in the figure) is used to extract a set of values from the table.

- Case frames (such as the one represented by M3) are created which relate the extracted information to the node representing the instance. This case frame is patterned after the relationship between the class node and the link node.

14.1.2 The Case Frame and Case Frame Type concepts

The case frame concept as used by the CUBRICON and KB Builder system refers to a standard pattern of molecular nodes and arcs which relate a set of nodes together. Examples of such case frames are seen in Figures 19, and 20. For reasons of flexibility and SW development economy the tool's ability to manipulate case frames has been organized around the concept of abstract case frame types. This concept and the associated concepts such as related-nodes and the namer case frame are discussed in this subsection.

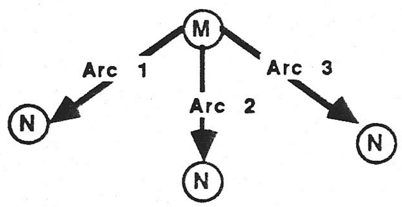
There are two basic requirements that lead to this approach. The first is a requirement to be able to dynamically specify new case frames and to modify existing ones without having to modify or extend the tool. The need for this became apparent during the course of the



Dresden's latitude is True ??

Figure 18: The Full Context of a Link

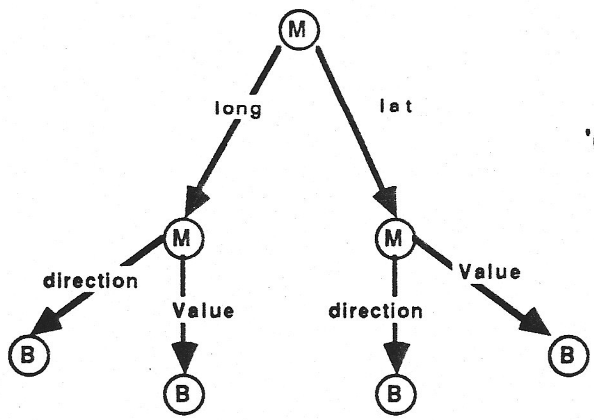
Simple Case Frame:



Defined by:

An Ordered list
of SNePS arcs.
'(arc-1 arc-2 arc-3)

Complex Case Frame:



Defined by:

An Ordered list
of path specifications.

'((compose long direction)
(compose long value)
(compose lat direction)
(compose lat value))

Figure 19: Simple and Complex Case Frames

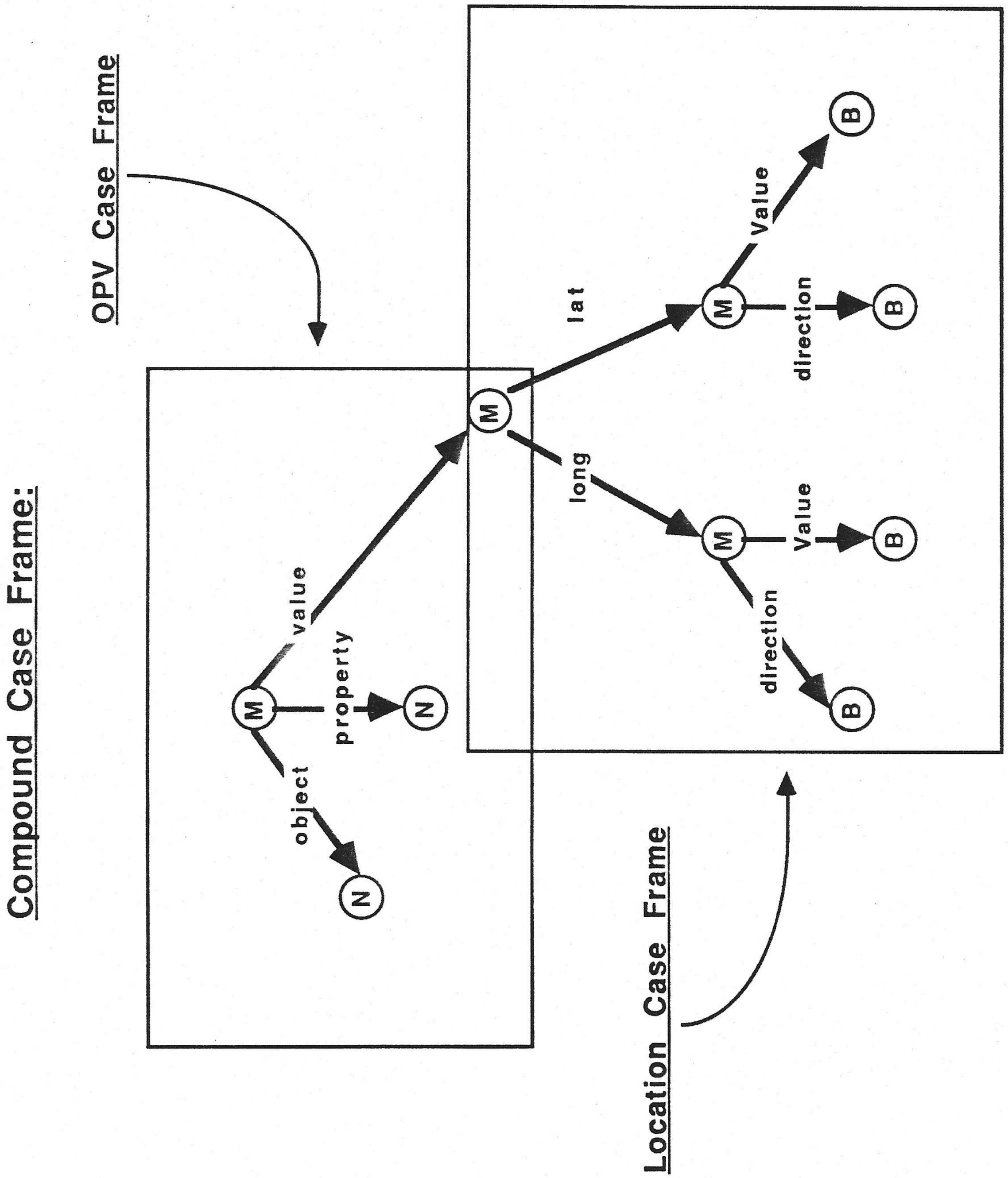


Figure 20: Compound Case Frames

development of the KB Builder tool as requirements to manipulate new case frames were incrementally added in conjunction with the evolution of the CUBRICON system. In the context of these requirements an examination of the functions that are required to manipulate the case frames was performed which revealed commonality that could be exploited. This led to the definition of a case frame type concept within the system which enabled the case frame manipulating functions to exploit the commonality. Structuring the tool to manipulate case frames of declaratively defined types has the following advantages:

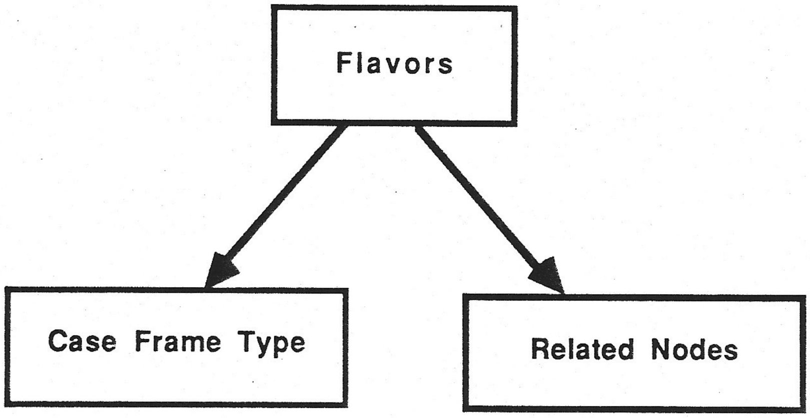
- The KB-builder tool software becomes independent of the choice of case frame representations that were chosen for the KB.
- The ability to examine and manipulate new case frames can be quickly added to the system.
- The organization of the software around abstract case frame type definitions is very compact. In previously investigated approaches each case frame type required its own file containing definitions of the case frame and functions to access, parse, display, edit, and create the case frame.

The approach chosen involved the development of presentation types and accessor/manipulator functions for the SNePS based concepts of a node, a base node, a molecular node, case frames in general, "a kind of"(AKO) nodes, and named nodes. It was not necessary to develop special code for each type of case frame used by the system (e.g. the location case frame, the OPV case frame, the PART case frame, ...).

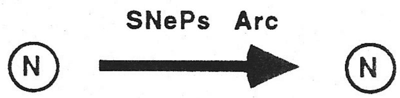
The requirements for a "WYSIWYG" editing capability led to a requirement to be able to easily "rewire" the arcs of the KB. This capability is supported by the rewire function and the **related-nodes** concept which identifies two nodes and an arc that relates them. This is exactly the information that is needed to replace a node in a case frame. The first node of a **related-nodes** instance is often a "non leaf" case frame node, and the second node is often a "leaf node" of the same case frame. When a case frame leaf node is presented on the screen (most of the information in the knowledge base is viewed in this way) a mouse action which can edit this part of the case frame must be able to refer to a related-nodes instance and not just the part itself. If this is not done then the information that the rewire-nodes function needs to replace the part is not available.

The two Symbolics flavors that have been defined to support the definition of case frame types and related nodes are presented in Figure 21.

The **case-frame-type** flavor collects all of the information the system needs to:



Related Nodes:



Case Frame Type:

Name of case frame

Ordered arc or Path list

Information on how to
input and present case frames
of the given type

Pattern which indicates if
lex arcs are expected on
certain of the case frames arcs

Figure 21: Flavors Based Concepts

- define the arc structure and order for the case frame type, and
- **scl::accept** and **scl::present** such case frames,

Note that while the arcs in SNePS case frames are unordered we have chosen to add an order to them for display purposes.

For each type of case frame known to the system an instance of the flavor **case-frame-type** is built and stored on the list ***case-frame-list***. The code which manipulates case frames utilizes these definitions. In this way the case frame related information is declarative and not procedural. Several of the case frames thus declared are given special designations. The special case frames include:

- the namer case frame,
- the super-class/sub-class case frame,
- the class-member case frames, and
- the link case frames.

The namer case frame is used to indicate the name associated with a base node. Such a base node is called a named node. By convention the first arc or path of the namer case frame type definition refers to the named node the second path refers to the node **name** and the third path refers to the name itself.

The super-class/sub-class case frame, and the class-member case frames, are used to define inheritance paths for finders and case frames. They also are used to display the AKO tree found in the left pane of the KB Builder screen. This ability to draw a tree of nodes related by case frames can easily be generalized. Currently the trees are only derived from the super-class/sub-class case frame, and the class-member case frame. In the generalized version the user would choose what case frames are to be used in the generation of the tree. This would allow the graphing of "a part of" trees, chain of authority trees, etc.

The Link case frame is used by the system to define the mappings between the RDBMS and the SNePS KB (cf. Section 15.1.1). Since the link is represented by a case frame embedded in another case frame no special purpose code is required to manipulate or edit these structures. A link is just another type of case frame on the ***case-frame-list*** list.

The system often has to manipulate "compound case frames" such as the case frame of Figure 17 or Figure 20. Such compound case frames may be nested several layers deep. The display software will handle this by recursively displaying the parts of the case frame.

For very deeply nested compound case frames the displays may be more extensive than desired. (This has not been the case in our use of the tool). If such deeply nested case frames are to be represented in the system a limit to the nesting level of such displays can be included. The remaining levels can be displayed in a pop up window via a mouse gesture upon user request.

14.2 KB BUILDER HUMAN COMPUTER INTERFACES

The four main capabilities of the tool are provided through two main interactive direct manipulation displays. One display, the Data Base Viewer, is oriented primarily toward the support of AMPS RDB browsing. The other display is orientated toward, the generation of linkages, the generation of KB structures from the RDB, and editing the KB. Figures 22 and 23 show the appearance of these displays.

Both displays have a menu of commands at the top and support *keyboard accelerators* for those same commands. The keyboard accelerators provide the means to invoke a command by entering the first letter of each command. This capability is in addition to pointing to the command in the command menu with the mouse.

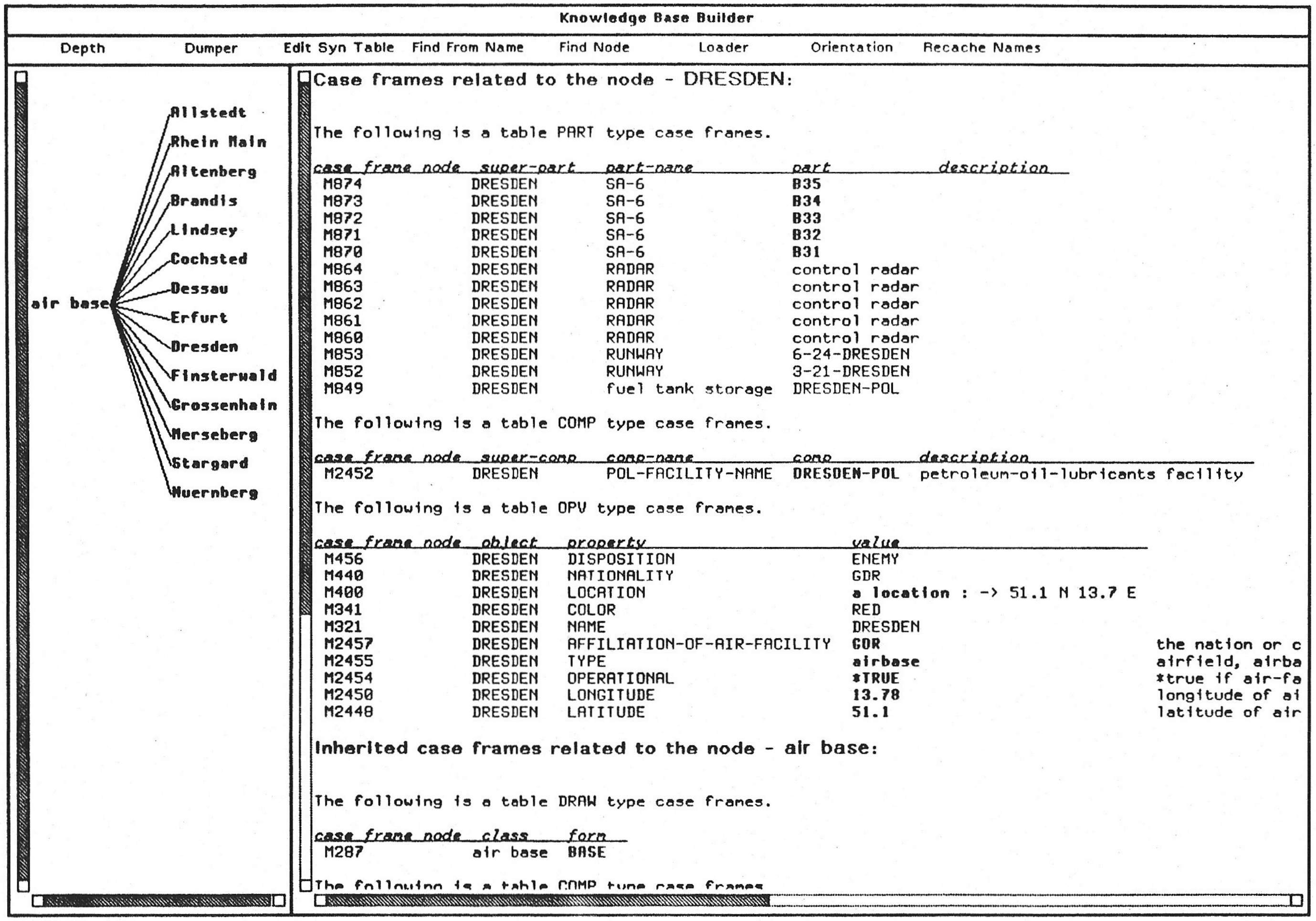
In addition to the commands presented on the top of the display every object presented in the display can be referred to with the mouse to invoke additional operations specific to that object. Nearly all of the supported activities are performed through such "point and click" references using the mouse. In all cases the operations that are available on the *mouse gestures* are show in the *mouse documentation line* at the bottom of the display. Since different commands are available when the mouse is on different types of objects or commands the mouse documentation line changes dynamically as the mouse moves from object to object.

In a few cases pop up menus appear which request the user to type in names or numbers. Whenever possible user typing is supported by computer completion of the typed input and computer enumeration of the remaining input possibilities.

14.3 SUPPORTED ACTIVITIES

The two main displays support a variety of activities, data base browsing, link generation, KB generation, and WYSIWYG editing. The details of how the tool supports these activities are discussed below.

Figure 22: The Data Base Viewer Display



Knowledge Base Builder

Depth | Dumper | Edit Syn Table | Find From Name | Find Node | Loader | Orientation | Recache Names

Case frames related to the node - DRESDEN:

The following is a table PART type case frames.

case frame node	super-part	part-name	part	description
M874	DRESDEN	SA-6	B35	
M873	DRESDEN	SA-6	B34	
M872	DRESDEN	SA-6	B33	
M871	DRESDEN	SA-6	B32	
M870	DRESDEN	SA-6	B31	
M864	DRESDEN	RADAR	control radar	
M863	DRESDEN	RADAR	control radar	
M862	DRESDEN	RADAR	control radar	
M861	DRESDEN	RADAR	control radar	
M860	DRESDEN	RADAR	control radar	
M853	DRESDEN	RUNWAY	6-24-DRESDEN	
M852	DRESDEN	RUNWAY	3-21-DRESDEN	
M849	DRESDEN	fuel tank storage	DRESDEN-POL	

The following is a table COMP type case frames.

case frame node	super-comp	comp-name	comp	description
M2452	DRESDEN	POL-FACILITY-NAME	DRESDEN-POL	petroleum-oil-lubricants facility

The following is a table OPV type case frames.

case frame node	object	property	value	
M456	DRESDEN	DISPOSITION	ENEMY	
M440	DRESDEN	NATIONALITY	GDR	
M400	DRESDEN	LOCATION	a location : -> 51.1 N 13.7 E	
M341	DRESDEN	COLOR	RED	
M321	DRESDEN	NAME	DRESDEN	
M2457	DRESDEN	AFFILIATION-OF-AIR-FACILITY	GDR	the nation or c
M2455	DRESDEN	TYPE	airbase	airfield, airba
M2454	DRESDEN	OPERATIONAL	*TRUE	*true if air-fa
M2450	DRESDEN	LONGITUDE	13.78	longitude of ai
M2448	DRESDEN	LATITUDE	51.1	latitude of air

Inherited case frames related to the node - air base:

The following is a table DRAW type case frames.

case frame node	class	form
M287	air base	BASE

The following is a table COMP type case frames

Figure 23: The KB Builder Display

14.3.1 Data Base Browsing Using the Data Base Viewer

The Data Base Viewer is intended to help the KB builder examine and become familar with the the AMPS database. It is easy to use and can display the entire contents of the AMPS RDB.

The data is displayed by table, the central concept in a RDBMS. The user can display the list of all tables by clicking on the **show tables** command in the command menu at the top of the display. This results in the display of all of the table names in the window pane underneath the command menu. Each table name is sensitive to mouse gestures and can be used to generate displays relating to that table. Normally the display of the table names is done just once and the table names are referred to through out the session. If new AMPS tables were created during a session then it would be necessary to redisplay the table names.

When the user points at a table name the mouse documentation line shows that, the table can be described, have its attributes listed, or be displayed.

> Choosing the description pops up a window with a small bit of english text describing the table.
>
> Choosing the attribute display causes the list of attributes for that table to be displayed in the lower window. (Descriptions of the attributes can be popped up via a mouse gesture where ever they appear.)
>
> Choosing the full display causes the entire contents of that table to be formatted and presented in the lower window of the display. The table is formatted in a tabular format with the table name preceeding the display and with the attribute list in the headings.

The lower box of the display contains both vertical and horizontal scroll bars. These are necessary since many of the tables take up more space than is available on the screen. The use of these scroll bar is interactively documented in the mouse documentation line. Every presentation made in the lower box is recorded and kept until the user invokes the **clear** command. Previous displays can be reviewed by simply scrolling the display forward or back to where the information was displayed. The mouse sensitivity of the screen display is continually preserved during scrolling operations.

14.3.2 The Generation of Links Between AMPS and SNePS

One of the primary uses of the KB Builder tool is to develop a mapping between the AMPS DB and CUBRICON's SNePS KB which is based on the definition of links (cf. Section 15.1.1). The KB Builder supports the interactive semi-automatic generation of these links based on information in the RDB. The tool performs a heuristic search of the RDB to identify information that might be used to build links. During the search the tool will interactively request information from the user to help it limit its search. When the link information has been collected the tool displays a list of candidate links for the user to select from. Selected links are then automatically added to the KB.

The semi-automatic definition of the links is done in the context of a given class node. The search proceeds in several stages,

- A list of RDB tables is found that may have information in them pertaining to members of the given class.
- These tables together with associated candidate key attributes are displayed and the user selects table/key-attribute combinations to be considered .
- For each table/key-attribute considered, candidate sought attributes are displayed and selected from.
- the user then selects the case frame type to be used to relate the link node with the class node.
- For each collection of link information identified a link is built and placed in the KB.

The tool uses the class node's name as the starting point for its heuristic search. This name is used to identify tables that may contain information relating to members of the given class node. This is done by searching the list of all tables in the RDB for those tables which have attributes whose name is "like" the name of the given class. This process identifies candidate table/attribute pairs. The attribute identified is a candidate for the key attribute to be used in the link.

An alternative approach would be to look for RDB tables with names that are "like" the given class name. This approach was tried and found to be inferior for two reasons. It does not identify key attributes which are required and it overlooked tables with pertinent information.

Figure 24 illustrates the selected approach. The given class node in this case is air base. An examination of the Figure reveals that the runway-char table contains an attribute called

Air-Facility-Char

Air-Facility-Name	Affiliation	POL Facility
Dresden	Germany	Dresden POL
Alconbury	UK	Alconbury POL

Runway-Char

Air-Facility	Bearing	Length
Dresden	30	500
Alconbury	20	400

SSM-Mission-Char

Mission-Number	Start Time	Origin
Dresden	13:00	Dresden
Alconbury	04:00	Alconbury

Figure 24: Attribute Based Table Selection

air-facility and the air-facility-char table contains the attribute air-facility-name. The tool would select these two tables because the names air-facility and air-facility-name are "like" the name air-base. The table SSM-mission-char is not selected since none of its attributes are "like" the name air-base. Note that the table runway-char would not have been selected if the alternative approach were used. The tables and key attributes that are selected by the tool are then displayed in a window (cf. Figure 25) so that the user can make a further selection from them.

In order for the tool to work it must be able to identify when one name is "like" another. This process is based on a synonym table which the user can interactively edit. This table might express the notion that base and facility are synonymous, or air-base could be associated with air-facility directly. In the process of finding like words the words are de-hyphenated into a list of component words. These component words are augmented by synonyms from the synonym table and all the resulting combinations are compared to see if they are contained in the attribute names of the tables.

Once the table/key-attribute pairs have been selected, the selected tables are used to generate a list of candidate sought attributes. These candidate sought attributes are presented in a display (cf. Figure 26) along with the associated table and key attribute. The user selects from this display the collections of table, key attribute, and sought attribute that are to be used to build a link. The type of case frame that will be used to relate the class node and the link is also selected through this display.

For each selected combination of table, key attribute, sought attribute, case frame type, a link is built and related to the given class node. It is assumed that instance names will be used as a key values.

14.3.3 KB Generation

The system enables the selective construction, in a controlled fashion, of large KB's based on information from the AMPS DB. This is an important capability since attempts to map the whole AMPS DB into SNePS resulted in KB's whose size taxed the capabilities of the computer systems used.

The system provides two basic KB generation capabilities, 1) the generation of instance nodes representing members of a given class, and 2) the generation of KB structures (i.e. case frames) which relate information to the class nodes.

Candidate AMPS RDB Tables, which may contain attribute information for air facility	Consider it.	Ignore it.
POL-FACILITY-CHAR petro/oil/lub facility;source-krs	☐	☐
AC-CAPS capabilities for each aircraft	☐	☐
AC-ROLE role for each aircraft	☐	☐
AC-TYPE types of aircraft, used primarily for refinement	☐	☐
AC-SCL what scls are carried by various aircraft, source = krs	☐	☐
AIRCRAFT-CHAR aircraft information, most from krs	☐	☐
AC-POOL-CHAR ac-pool info, source = krs	☐	☐
UNIT-AC aircraft at unit information	☐	☐
INTEL-AIM-POINTS Recommendations from INTEL for which targets and aim-points are worthwhile	☐	☐
AC-RESOURCE-UTILIZATION-CHAR resources to be used by missions	☐	☐
MISSION-CHAR mission info, eg. a/c & home-base info; one entry per mission-name; THIS IS NOT	☐	☐
MAINTENANCE-CHAR Maintenance-specific information, see also STANDARD-MISSION-TASKS-CHAR, AMPS-RES	☐	☐
UNIT-CHAR unit information. See AC-POOL-CHAR to find availability and AC type information	☐	☐
SAM-SITE-CHAR sam-site info, source=krs	☐	☐
RUNWAY-CHAR runway information, enemy in degree heading	☐	☐
RADAR-FACILITY-CHAR radar-facility information, source=krs	☐	☐
MUNITION-DUMP-CHAR how many of each munition are at each dump	☐	☐
AIR-FACILITY-CHAR info specific to air-facilities	☐	☐

Do It ☐ Abort ☐

Figure 25: Table and Key Attribute Selection

CANDIDATE ATTRIBUTES FOR air facility	yes	OPV	CHAR	COMP	PART
Top					
RUNWAY-NAME name of runway	☒	☐	☐	☒	☐
AFFILIATION-OF-RUNWAY a higher command	☐	☐	☐	☐	☐
LENGTH the length of the runway	☐	☐	☐	☐	☐
WIDTH the width of the runway	☐	☐	☐	☐	☐
SURFACE-CONDITION the reported condition of the runway	☐	☐	☐	☐	☐
CONSTRUCTION composition of runway surface,e.g. concrete, dirt, sand, grass..	☐	☐	☐	☐	☐
DISPOSITION friend, enemy, neutral, etc	☐	☐	☐	☐	☐
TYPE kind of facility	☐	☐	☐	☐	☐
DEGREES heading in degrees	☐	☐	☐	☐	☐
AIR-FACILITY-NAME name of air-facility (a facility that supports a/c)	☐	☐	☐	☐	☐
LATITUDE latitude of air-facility	☒	☒	☐	☐	☐
LONGITUDE longitude of air-facility	☒	☒	☐	☐	☐
POL-FACILITY-NAME petroleum-oil-lubricants facility	☒	☐	☐	☒	☐
OPERATIONAL *true if air-facility is operational	☒	☒	☐	☐	☐
DISPOSITION friend, enemy, neutral, etc	☒	☒	☐	☐	☐
TYPE airfield, airbase, etc	☒	☒	☐	☐	☐
AFFILIATION-OF-AIR-FACILITY the nation or command that controls the air-facility	☒	☒	☐	☐	☐
ESTABLISHED date first operational	☒	☐	☒	☐	☐
COMMO-FREQ frequency for communications	☒	☐	☒	☐	☐
COMMO-CALL-SIGN call-sign identifying the air-facility in communications	☒	☐	☒	☐	☐
More below					

Do It ☐ Abort ☐

Figure 26: Final Selection of Link Related Information

14.3.3.1 Knowledge Base Instance Generation

The user of the KB Builder tool first uses the tool's KB editing capabilities to build a skeletal class structure in the KB. Ultimately this skeletal structure must be populated with instances of the classes. For example, the node representing the class air base must be related to an instance node representing Dresden air base. The KB builder tool supports the semi-automatic generation of nodes representing such instances from the RDB information.

For a given class node the tool identifies the names of instances of that class using techniques like those used to identify link information. The tool first uses the name of the given class to find tables with attribute names "like" the name of the given class node. The table and attribute pairs are displayed and the user of the tool selects one such pair.

The attribute of the table/attribute pair selected by the user references a column of the selected table. This column contains the names that are to be used as instances for the given class. For example the air-facility-char table may have an attribute called air-facility name. In the column under this attribute are all of the names of airbases contained in the RDB. This set of names are used as the names of the instances of the air base class and are used by the tool to build instance nodes.

14.3.3.2 Case Frame Generation

Once the links and instance nodes have been created for a given class node, the links are used to build case frames that relate RDB information to the instances. The procedure for using a link to relate RDB information to an instance node was discussed in Section 2.1.

We note that if an instance has super classes in addition to its class then it may inherit links from the super classes as well as from the class. These links work in the same way as those associated with the class.

14.3.4 WYSIWYG Editing Capabilities

The Knowledge Base Builder provides "what you see is what you get" (WYSIWYG) editing capabilities through the use of the Knowledge Base Builder window shown in Figure 23. This display has two windows, the one on the left is the AKO tree display and the one on the right is the case frame information display. This later display presents information related to a given class or instance node through molecular nodes of known case frame types.

14.3.4.1 KB Displays

These displays hide many of the SNePS representation details from the viewer so that the information which is pertinent to the user's task is emphasized. As an example consider molecular node M400 in the display in the right window of Figure 23. This node is found in the table of OPV type case frames and gives Dresden's location. Figure 27 shows a diagram which includes the SnePS network corresponding to this line. The design of the SNePS representations used by the case frame types is a separate concern and would be handled by a different display.

When the KB Builder displays a node it attempts to present that node by printing,

- it's name if the node is a named base node.
- the name of the nodes lexeme if it is a lexed node.
- a standard case frame presentation if the node is a case frame of a known type.

Base nodes are given names in the KB through the use of the namer case frame. If a base node is connected to such a case frame then is is said to be a named node and the named is used as the printed representation of the node. If a base node is not named then the internal name of the node itself is used. An example of this is the node B35 in Figure 23. It represents an unnamed SA-6 which is part of Dresden.

Case frames very rarely refer to a name directly. Instead they often point to a molecular node which represents the concept of the name and that node uses a lex arc to point to a base node which contains the name. Such molecular nodes are referred to as lexed nodes and the corresponding base node as the lexeme. When such nodes are encountered the name contained in the lexeme is used as the printed representation of the node.

When compound case frames are encountered, such as is the case with node M400 of Figure 23, a standard representation is used for the subordinate case frame (the location case frame in this example). A printer function which is stored in the definition of the case frame type is used to print a representation of the subordinate case frame. This function may in turn recursively call other case frame printer functions to display an case frames that are subordinate to it.

The left window (cf. Figure 23) displays the skeletal class structure of the KB which is also called it's "A Kind Of" (AKO) hierarchy. This display also includes presentations of the instance node as they relate to their classes. In the figure the air base class node and its instances are shown. This display hides the details of how the relationship between the

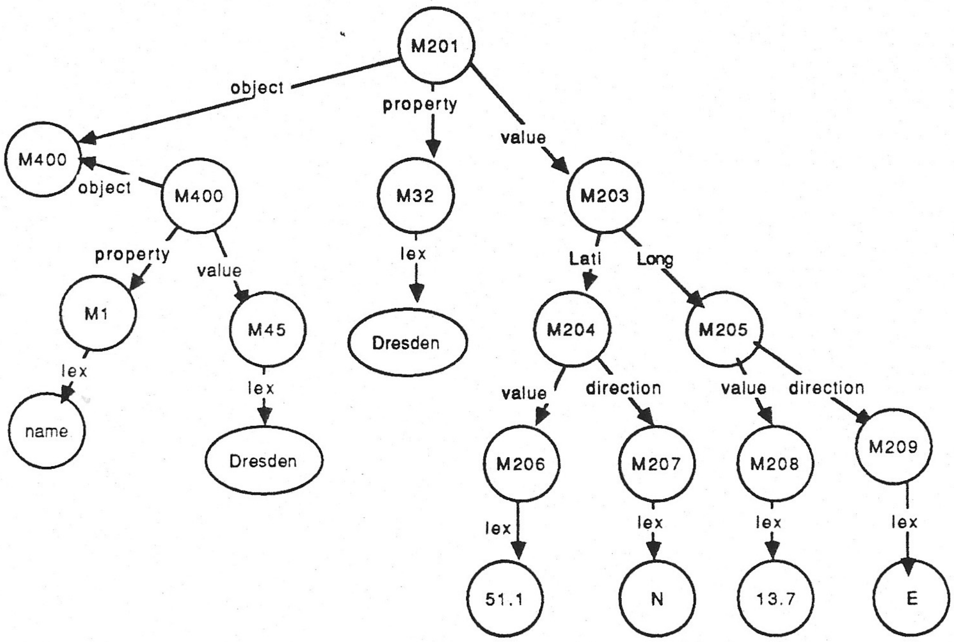


Figure 27: SNePS Structures Corresponding to KB Builder Display Items

instances and classes and the relationship between the classes and their super classes are implemented.

The user can controls the extent of the graph in this display by,

- selecting which class or instance node is to be the root of the display tree,
- the depth of the display tree,
- the orientation of the display tree (horizontal or vertical), and
- the size of the window that the AKO trees are displayed in.

For example the user can select a new display root by pointing to the "Find From Name" command at the top of the display. This results in a popup window to which the user can type node names supported by completion and enumeration of acceptable input. When a new node is selected a redisplay is performed with the superclass of the air base node (e.g. the facility node) as the new root of the display.

There are several ways that a new AKO tree root can be specified. The "Find From Name", and "Find Node", commands can be used from the command menu. Also the user can point to any base node on the display including the right hand display and request that that node or one of it's superior nodes be the root of the AKO tree display. This combination of capabilities enables the user to browse the KB and to examine its AKO structure.

The system is currently limited in its ability to find and refer to unnamed nodes since most of the information in the RDB is referred to by named entities. The "Find Node" command provides a limited means to identify such unnamed nodes. If the node is found in the context of some other named node then it can be easily examined (e.g. node B35).

To display tables of case frame information for a base node the user points to the desired node in either the left or right window invokes a function which redraws the right hand display for the selected node. The right hand display of Figure 23 was produced by pointing to the Dresden node in the AKO tree display and selecting it to form the basis of the case frame display. Similarly any other base node could be selected to form a new display. For example in Figure 23 information relating to the unnamed SA-6, B35, can be displayed simply by pointing to it and selecting it.

The case frame display in the right window provides a list of all of the case frames that relate to the given node or to classes and super classes of the given node. The related case frames are grouped by case frame type and by the node they relate to. In the case frame display of Figure 23 three tables of case frame information are fully visible. For the node Dresden there are two tables, a table of PART case frames, and a table of OPV case frames. For Dresden's class node, air base, a table of DRAW case frames is visible.

Each case frame table is preceeded by a comment identifying the type of case frame contained in the table. The tables themselves contain a header line which is underlined. The rows under the header present information regarding the case frames.

The first item in the header is always "case frame node" and the entries in the column under this node identify the exact molecular node used in the case frame. These nodes are presented in the table so that the user has a displayed object that refers to the case frame as a whole. This object is referred to when the user wishes to invoke commands that operate on on the case frame as a whole. An example would be the cut and paste operations which enable the user to move case frames from one base node to another.

The rest of the items in the header display the names of the arcs used in case frames of the type that the table represents. In Figure 23 we see that the PART case frame utilizes the arcs super-part, part-name, part and description. From the same figure the OPV case frame can be seen to involve the object, property, and value arcs.

The items in the rows beneath the arc names are representations of the nodes found at the end of that arc for the given case frame. For example the OPV case frame for M400 has a value arc which points to a location case frame. The printed representation of location case frames appears in the table.

14.3.4.2 Editing Capabilities

The editing capabilities of the display are all provided in the context of pointing references to items in the display. Each type of display item has a different set of operations that are available for it. These operations are invoked by pushing various combinations of keys and buttons on the pointing device. Such combinations are referred to as "mouse gestures". The operations that are available on various mouse gestures are dynamically documented in a one line display at the bottom of the screen called the mouse documentation line. Pushing the middle mouse button for any type of display item will popup a menu of operations for that item.

The editing capabilities include the ability to,

- Change the value of any item on the screen.

- Edit the displays of case frame objects.

- Cut and past nodes related to molecular nodes.
- Add new case frames, and base nodes.
- Detach case frames and base nodes from related nodes.
- Reorganize the AKO tree.

We note that nodes which become detached and isolated during the editing process are deleted.

When editing case frame displays the system utilizes information stored in the definition of that case frames type to control the editing process. For example when editing a location case frame the user is prompted for the four pieces of information required, the numerical value of the latitude, either N or S, the numerical value of the longitude, and either W or E. When the user enters such values the tool will only accepts values in the correct range and will display the acceptable possibilities if the help button is pressed.

15 EVALUATION

15.1 OVERVIEW OF APPROACH

A thorough evaluation of CUBRICON was conducted on 3 October through 6 October, 1989. The purpose of the evaluation was to assess CUBRICON with respect to measures of human-computer interface effectiveness and efficiency. Feedback about the performance of CUBRICON from these perspectives is contained in this section. Recommendations for further research are also provided.

The evaluation was performed in two parts. The first part of the evaluation focussed on human engineering issues relevant to CUBRICON. Ms. Mary Lloyd, a human factors specialist from Calspan Corporation, conducted this part of the evaluation. Ms. Lloyd has many years of experience in human factors engineering, including experience in the conduct of human engineering evaluations of prototype systems. She was unfamiliar with the CUBRICON Project and was therefore able to provide an independent and unbiased evaluation.

The second part of the evaluation was conducted to evaluate CUBRICON from an Air Force applications point of view. This part of the evaluation assessed the applicability of CUBRICON interface concepts to typical and emerging Air Force applications. Mr. Albert Frantz, an engineer from the Rome Air Development Center (RADC/COAD), was employed during this part of the evaluation. Mr. Frantz has experience in the development and management of Air Force C2 system development efforts, and therefore was able to represent the perspective of Air Force users within the present evaluation. He had no prior involvement with the CUBRICON Project, and therefore was also able to provide an independent and unbiased review.

The evaluation addressed the general goals of an intelligent human-computer interface which were outlined in the Statement of Work (SOW). Human factors issues that bear on these goals, as well as other human factors issues that relate to the CUBRICON design and future directions, were evaluated. Results of the human engineering evaluation are discussed with respect to these goals in Section 15.3.2.

There are a number of issues which constrained the approaches available for the evaluation of CUBRICON. First, CUBRICON is a prototype system. Tasks to be tested had to be confined within current CUBRICON capabilities. For example, the evaluation had to be limited to the vocabulary and grammar supported by CUBRICON, and it had to employ discrete speech input. The range of output modalities was limited to color and monochrome graphics (including a map presentation system), natural language text, synthesized voice, and tables. Also, CUBRICON is implemented on a Symbolics Lisp Machine which provided a very slow response time. This detracted from the "conversational feeling" that one gets from an interface of this nature. Subjects were aware of these limitations and tried to judge the merits of the underlying concepts in spite of implementation limitations.

Second, CUBRICON represents an exploration of new technology. It is a "one of a kind" system. It was not built as an improvement to a pre-existing system. It therefore was not possible to compare performance using CUBRICON to that using traditional technology.

Third, it was not possible to employ "real users" as test subjects. The ultimate CUBRICON applications are yet to be determined and military personnel serving in roles related to the hypothetical problems employed in the CUBRICON evaluation are not available. We did, as noted above, employ an Air Force systems development engineer who was very knowledgeable in tactical military planning tasks and with computer-based systems being developed for these tasks. We feel that the user perspective was well served by the Air Force Evaluator.

Finally, since this is a basic research effort, budgets were limited. We were only able to employ two subjects in the evaluations. While valuable insights have been drawn, there are certainly strengths and weaknesses that were not identified, and we have little information relating to how the range of individual differences will affect the effectiveness of CUBRICON design concepts.

Each of the two participants were able to use the CUBRICON voice recognition system directly, rather than working through an intermediary. While this was not originally planned, our experience in preparing for the evaluation led us to believe that working through an intermediary would have seriously hindered the evaluation process. Only with direct interaction could subjects fully experience CUBRICON and provide meaningful evaluation.

Also based on the test preparations, a script-based evaluation was added to the Air Force applications oriented part of the evaluation. This was in addition to the already planned problem solving task. By using the script which was developed for the human engineering part of the evaluation, we were able to ensure that the Air Force Evaluator would: (1) exercise and evaluate all important features of the CUBRICON system; and (2) Experience and evaluate a relatively error-free conversation with CUBRICON.

Other than these deviations, the CUBRICON evaluation was conducted according to the Test Plan delivered to DARPA and RADC in March, 1989 ("Test Plan/Procedures; Intelligent Multi-Media Interface Project", CLIN 002, ELIN A002). The evaluation attempted to identify those aspects of CUBRICON that worked well as well as those that did not work well, and to recommend enhancements and directions to guide future efforts. These evaluations were meant to be constructive. Any criticisms are not in any way intended to minimize the hard work and dedication that went into the CUBRICON development efforts. Much has been accomplished during this effort, and much remains to be done.

15.2 PROCEDURES

As stated above, the CUBRICON evaluation proceeded in two stages. The first stage employed a human factors psychologist and focussed on interface engineering issues. The second stage employed an Air Force engineer who was knowledgeable in computer-based tactical planning systems. This part of the evaluation focussed on the applicability of CUBRICON to military planning problems.

Each Evaluator received about five hours of training, including voice training and hands-on interactive practice. Each Evaluator was proficient with the procedures and techniques for using CUBRICON, before conducting their formal evaluations. The evaluations themselves were structured to the perspective of each particular Evaluator. Figures 28 and 29 show the schedule of the training and evaluations conducted.

15.2.1 Stage 1. Interface Engineering Evaluation

This stage of the CUBRICON evaluation proceeded in two steps. First, the human factors specialist interacted with CUBRICON by following a prepared script (included in Appendix E). This script was developed to exercise all important features of CUBRICON, especially as they relate to the evaluation criteria.

The second step in this stage of the evaluation involved free-form exploration of CUBRICON's capabilities. The Human Factors Evaluator was instructed to interact with CUBRICON in an open-ended fashion to: 1) more fully evaluate CUBRICON's performance vis-a-vis the evaluation criteria; and 2) Stress the system to find out where weaknesses exist. This part of the evaluation allowed the human factors specialist to tailor the interactions with CUBRICON to tease out data specifically addressing the evaluation criteria.

The Human Factors Evaluator conducted these evaluations with the aid of the engineering evaluation checklist. This checklist guided the evaluations. The evaluator was required to provide judgements about each evaluation item by checking the appropriate selection and noting the reasons behind the selection. The completed checklist is contained in Appendix E. The summary portions of the checklist are provided in Section 15.3.3.1 At the conclusion of the evaluation session, the Human Factors Evaluator answered open ended questions that allowed more general impressions to be expressed. These questions included solicitation of suggestions for improving the CUBRICON user interface. This questionnaire with the evaluator's answers is presented in Section 15.3.3.2.

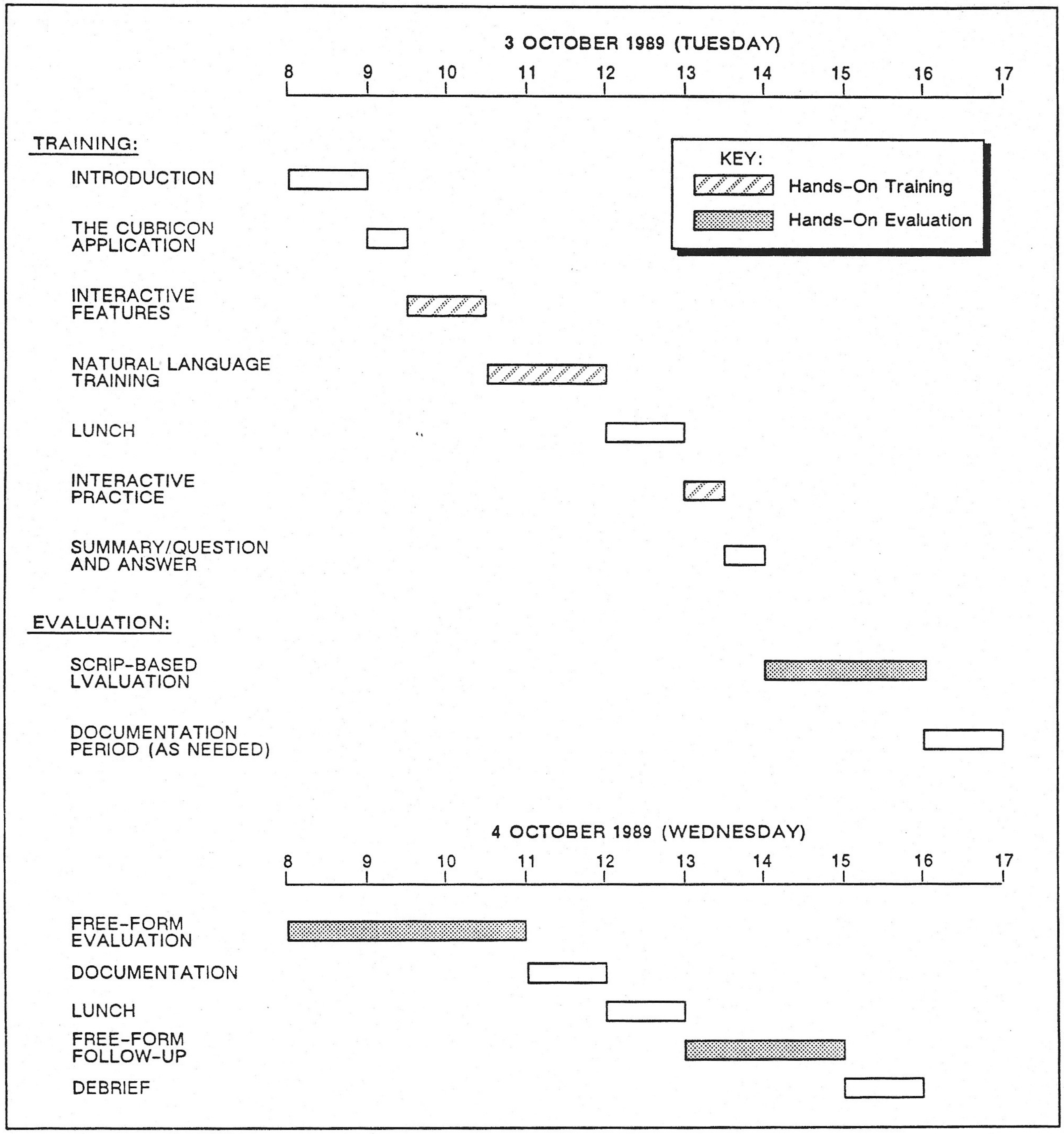


Figure 28: CUBRICON Interface Engineering Evaluation Schedule

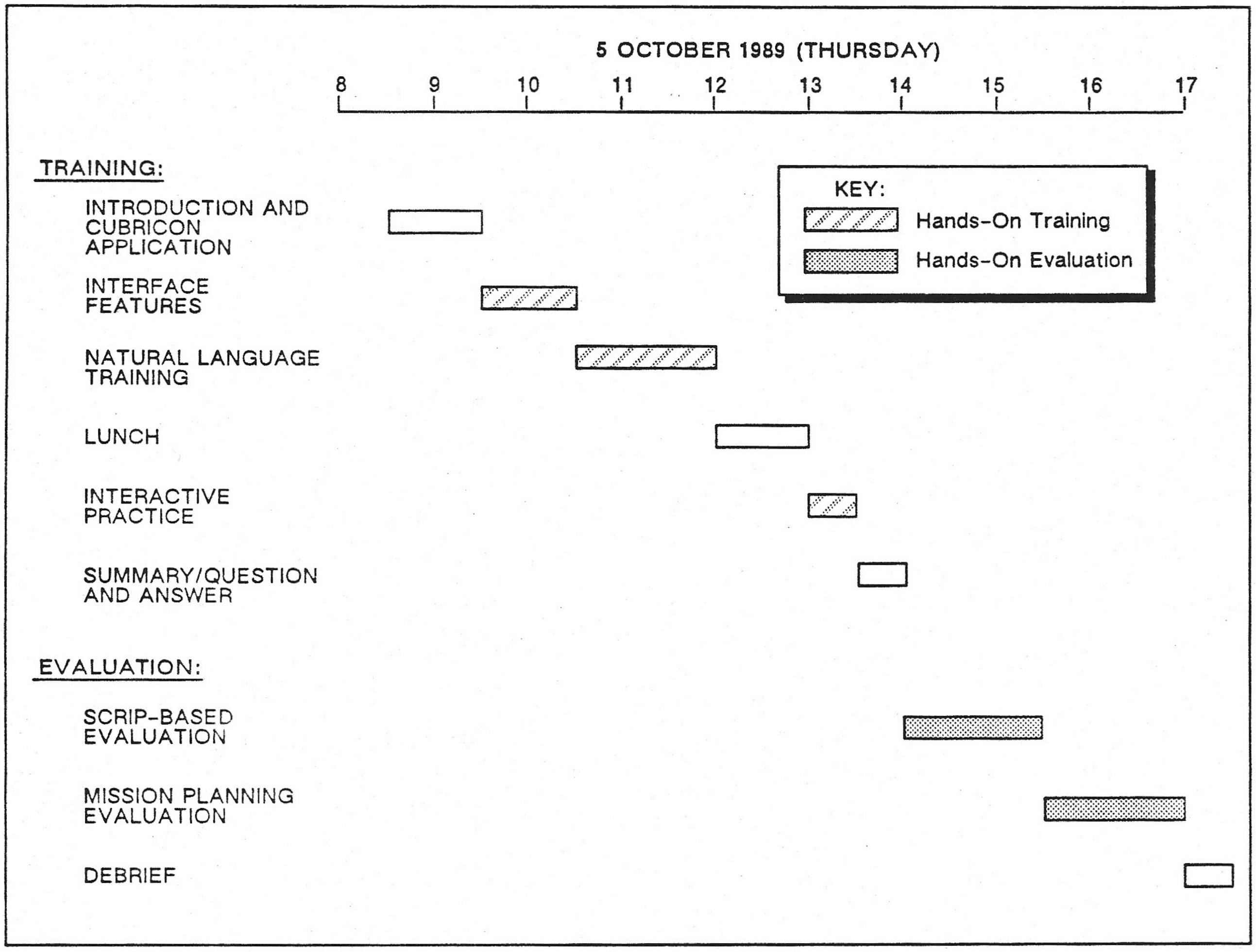


Figure 29: CUBRICON Air Force User Evaluation Schedule

15.2.2 Stage 2. Air Force User Evaluation

This part of the evaluation also proceeded in two steps. First the Air Force Evaluator used the script described above to guide the evaluation. This was not originally planned but was added to allow the Evaluator to experience all important features of CUBRICON in a relatively error-free manner and hopefully gain a better sense of conversational flow. This part of the evaluation was followed by a problem-solving task in which the evaluator was asked to use CUBRICON to solve typical military planning tasks. During this part of the evaluation the Evaluator was free to pursue the problem in any manner he thought appropriate. This part of the evaluation afforded more applied problem-solving experience.

The evaluations were accomplished through observation and subsequent debriefing of the Evaluator. These sessions were observed by the Test Conductor who recorded any difficulties experienced in using CUBRICON. The sessions were video taped and the Evaluator was debriefed with a questionnaire following the session. The completed questionnaire is contained in Section 15.3.4.

Finally, the data from this stage of the evaluation was analyzed together with the results of the first stage of the evaluations. Conclusions about the overall system design and functionality were drawn. These are summarized in Section 15.3.1. Conclusions and recommendations drawn are presented in Section 15.4.

15.3 Results

15.3.1 Summary

The results of this evaluation provide insights into the strengths, weaknesses, and effectiveness of integrated multi-media, human-computer interfaces. Specific recommendations for the improvement of CUBRICON itself were also obtained. A discussion of these results is contained in this section.

The concept of an integrated multi-media human-computer interface in which users are able to interact with a computer system via a combination of speech input/output and direct graphic interactions was supported during this evaluation. Both evaluators found that the ability to perform map- and form-based mission planning activities by pointing at objects and describing desired actions verbally, was superior to more traditional typing-based approaches. Further, the ability to interact directly on numerous windows simultaneously was found to be advantageous when information of interest was displayed on more than one window.

The concept of a unified system in which various displays and presentations reflected a single integrated underlying reality was also supported. For example, the ability to manipulate objects on one display and view the effects of that manipulation on other displays, was judged to be an important goal of integrated multi-media interfaces. In fact, the Air Force Evaluator suggested that the CUBRICON system didn't carry this concept far enough. He suggested that CUBRICON should provide tools for real-time sensitivity analyses in which parameters defining a mission plan could be manipulated in one window and the results simultaneously presented in another window.

The Air Force Evaluator found the concept of automatic window management to have merit. The automatic removal of old windows was specifically noted by both Evaluators as having potential. The Human Factors Evaluator stated that windows were usually organized for efficient use. However, she did express a desire for more user control over windowing in general. Both Evaluators liked the concept of iconizing used windows to allow subsequent recall if desired. This window iconization and recall feature is only partially implemented in CUBRICON at this time. Both Evaluators recommended full implementation. The results of the evaluation certainly support further research into automatic window management but indicate a need for making available user control over windows as well.

Several criticisms of the CUBRICON interface were also obtained during the evaluation. These tended to deal with specific interface design issues rather than more general conceptual issues. Several recommendations for the implementation of intelligent integrated multi-media interfaces can be gleaned from these specific criticisms.

The CUBRICON implementation of speech input was criticized by both Evaluators for lack of robustness. The limited vocabulary and grammar available for speech input made the formulation of inputs difficult and unnatural. Difficulties arising from the limited vocabulary and grammar, were compounded by limitations of the inexpensive speech recognition system employed which sometimes had difficulties in recognizing speech inputs. Additionally, the CUBRICON vocabulary was defined very narrowly. Available terms were understood in a restrictive way; for example, it was possible to *display* a map but it was necessary to *present* a flight path. These two terms could not be interchanged even though it would have been natural to do so. A more robust speech and natural language input capability is needed to achieve a truly natural interface.

Another limitation of the CUBRICON design noted by both Evaluators was that there was an over-reliance on speech without the availability of non-speech-based shortcuts. For example, the process of selecting an object displayed on a map, for input on the mission planning form, required natural language, either typed or spoken. It was necessary to speak in complete sentences, such as, "put this (mouseclick) SAM, here (mouseclick)." A more efficient approach (i.e., quicker and less prone to mistakes) would be to simply grab the object with a mouseclick and put it where desired with a second mouseclick (perhaps with accompanying words grab and put). The ability to point and talk is a major strength of CUBRICON. A next step is to add flexibility and allow for operational shortcuts to improve interface efficiency.

A criticism made by the Human Factors Evaluator was that CUBRICON did not provide sufficient user assistance in the way of menus, prompts, or similar types of guidance. Even the mission planning form, which by its nature provides prompts as to the information to enter on the form, did not distinguish between required and supplemental entries. It is difficult for novice users to know what the system is capable of doing, and how to undertake relatively complex tasks, without guidance. This was especially troublesome for tasks that required well defined and rather rigid procedures. More explicit user guidance should be available on CUBRICON-type interfaces to application systems. Of course, these features are very application dependant and were not the focus of this research. These features would be incorporated in a final implementation.

Error management is another area in which CUBRICON received criticism from the Human Factors Evaluator. Too often an error in entering information or requests led to the message, "sorry, do not understand request, please try again." The user, in this situation, is given no information that would help in reformulating the input. The specific aspect of the request that was not understood should be identified for the user, and provisions for correcting the misunderstood part of the input should be available, rather than requiring the user to repeat the entire input[1].

The evaluation also identified a need for more user control. This was reflected by comments from both Evaluators, but was primarily a concern of the Human Factors Evaluator. While CUBRICON attempted (with some sucess) to provide outputs that clearly provided the information desired and needed by the user, there were situations in which CUBRICON displays deviated from that which was actually desired by the user. This is inevitable. In these situations, there was little the user could do in directly affecting changes to the display format or content. For example, it is possible to zoom-in on any point on any map, but it was not possible to tell the system how much area should be included in the zoomed-in area and how much resolution to provide.

Sometimes CUBRICON provided too much information. This was noted by both Evaluators and is particularly evident in the generation of a table to supplement each map window displayed. When the monochrome display contains the mission planning form, all map

[1]An update to CUBRICON since the interface evaluation has incorporated provision for more specific feedback when inputs are not understood. This improved part of CUBRICON was not subjected to human factors evaluation, but is expected to make the process of error correction much easier.

windows and corresponding tables are placed on the color graphics display. Since only four windows will fit on the color graphics display, the creation of a table for each and every map quickly exceeds the display capability. Deletion of old windows and overlapping among displayed windows, was frequently necessary. There was often no obvious way to tell which table corresponded to which map. Suggestions were made by the Air Force Evaluator about how to present information in support of maps without requiring numerous tables (see Section 15.3.4).

15.3.2 Evaluation with Respect to SOW Goals

The Human Engineering Evaluation was developed to provide assessment of CUBRICON performance with respect to human-computer interface efficiency; and especially with respect to the human-interface goals specified in the Statement of Work (SOW). This section relates the results of the Interface Engineering Evaluation directly to these goals. As stated in the Evaluation Plan, these goals are:

1. "... minimize the requirement for translation and reformulation of information on the part of the the human. The computer should accept information from the human in a form that is natural for the human" (SOW, p. 4).
2. "Formats should be flexible to conform to individual styles yet need to be unambiguous and usable by more than one human user" (SOW, p.4).
3. "The system should assist the user in accessing an appropriate amount of information that is relevant to his needs" (SOW, p. 4 and p. 7).
4. "Machine outputs should be organized in a way that the human can easily assimilate the information within the context of the task(s) being performed" (SOW, p.4).
5. "The context of all communication must be kept clear" (SOW, p.4).
6. Speech, natural language, and graphics must be integrated for both computer input and output (SOW, p. 2).
7. "...dynamically define how information will be presented and how human/computer dialogue can be adapted based on the context of the dialogue or the decisions being made" (SOW, p.4).
8. "...track the focus space of the human/computer discourse ..." and determine "the appropriate referent of definite references, particularly those definite references involving multi-media expressions" (SOW, p. 7).

Each of these goals is restated below with summarized excerpts from the completed Interface Engineering Evaluation Checklist. These excerpts are kept as near to the original statements made by the Evaluators as possible to avoid the possibility of misrepresenting the evaluator's intent. The Evaluator's comments are shown in italics. A reference to the actual checklist item is provided in parenthesis at the end of each excerpt. The complete checklist is provided in Appendix E.

1. "... minimize the requirement for translation and reformulation of information on the part of the the human. The computer should accept information from the human in a form that is natural for the human" (SOW, p. 4).

   - **Convienience of Input Media Selection:** Inputs to CUBRICON could *usually* be made using the most convenient and desired media/modalities and in a manner that seemed natural and efficient. *Some limitations to this included: 1) It was not possible to point at the form for output; 2) The allocation of the mouse from screen to screen was cumbersome; 3) specification of location for zoom-in could only be accomplished by mouse click (i.e., could not accomplish with voice); and 4) It was inconvienient when speech input is misinterpreted and required re-entry* (1.1-1).
   - **Inputs Understand the First Time:** Issuance of commands to CUBRICON was *sometimes* efficient and easy (3.1-11). However, inputs to CUBRICON were *rarely* understood correctly the first time without the need for clarification or reformating (1.1-2). *The speech system was difficult to use due to the frequency of recognition errors* (3.1-11).
   - **Desired Terminology:** *Sometimes* verbal reference to objects within the CUBRICON data base could be made using desired and natural terminology. *The structure of command sentences was somewhat rigid (e.g., "zoom-in on this -mouseclick- point" was acceptable, "zoom-in on this -mouseclick- location" and "zoom-in on this -mouseclick- " were not). The system would tolerate ommissions of "the", however. Also, command verbs tied to specific actions were difficult to remember* (1.1-3). The CUBRICON vocabulary and grammar *usually* was sufficient for expressing desired concepts and data. *The vocabulary and grammar seemed appropriate for the application* (1.3-8).
   - **Input of Spacial and Goegraphic Information:** CUBRICON *usually* provided for efficient specification and input of spatial/geographic information when this was done. *However, this type of operation was restricted to the specification of flight paths. The main difficulty with this operation was that the feedback for*

first location specification was not provided until the second location was specified. (Test Conductor Comment: It was also not possible to make any changes to waypoints once they were made) (1.1-4).

- **Data Entry on Forms:** The use of data entry forms was *usually* straightforward and not prone to errors (e.g., areas for data entry were clearly delineated and movement between them was natural and efficient). *However, It was not clear which areas were required to be completed or whether there was a hierarchical order necessary for completing them. Format, and movement between areas was straightforward* (1.1-5)

- **Multiple Media Inputs:** The ability to point and speak at the same time was *always* helpful in making inputs. *This made CUBRICON easy to use* (1.2-1).

- **Inputs Using Multiple Windows:** It was *usually* possible to make inputs efficiently using multiple windows (e.g., pointing at objects in different windows when defining a target list). *However, it was not possible to point at the form to get inputs for another window or another part of the form (i.e., it was possible to point at the form for input, but it was not possible to point at the form to get something from the form). Also, the process of allocating the mouse from screen to screen was cumbersome* (1.2-4, 1.2-5).

- **Inputs Relative to Context and Ongoing Dialogue:** *Sometimes* the formulation of inputs to CUBRICON flowed naturally from the context of the displays and dialogue and did not require translation in order to achieve acceptable structure and formats. *Formulation of data input to the form was automatically structured by the system (e.g., "arrival time is six" formatted as 6:00) making it convienient to use. However, the structure of commands in general requires memorization* (1.3.1). Inputs to CUBRICON could *usually* be made within the ongoing dialogue without invoking special procedures or calling special displays. *Most input is done on one screen containing the form, the command scroll area, and the system (i.e., the monochrome display)* (1.3-2).

- **Recall of System Outputs:** It was *not* possible to have (verbal) messages repeated when needed. *However, speech messages were represented in the text window for subsequent review* which provided for this function (2.5-5).

- **Control of Map Displays:** It was *sometimes* possible to clearly and easily specify desires for control and transformation of maps. *However, zoom-out and recall of stored (i.e., iconized) maps/tables weren't enabled. Also, it was not possible to designate the desired area for zoom-in* (3.1-1).

- **Correcting Errors During Input:** The process of making corrections and "on-the-fly" changes during input was *rarely* straightforward and efficient. *In* *fact, it was extremely difficult or impossible to change or undo something, such as adjusting points on the flight path. Also, incorrect text due to speech recognition errors was tiresome to correct* (3.2-1). A requirement for an explicit ENTER action prior to CUBRICON processing of user inputs *usually* was imposed when necessary to permit user review or reconsideration. Text input did not require "enter", however. Punctuation served this function. This is not a typical method. Most users will probably be accustomed to the use of the ENTER key (3.2-2).

2. " Formats should be flexible to conform to individual styles yet need to be unambiguous and usable by more than one human user" (SOW, p.4).

   - **Ease of Graphic Interactions:** Pointing at desired objects *usually* could be accomplished equally well on the various types of windows displayed (e.g., tables, maps) and on the monochrome display as well as the color display. *However, pointing on the form was only enabled for input to the form (not for output from the form). Also when numerous icons were displayed in close proximity, more than one icon was picked-up by a point and the system crashed* (1.1-6).
   - **Standardization Across Displays:** Standard displays *usually* used standard formats that were readily identifiable and usable (e.g., standard information was contained and consistently organized in display headings). *However, windows were not uniquely identified with labels* (Test Conductors Note: The lack of labels on windows made it difficult to determine which maps were associated with which table) (2.4-7).
   - **Availability of Prompts:** CUBRICON *never or rarely* provided prompts to help in making standard or required inputs (e.g., guides for accomplishing complicated procedures) or as a reminder when omissions were inadvertantly made. For example, *there were no prompts available for filling out the form that helped identify what areas were required to plan a flight path. Also, error feedback was not informative or diagnostic* (2.4-8, 1.3-4).
   - **Display Customization:** It was *never* possible to customize displays to meet personal preferences (e.g., reorganize table columns, redefine area displayed on a map, redefine window layout) (3.1-2).
   - **Window/Display Management:** The automatic management of windows (e.g., positioning, sizing, and removal) *sometimes* was accomplished in a way that facilitated their use. *Support for user intervention to achieve alternative window organizations when desired was limited* (3.1-3).
   - **System Operations:** System operations *usually* reflected user inputs and desires in a logical fashion (3.1-10).

3. " The system should assist the user in accessing an appropriate amount of information that is relevant to his needs" (SOW, p. 4 and p. 7).

   - **Map Displays:** Map displays *sometimes* contained an appropriate amount of area at an appropriate scale (without resizing) for task accomplishment (e.g., zoomed-in or out to correct amount of detail and area coverage). *User control of zoomed area size would enhance system, along with addition of zoom-out feature* (2.3-1). Map and other graphic displays, and symbols used within them, *usually* were large enough to provide the resolution needed to resolve objects and determine necessary relationships among objects. *In one instance a cluster of icons couldn't be deciphered. Otherwise, maps and symbology were easy to read and access. A Removable grid would be useful for distance relationships. Also, tables on the graphic display were difficult to read due to text size* (2.3-2).
   - **Level of Detail in Displays:** CUBRICON responses to requests for information *usually* provided the information in a level of detail consistent with the request and the context of the request (e.g., only necessary information was displayed, yet sufficient detail was provided for the task) (2.3-3).
   - **Managing Displays and Access to Large Volumes of Information:** It seemed that the information being displayed was *usually* well controlled (e.g., not overwhelming). *Automatic declutter (i.e., removal of old or unimportant windows) prevented information from becoming overwhelming* (2.3-6). When a request for information resulted in a large volume of information, CUBRICON *sometimes* provided a means for dealing with the information in an organized and efficient manner, and/or helped the user rescope the request. Many of the techniques that could help accomplish this goal such as scroll bars (*not yet implemented*), presentation of the amount of information to come, and the use of information summaries, *are not available on CUBRICON. Further, in some situations it wasn't clear that information on tables could be scrolled at all* (2.3-4).
   - **Highlighting of Critical Information:** *Sometimes* an appropriate means for highlighting critical information was used (considering the nature of the critical information, the task context, and other coding schemes in use) *Required information on the form (for planning a flight path) wasn't indicated. Highlighting on the map was effective. Flashing of items in some instances was excessive and inappropriate* (2.3-5).

4. " Machine outputs should be organized in a way that the human can easily assimilate the information within the context of the task(s) being performed" (SOW, p.4).

- **Display Clarity:** CUBRICON outputs were *sometimes* clear and understandable without requests for clarification. *Voice was difficult to interpret on many occasions, however, this problem could disappear with increased training and usage* (2.1-1). Information needed for interpreting displays *sometimes* was readily available. *Maps and tables were rarely uniquely identified. A key wasn't available for decoding map symbols, but I'm not sure it is necessary* (2.1-2).

- **Labelling:** CUBRICON displays *sometimes* employed labels that were clear, consistent, and helpful. *Labels were available within displays, but labels for the displays themselves (maps and tables) were not available. Labels within the form would probably be more meaningful to the mission planner* (2.1-3).

- **Window Layout:** The general organization and layout of windows was *usually* efficient for the tasks at hand. *Window size seemed appropriate. The location of windows was sufficient for task completion except when automatic deletion removed a map still in use* (Test Conductor's Note: this occurred following a permanent zoom-in when the context map map was removed) (2.1-8).

- **Speech Outputs:** CUBRICON speech outputs were *usually* constructed in a way that maximized overall communication efficiency and understandability. *Terminology and phraseology was standardized and consistent across the entire interface and vocabulary/terminology for speech input was similar to that for speech output* (2.5-2).

- **Information Coding:** CUBRICON displays *sometimes* employed coding schemes that were clear, consistent, and adequately captured the important distinctions among display elements. *However, the maps and tables themselves need to be labelled. Map icons were easy to distinguish from each other. Boxing of highlighted items on tables were difficult to distinguish. Bold face type would be easier to see. Also, the red arrow pointer was difficult to see in enemy territory. Use of a distinct color would make it easier to distinguish* (Test Conductors Note: when multiple flight paths intersect at a common waypoint it is not possible to tell which flight path goes with which mission) (2.1-4). Auditory and voice coding was *rarely* employed to effectively communicate important distinctions among auditory displays (2.2-9).

- **Media/Modality Selection:** Outputs were *usually* presented using media or modalities that were appropriate for the content and context of the communication. *The ability to obtain a hard copy would enhance the system* (2.2-2). Information that was presented for comparative purposes, *usually* was displayed in a manner suited for such comparisons (e.g., side by side in a table, highlighted on a map using clear distinguishable codes, etc.). *Comparison of maps could be made* *easier by allowing the user to pull a map up from storage (i.e., recall of iconized windows which are not yet implemented). Also, two different forms cannot be displayed side-by-side for comparison, but this may be necessary* (2.2-3).

- **Map Displays:** Maps were *sometimes* presented in a way that facilitated their readibility and use. *Occasionally areas of the maps were overly cluttered with icons that couldn't be differentiated. Also, because windows themselves are not labelled, there was no way to quickly see which tables were asociated with which maps. Map scales were also ambiguous* (2.2-5).

- **Multi-Media Output Coordination:** It was *sometimes* possible to relate items in tables or on forms to their graphic representations (e.g., on a map). *Maps and tables need to be uniquely identified and the association between them more definitively demonstrated (e.g., a line connecting them or coded in some way)* (2.2-7).

- **Voice Outputs:** Voice outputs were *sometimes* constructed in a manner that facilitated accurate perception and understanding. Important words were *usually* placed near the end of messages so that surrounding sentence structure would provide context and facilitate intelligibility. *However, when voice output was given regarding mission Duration the message, "the duration is this" was given, thus failing to provide critical information. Multi-syllable words were used when appropriate* to provide linguistic redundancy and reduce phonemic uncertainty within any given word (Test Conductor's Comment: when CUBRICON said "the duration is this" it also blinked the appropriate icon) (2.5-3).

- **Display Dynamics:** Display changes *usually* did not disrupt the ongoing dialogue (e.g., did not remove needed windows, display changes were consistent with expectations). *However, it was not possible to input data while display changes were in progress. Also, the E-W Germany context map (following a permanent zoom-in) would occasionally be removed when it still served a useful reference function* (3.1-4). Output formats were *usually* consistent with expectations based on the preceding dialogue and the context of pre-existing displays (2.4-6).

- **Response Time:** The response time for voice, text, and graphics inputs was *rarely* sufficiently fast to ensure efficient, continuous dialogues. *System response time was too slow* (3.1-13).

- **Presentation of System Status:** *Sometimes* CUBRICON clearly communicated its activities when processes were not immediate. *The Symbolics status line provided for this function by saying "run" or "user input" but specific CUBRICON activities weren't communicated* (Test Conductor's Comment: there was a prompt on the text window that indicated when CUBRICON was ready to accept inputs) (2.1-6).

- **Consideration of the Temporal Context:** Information that was needed temporarily was *usually* made available on a temporary basis (rather than permanently cluttering displays with such information). The use of window overlays served this purpose. *The ability to display and remove windows as desired, would enhance this aspect of the system* (2.2-11).
- **Tables:** Tables *usually* presented information in a manner that facilitated efficient use. *However, ordering of column information may need to be rearranged. Feedback from military type would be helpful (e.g., the "name of item" column was imbedded between other columns and was the 5th of 6 columns)* (2.2-4).

5. "The context of all communication must be kept clear" (SOW, p.4).

- **Understanding Inputs Based on Dialogue Context:** Inputs that were illogical based on the task and data context, were *rarely* noted by CUBRICON and communicated. *There was very little error trapping. The system would proceed with next command without recognition that required information was omitted. Error feedback messages were not informative* (Test Conductor's Note: CUBRICON would state that it didn't understand a request, but would not state the source of the misunderstanding or help the user reformulate the input) (1.3-3).
- **Highlighting of Contextually Important Items:** When items were selected (by the user or the system) this was *usually* conveyed clearly to the user. *However, the first waypoint defined when drawing a flightpath isn't indicated until second point is selected. Also selected items on tables were boxed. This was hard to decifer, especially on the color graphics display. Bold face would stand out better* (2.1-5).
- **Spatial/Geographic Context:** Spatial relationships among graphic elements (e.g., elements on a map) were *rarely* presented clearly. *It was not possible to querry for exact distances. An option to impose a grid on the map would be helpful. Also part of the scale on the map was obscured at the origin* (Test Conductor's Note: The scale presented on the border of the map was the UTM scale, while voice references used latitude and longitude. This was noted during the program but was not corrected because it would have taken engineering resources away from more critical activities. Updates to CUBRICON should correct this problem) (2.1-7).
- **Display of Context Needed for Information Interpretation:** When displayed information was relevant only in a certain context, this *usually* was clearly

communicated. *However, not all pertinent time stamps were given on the map during flight path presentation (e.g., target strike time was not presented)* (2.4-1). Adequate contextual information was *usually* available to assist in the proper interpretation and use of displayed information (2.4-2). When information was best interpreted relative to some significant level or critical value, this comparison was *usually* clear from the display. *However, some critical times (departure time, strike time) were not presented during the flight path presentation* (2.4-3).

- **Distinguishing Active from Inactive Displays:** When not all items were active, CUBRICON clearly indicated which were active (Test Conductor's Note: all windows were active except those that were iconized). *Recall of inactive windows (i.e., those that were iconized) was not enabled* (Test Conductor's Note: This feature was planned but not implemented within the scope of the present program) (3.1-7).
- **Display of Information Structure and Relationships** CUBRICON *sometimes* communicated information in a manner in which the structure of and relationships among the data data being entered or displayed were clear. *However, hierarchical relationships among data on the form wasn't clear* (Test Conductor's Note: The form employed by CUBRICON was built based on existing Air Force planning forms. There was no attempt to improve upon the form design during the program other than to make modifications necessary to make it fit on the CRT) (2.4-4).
- **Maintaining Context During Display Dynamics:** When displays were changed (e.g., removing windows or information, zoom-in or out, panning, scrolling), adequate cues were *sometimes* provided for maintaining orientation to the larger context. *When performing a zoom-in, CUBRICON noted the area to be zoomed-in with a box drawn on the original map. However, there were no cues provided for scrolling context on scrollable windows* (2.4-5). Windows were *usually* managed in a way that minimized disruption to display context (2.4-9).
- **Feedback of CUBRICON's Understanding of Inputs:** Feedback about CUBRICON's acceptance and understanding of inputs *usually* was sufficiently quick and clear. *However, system response was slow and acceptance of the first point on flight path wasn't clear* (3.1-5).
- **Ability to Tell System to Ignore Verbal Inputs:** There was *always* a simple means for indicating to CUBRICON when verbal inputs were meant for CUBRICON and when they were not (e.g., ignore and continue) (3.1-6).

6. "Speech, natural language, and graphics must be integrated for both computer input and output (SOW, p. 2)."

- **Coordination of Multi-Media Input:** Mouse points *usually* were correctly related to the intended objects described via natural language (1.2-2). It was possible to point at multiple objects as part of an input, and these *usually* were correctly integrated and understood within the dialogue by CUBRICON (1.2-3).

- **Design of Multi-Media Output:** Speech, graphic, and textual outputs were *usually* used appropriately and in the right proportion to clearly, concisely, and efficiently accomplish the necessary communications. *However, on some occasions the presentation of both voice and text messages seemed overly redundant. The requirement of having to processing both voice and text seemed to increase workload. Graphic output was clear and easy to use. An exception to this was the flashing of data in the text window after a flight path was presented* (2.2-1).

- **Clarity of Cross Media/Modality References:** CUBRICON *usually* made unambiguously clear, which graphically displayed objects were referred to via an associated media/modality. (e.g., verbal outputs were related to associated displayed items in a clear and unambiguous fashion) (2.2-6).

- **Effectiveness of Cross Media/Modality Orientation:** CUBRICON speech output *sometimes* was helpful in providing orientation to other system outputs (e.g., created or modified maps, tables, etc.). *Voice output may have been more useful if system response times were faster or under greater workload conditions* (2.2-8).

- **Display Integration:** When relations among information components are important, integrated displays (individual or multi-media) that show those relations were *sometimes* provided. *The relationship between map and entity tables wasn't demonstrated* (Test Conductor's Note: It was difficult to determine which table was associated with which map) (2.2-10).

- **Temporal Coordination of Multi-Media Outputs:** CUBRICON speech outputs were *sometimes* coordinated with ongoing tasks and outputs using alternate modalities. *However, this was a problem because of slow system response time* (2.5-4).

7. "...dynamically define how information will be presented and how human/computer dialogue can be adapted based on the context of the dialogue or the decisions being made" (SOW, p.4).

   - **When to Use Speech:** The following items relate to the appropriateness of CUBRICON's decisions about when to use speech output (2.5-1):

- CUBRICON speech output did not interrupt user inputs, *because most of the time, speech output occured when processing was taking place and input data could not be made. However, it was not possible for the human to interrupt (i.e., override) the speech output. A future enhancement could a method for stopping voice output when user input occurs.*
- *It was difficult to evaluate whether speech output was used when there was a requirement for rapid two-way exchanges of information, since system response time very slow* (Test Conductor's Comment: Good human factors design would employ speech when there is a repuirement for rapid two-way exchanges of information).
- *Speech was used to inform the user about display events that were about to happen, and to present information about displayed items* (Test Conductor's Comment: this is consistent the human factors guideline that speech be used when the information to be presented deals with a future time requiring some preparation, and especially when it is intended for immediate use).
- *Speech was used to draw attention to the appropriate display* (Test Conductor's Comment: This is consistent with the human factors guideline that speech be used when it is important to elicit attention from other tasks or activities).
- *Speech presentation allows user to fixate on map display activity while receiving information about the display via voice output* (Test Conductor's Comment: This is consistent with the human factors guideline that speech be used when information must be presented independant of head or eye movement). *However, at other times the voice output seemed extraneous and overly redundant.*

- **Control Over System Processes:** It was *always* possible to cancel partially completed inputs (including voice inputs) and ongoing CUBRICON processes by invoking an explicit cancel command (Test Conductor's Note: It is not possible to stop ongoing CUBRICON processes once they are initiated) (3.1-8).
- **Control of Displays:** Control of the CUBRICON interface was *sometimes* handled effectively. Updating of displays was efficient and did not require excessive effort. *However, very little control of windowing operations was available to the user if desired* (3.1-9).
- **Control of Dynamic Presentations:** It was *rarely* possible to maintain control over dynamic displays (e.g., PAUSE and CONTINUE commands). *Specifically, there was no user control over flight path presentation once initiated* (3.1-12).

- **Error Protection:** There was *never* ample protection against actions that result in the deletion or significant altering of information (e.g., warnings, undo capability, feedback about results of change prior to action, etc.) (3.3-1).

8. "...track the focus space of the human/computer discourse ..." and determine "the appropriate referent of definite references, particularly those definite references involving multimedia expressions" (SOW, p. 7).

   - **Pronoun Referents:** CUBRICON was *usually* able to correctly relate pronouns and indefinate references to their proper referent (based on the preceding dialogue). *Voice output used pronouns correctly. However,pronouns were not accepted as input* (1.3-5).
   - **Correct Interpretation of Inputs:** CUBRICON was *usually* able to correctly interpret inputs based on the context of the dialogue (e.g., requests for information produced outputs relevant to the dialogue; requests that made no sense based on the context were questioned) (1.3-6).
   - **Resolution of Ambiguous Mouse Points:** Ambiguous mouse points were *usually* resolved correctly by CUBRICON based on the context of the dialogue (i.e., inaccurate points were correctly resolved; Incorrect points that made no sense were corrected or questioned). *CUBRICON did have a problem when a selected icon was in close proximity to other icons and there was no information to allow disambiguation* (1.3-7).

15.3.3 Interface Engineering Evaluation

This section contains the results of the interface engineering evaluation, conducted by the human factors psychologist. It is broken into two subsections. The first subsection contains the results of the Interface Engineering Evaluation Checklist. These are provided at the top-level only. The complete checklist, including supporting data, is provided in Appendix E. The second subsection includes the completed Interface Engineering Evaluation Questionnaire.

15.3.3.1 Overview Interface Engineering Ratings

This section contains the summary portions of the Human Engineering Evaluation. Responses of the Human Factors Evaluator are presented in italics. The entire checklist containing all Evaluator responses is contained in Appendix E.

The following instructions appeared at the top of the Interface Engineering Evaluation Checklist:

> Rate CUBRICON's performance with respect to the evaluation categories. The numbered items (-1, -2, etc.) within each category will help in making your assessments. These numbered items are not intended to serve as the sole basis upon which to make your assessments. All observations you believe are relevant should be considered. State the rationale on which you base your ratings.
>
> Refer to Smith and Mosier [Smith86] to guide your evaluation. Many of the numbered items include references to Smith and Mosier. These references are listed within parentheses at the end of the items. Bear in mind that CUBRICON is built using new technology. Its approach to user-interface design is new and innovative. The guidelines in Smith and Mosier were developed for conventional interfaces. If CUBRICON violates any of the Smith and Mosier guidelines, ask yourself whether the violations could represent an improvement over conventional user-interface approaches, or whether they are the result of poor design.
>
> Finally, be critical! Don't be afraid to tell us what you think (good and bad). Stress the system. Find out where its weak points are and tell us how we can make it better. If you need more time, take it. The results of this evaluation will guide future design efforts.
>
> Note: The numbered items within the evaluation categories are also cross-referenced to the top-level CUBRICON goals that were stated in the SOW. These are noted using the *number* format. These references are not meant to be used during the hands-on portion of the CUBRICON evaluation but will be used during subsequent analysis and reporting.

Answers provided by the human factors psychologist to the top-level questions on the checklist are provided below. As noted above, the complete checklist is provided in Appendix E.

-1- The Efficiency and Effectiveness of Making INPUTS to CUBRICON.

1.1 Rate the general ease, naturalness, and effectiveness of making inputs to CUBICON:

Rating Comments

Excellent	.....	*Frequent misinterpretations of speech input was inefficient. However, a better speech recognition system and increased training and practice could alleviate this problem. Allocation of mouse to screen (color graphics or monochrome) via the keyboard was cumbersome. This could be improved by using the right and left buttons on the mouse to select the desired screen. The use of specific command verbs to initiate specific actions was difficult to remember, particularly when the verbs have similar meanings (e.g., display and present). A more generalized use of verb commands would lessen memory load. The option of input media (speech, text, pointing) or combinations of media made the system enjoyable to use. It accommodates differences in task demands and user preferences.*
Very Good	.....	
Adequate	..X..	
Poor	.....	
Extremely Poor	.....	

1.2 Rate the ability of CUBRICON to accept, integrate, and understand inputs that were made using multiple media/modalities:

Rating		Comments
Excellent	.....	*Being able to point at several objects as part of an input, the combined use of speech and pointing for an input, and the use of multiple windows for an input are all excellent features that made the system easy to use.*
Very Good	..X..	
Adequate	.....	
Poor	.....	
Extremely Poor	.....	

1.3 Rate the ability of CUBRICON to understand inputs based on the dialogue context:

Rating		Comments
Excellent	.....	*Formatting of speech input was somewhat rigid. The system was not very tolerant of deviation from this structure. Allowing verb commands with similar meanings (e.g., display and present) to be used interchangeably would be helpful.*
Very Good	.....	
Adequate	..X..	
Poor	.....	
Extremely Poor	.....	

-2- The Efficiency and Effectiveness of CUBRICON OUTPUTS.

2.1 Rate the general understandability, effectiveness, and smoothness of CUBRICON outputs:

Rating		Comments
Excellent	.....	*Voice output was sometimes difficult to interpret. Additional labelling of maps and tables would be helpful. Outputs were generally clean and easy to understand.*
Very Good	..X..	
Adequate	.....	
Poor	.....	
Extremely Poor	.....	

2.2 Rate the appropriateness and effectiveness of CUBRICON media/modality selection and integration:

Rating		Comments
Excellent	.....	*I think the integration of media was effective. Its effectiveness would probably be more apparent with a faster system response time and heavier workload conditions. Voice messages about map display changes were convenient since it allows the user to remain fixated on the display while changes are described.*
Very Good	.....	
Adequate	..X..	
Poor	.....	
Extremely Poor	.....	

2.3 Rate CUBRICON's effectiveness at selecting and controlling output quantity, level of detail, and resolution:

Rating		Comments
Excellent	.....	*Need to indicate scrolling option on tables. On the whole, maps were easy to read and use. Text in tables on graphics display was difficult to read* (too small).
Very Good	.....	
Adequate	..X..	
Poor	.....	
Extremely Poor	.....	

2.4 Rate how well CUBRICON maintained context clarity:

Rating		Comments
Excellent	.....	*Marking of original zoomed-in area boundary on original map was helpful. The relationship between maps and tables needs to be made explicit. Labels on maps and tables to uniquely identify contents are needed.*
Very Good	.....	
Adequate	..X..	
Poor	.....	
Extremely Poor	.....	

2.5 Rate the appropriateness and effectiveness of voice output as used within CUBRICON integrated outputs:

Rating		Comments
Excellent	.....	*Evaluating the use of voice output is problematic because system response time was slow and this magnified the feeling that voice output was extrinsic to the task. A frequent user of the system would not require as much voice feedback as is currently provided and a means of adjusting the level/amount of voice feedback should be addressed in future enhancements of the*
Very Good	.....	
Adequate	..X..	
Poor	.....	
Extremely Poor	.....	

system.

-3- Sequence and System Control Issues

3.1 Rate the efficiency and effectiveness with which the CUBRICON user-interface was controlled:

Rating		Comments
Excellent	.....	*User control of windowing was limited,*
Very Good	.....	*although some features that would*
Adequate	..X..	*improve user control (e.g., zoom-out)*
Poor	.....	*were not enabled yet.*
Extremely Poor	.....	

3.2 Rate the efficiency and effectiveness of error management and control within the CUBRICON user-interface:

Rating		Comments
Excellent	.....	*Making corrections to text input may*
Very Good	.....	*be easier if user has knowledge of*
Adequate	.....	*EMACS. Without this knowledge, re-typing*
Poor	..X..	*of whole lines is required as text can't*
Extremely Poor		.....
		function of current changes being made
		to the system.

3.3 Rate how well CUBRICON performs the functions of data protection:

Rating		Comments
Excellent	.....	
Very Good	.....	
Adequate	.....	
Poor	.....	
Extremely Poor	..X..	

15.3.3.2 Completed Interface Engineering Evaluation Questionnaire

This section contains the questionnaire completed by the Human Factors Evaluator. Evaluator responses are presented in italics.

The following instructions were provided at the top of the Interface Engineering Evaluation Questionnaire.

> This questionnaire is intended to provide general information about how well you believe CUBRICON performed with respect to human factors considerations, and to solicit suggestions for improvement. Answer the following questions and be prepared to discuss your answers."

The questionnaire along with the answers provided by the Human Factors Evaluator, are contained in their entirety below:

1. How would you rate the overall "user-friendliness" of the CUBRICON user interface? Why?

   Low. Error messages were not informative; No help provided for structuring commands properly (system just says "... can't understand..."); Very little control over processes; No indication on form of required information; feedback about inputs and system status was redundant. There was also sometimes a mismatch between voice I/O and text displayed in the text box (e.g., in response to a question about mobility while pointing at an airbase, the voice response was "complete miss on mousepoint one", while the text box said that the airbase was not mobile).

2. What aspects of CUBRICON did you find to be especially efficient and helpful in accomplishing desired actions? Explain.

   Liked using the mouse for specifying location and entering information on the form by pointing. Also liked the ability to speak and point at the same time in forming inputs. Nice to not have to type.

3. In what ways do you think the CUBRICON interface approach or philosophy provides advantages over more conventional human-computer interfaces? How can

these advantages be more effectively realized?

Lets you select the mode of input (e.g., voice vs. typing vs. pointing), but need a better speech recognition system. The CUBRICON interface can better accommodate individual differences/preferences and a broader variety of task demands.

4. Are there any aspects of the CUBRICON interface approach or philosophy that are inferior to conventional approaches to human-computer interface design? What are they? Why are they inferior?

   User has very little control. Cannot determine what to declutter (e.g., can't choose which windows to remove to make room for other windows to be displayed – this decision is made automatically by the system. Also when a window is removed, it is not possible to get it back (implementation of iconized window recall feature will overcome this limitation). Finally, the form, when displayed, covers up pre-existing windows (primarily tables) on the monochrome display.

5. Did you encounter any problems in using CUBRICON? What were they? How could they be avoided?

   - *Extremely difficult to correct errors.*
   - *Can't zoom-out.*
   - *Can't specify parameters for zoom-in (e.g., what area to cover).*
   - *System response time is too slow.*

6. Would menus or some other dialogue style be a better method in certain circumstances? What are the circumstances? What other dialogue methods would be better? Why?

   Yes. Although the integrated environment, with everything on one screen, provides freedom to combine multi-media inputs as desired, you don't know what to do: what are the options?, what are the procedures? what are the limitations (e.g., what verbs can be used?)? When a particular procedure must be employed, or when only a limited number of ways of doing things or entering information are available, a menu of choices or procedural aids should be used. This may not be as much of a problem for experienced users and simple applications (like the current CUBRICON application). However, for novice users it is an important *consideration.*

7. Are there any risks that must be kept in mind when applying the CUBRICON interface to specific applications? What are they and how can they be avoided or reduced?

   May not be applicable to all applications. For example, applications that do not employ maps or graphically oriented interfaces may not be well suited to a CUBRICON-type interface.

8. Do you think this line of research should be continued, or should it be redirected in some way or discontinued entirely? Explain.

   Continued. Multi-media has potential for many tasks. May even want to consider adding additional media such as head or eye movement for computer input (e.g., situations where hands are unavailable such as aircraft pilot tasks).

9. In what ways do you believe CUBRICON could be improved?

   – Provide option to turn off (or limit) voice output. Could become too much, especially for experienced user.
   – Vocabulary (particularly command verbs) should be understood in all relevant contexts and uses (e.g., should be able to display a flight path, and not just present it).
   – Interplay between voice output and other CUBRICON outputs is difficult.
   – Provide more robust voice recognition system. Can't always remember how a word was trained (e.g., the, pronounced tha, and the, pronounced thee).

10. Are there any other comments you care to make that will help us continue this research or improve CUBRICON.

    I think that many of the criticisms noted above may be due largely to the fact that the system is a prototype system, still undergoing modifications. It may not really have been ready for final human factors testing. For example, the lack of informative error feedback or the inability to zoom-out may mean that these features were not in place yet.

To make the system more user-friendly, however, the user needs guidance as to what operations can be performed, the procedures required, and the power to control operations.

I enjoyed using the system and liked being able to select the input media that seemed appropriate to the task being performed. The media were integrated well and output was more informative with multi-media.

15.3.4 Air Force User Evaluation

This section contains the questionnaire completed by the Air Force User Evaluator. Evaluator responses are presented in italics.

The following instructions were provided at the top of the User Evaluation Questionnaire.

> This questionnaire is intended to provide general information about usability and applicability of CUBRICON within military mission planning applications, and to solicit suggestions for improvement. Answer the following questions and be prepared to discuss your answers.

The questionnaire along with the answers provided by the Air Force Representative, are contained in their entirety in this section.

1. Do you think an interface like CUBRICON would provide an effective tool for working with computer-resident data bases and related military mission planning tasks? Why?

   Yes. If the speech capability would support continuous speech, it could be faster and more efficient than typing. In general, any of the capabilities that would allow the planner to work faster could be helpful.

2. What aspects of CUBRICON did you find to be especially efficient and helpful in accomplishing desired actions? Explain.

   I thought that the speech understanding and parsing had the most interesting potential. Another area that has potential is the automatic removal and handling of windows for the user. The idea that a pop-up window would not cover up a portion *of the map that was recently referenced was a good idea. Some of the other concepts, I take for granted because of my work with the Symbolics and TEMPLAR, but these represent a significant improvement over existing command and control capabilities. One area I really liked was the language learning. For terms that I thought were to long like fourty-fifth-tfw-ef-111, I abbreviated it as fourty-fifth and it then expanded it to the required term for the command line interface. I also think that the idea of expanding out the targets to show the aimpoints is an excellent idea and helps to declutter the map from all the targets.*

3. What aspects of CUBRICON did you find to be especially difficult to use or inefficient? Why?

   In general I didn't like the interface to the forms and tables. Often the easiest way to use a form would be to mouse on a slot and type or speak to enter a value. The user should have to do a minimal number of entry modes, during execution of a specific process.

   If I am talking and moving the mouse to point to things I don't like the idea that I have to switch modes to keyboard to enter a function-X to mouse on something on the map or a table. The tables should pop up on the monochrome display so they can be moused with just a click. It would be nice to be able to display only the part of the form you were working on ie. a single mission in a window and be able to iconize it when you are done. Then when you want to look at the package show them in an integrated way. The system queries after they were parsed seemed much to slow and might make CUBRICON difficult to demonstrate unless it speeds up significantly on a larger Symbolics. Since this is a 6.1 effort designed to show something else it is not really an issue but does detract from CUBRICONs use.

4. How would you compare CUBRICON's approach to working with computers to other more conventional computer interfaces? Describe specific features of both types of systems used in your comparison, and state whether CUBRICON was better, worse, or about the same in terms of its capabilities.

   Once again my answer to this one is slanted because I use a Symbolics at work and an Amiga at home, so I am very familiar with window/icon/mouse/pointer interfaces in addition to animation. In general CUBRICON windows are better than the ASCII type displays that still dominate existing command and control sys-

tems and IBM PC type interfaces, but are not as good as some of the windowing systems I have seen that work interactively (i.e., dynamically tracking resources available in one window while changes are being made in another). This was one of the key complaints about TEMPLAR that you couldn't do stuff like that because of the single form format of the monochrome screen.

5. Did you encounter any problems in using CUBRICON? What were they? How could they be avoided (e.g., better training, redesigning)?

   Yes, Several bugs in software. If I deviated much from the script things tended not to work (like asking for ac-pools at Rhein Main). Once in a while I would have trouble getting the DEC talk to understand certain words. Parsing speed appeared very good, but execution of the commands was too slow.

6. Consider the following specific and comment on how well or poorly CUBRICON performed with respect to them:

   - Organizing outputs for understanding.

     - There was a tendency to repeat things, i.e., Blink a Base, point at it with a label, then re-displaying a table with it added to it, talking and printing text in the text screen seemed like overkill.

   - Keeping track of routine information.

     - The Iconize screen would be helpful if they were labeled with something understandable and could be de-iconize. Also see my comment on tracking the resources in 4.

   - Keeping you informed about the overall situation and progress towards task accomplishment.

     It keeps you informed about the overall progress (possibly over-informed). It doesn't really tell you what you have to do to complete the overall task (the order in which things have to be done).

   - Allowing you to make inputs easily and efficiently.

Speech was easy but not always more efficient than typing in a value. It would be real nice to be able to click on a slot form, say the value and have it entered in the slot. Saying "Enter six here ¡point and click¿ enter" was to slow. I also liked the ability to extract thing with the mouse, but here it might be easier to point and click to extract it then slide over to the window and click another mouse button to put it somewhere. The system currently requires "Enter $< typefunction - X >< Click - nurnberg >< typefunction - X >$ *here* $< clickOCA - origin2 >$ *enter."*

- Allowing you to focus on the application.

  Overall it tends to keep you focused on the application unless the speech is failing or response time is too slow.

- Meeting your personal preferences for problem solving approach and information display approaches.

  Partially; better windows and faster speech would be nice.

7. How can we make CUBRICON better?

   Windows and icons should be exposeable with just a mouse click. Continuous speech would be nice. More understandable (human-like) output speech. The table information should high-light when you have the mouse on it and bold when selected. It would be nice if the color and mono screens were connected to each other, i.e., if you move up/down/left/right on one screen you end up down/up/right/left on the other screen instead of function-X. Some map features like display the object name in a small sub-window when the mouse is over it and instant tracking of the lat-long in another would be nice. Being able to do inquiry about an object with a mouse click or having the ability to assign functions to user assignable hot-keys also would be nice.

15.4 Conclusions and Recommendations

Briefly, the following conclusions and recommendations can be drawn form this evaluation:

- Continue research in intelligent, integrated multi-media interfaces (great potential).
- Speech/vocabulary must be robust. It must capture the fluid and variable form and style in which language is organized and used to express ideas and information.
- Provide for ultimate user control. Automatic interface management offers good potential but users must have ultimate control and authority over it.
- Incorporate demonstrated human-computer interface (HCI) technology to supplement new integrated multi-media technology (don't throw out the baby with the bathwater). For example, human-like natural language I/O should be combined with proven techniques to best harness the full potential and power of the computer.
- Continue the development of the CUBRICON system. CUBRICON offers potential as a research testbed and may eventually lead to an interface system that can serve within an actual application.
- Perform research to better understand how to apply intelligent integrated human-computer interface technology for improved system effectiveness.
- Need faster computer (than the computer used for CUBRICON) to realize ultimate potential of human-like human-computer interfaces combining sophisticated graphics and natural language/speech I/O. Recognition of speech inputs and generation of system outputs must be as fast, or nearly as fast, as human-to-human dialogue.

16 Future Direction

(section forthcoming)

17 Summary

(section forthcoming)

18 References

[**Aceves84**] Aceves, J.J.G., Poggio, A. & Eliott, D. 1984. Research into multimedia message system architecture, Final Report. *Proj. No. 5363*, SRL International, Menlo Park, CA.

[**Andriole86**] Andriole, S.J. 1986. Graphic equivalence, graphic explanations, and embedded process modeling for enhanced process modeling for enhanced user-system interaction. *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 16, No. 6.

[**Arens88**] Arens, Y., Miller, L., & Sondheimer, N. 1988. Presentation Planning Using an Integrated Knowledge Base, *Architectures for Intelligent Interfaces: Elements and Prototypes*, J.W. Sullivan & S.W. Tyler (eds.), Addison-Wesley Pub. Co., pp. 93-108.

[**Bates78**] Bates, M. 1978. The theory and practice of augmented transition network frammars, in *Lecture Notes in Computer Science, vol 63, Natural Language Communication with Computers*, G. Goos, J. Jartmanis, & L. Bolc (eds.), Springer-Verlag, Berlin etc.

[**Bly86**] Bly, S.A., & Rosenberg, J.K. 1986. A comparison of tiled and overlapping windows. in *CHI'86 Proceedings*, ACM 0-89791-180-6/86/04000101, pages 101-106.

[**Carberry87**] Carberry, S. 1987. First international workshop on user modeling, *AI Magazine*, Vol.8, No.3, pages 71-74.

[**Carbonell85**] Carbonell, J.G., Boggs, W.M., Mauldin, M.L., & Anick, P.G. 1985. An integrated natural language interface, in *Applications in Artificial Intelligence*, S. Andriole (ed.), Petrocelli Books, Inc., pages 227-245.

[**CACM88**] *Communications of the ACM: Special Issue on HyperText* 1988. Vol. 31, No. 7.

[**Cheikes88**] Cheikes, B.A. & Webber, B.L. 1988. The Design of a Cooperative Respondent, *Architectures for Intelligent Interfaces: Elements and Prototypes*, J.W. Sullivan & S.W. Tyler (eds.), Addison-Wesley Pub. Co., pp. 3-18.

[**Conklin87**] Conklin, J. 1987. Hypertext: An introduction and survey, *Computer*, Vol. 20, No. 9, pages 17-41.

[**Feiner82**] Feiner, S., Nagy, S. & van Dam, A. 1982. An experimental system for creating & presenting interactive graphical documents, *Trans. Graphics*, Vol. 1, No. 1, pages 59-72.

[**Grice75**] Grice, H.P. 1975. Logic and conversation, in P. Cole & J.L. Morgan (eds.), *Syntax and Semantics, Vol. 3: Speech Acts*, Academic Press, pages 41-48.

[**Grosz78**] Grosz, B. J. 1978. Discourse analysis, in D. Walker (ed.), *Understanding Spoken Language*, Elsevier North-Holland, New York, pages 229-345.

[**Grosz81**] Grosz, B.J. 1981. Focusing and description in natural language dialogues, *Elements of Discourse Understanding*, A.Joshi, B. Webber, & I.Sag (eds.), Cambridge Univ. Press, pages 84-105.

[**Grosz85**] Grosz, B.J. & Sidner, C.L. 1985. Discourse structure and the proper treatment of interruptions, *Proc. of IJCAI*, pages 832-839.

[Grosz86] Grosz, B.J. 1986. The representation and use of focus in a system for understanding dialogs," in *Readings in Natural Language Processing*, B.J. Grosz, K.S. Jones, B.L. Webber (eds.), Morgan Kaufmann Pulishers, pages 353-362.

[Haller] Haller, S.M. 1989. Technical report: spatial relations and locative phrase generation in a map context, Computer Science Department, State University of New York at Buffalo.

[Herkovits85] Herkovits, A. 1985. Semantics and pragmatics of locative expressions, in *Cognitive Science, vol 9*, pages 241-378.

[Hilton87] Hilton, M.L. 1987. Design and implementation of the MAP DISPLAY SYSTEM, RADC Report.

[Hollan88] Hollan, J., Miller, J.R., Rich, E., & Wilner, W. 1988. Knowledge Bases and Tools for Building Integrated Multimedia Intelligent Interfaces, *Architectures for Intelligent Interfaces: Elements and Prototypes*, J.W. Sullivan & S.W. Tyler (eds.), Addison-Wesley Pub. Co., pp. 19-38.

[Kass88] Kass, R. & Finin, T. 1988. General User Modelling: A Facility to Support Intelligent Interaction, *Architectures for Intelligent Interfaces: Elements and Prototypes*, J.W. Sullivan & S.W. Tyler (eds.), Addison-Wesley Pub. Co., pp. 169-184.

[Kaplan82] Kaplan, S.J. 1982. Cooperative Responses from a Portable Natural Language Database Query System, *Computational Models of Discourse*, M. Brady (ed.), The MIT Press.

[Katz84] Katz, A. 1984. An experimental internetwork multimedia mail system, *Proc. IFIP 6.5 Working Conf. Computer Message Services*, Nottingham, England.

[Kobsa86] Kobsa, A., Allgayer, J., Reddig, C., Reithinger, N., Schmauks, D., Harbusch, K., & Wahlster, W. 1986. Combining Deictic Gestures and Natural Language for Referent Identification, *Proc. of the 11th International Conference on Computational Linguistics*, Bonn, FR Germany.

[Maida85] Maida, A.S. & Shapiro, S.C. 1985. Intensional concepts in propositional semantic networks," in *Readings in Knowledge Representation*, R.J. Brachman & H.J. Levesque (eds.), Morgan Kaufmann Pub., pages 169-190.

[Mason88] Mason, J.A. & Edwards, J.L. 1988. Explicit Models in Intelligent Interface Design, *Architectures for Intelligent Interfaces: Elements and Prototypes*, J.W. Sullivan & S.W. Tyler (eds.), Addison-Wesley Pub. Co., pp. 151-168.

[Martin73] Martin, J. *Design of man-computer dialogues.* Prentice Hall

[McKay80] McKay, D.P. & Shapiro, S.C. 1980. MULTI - A LISP based multiprocessing system, *Conference Record of the 1980 LISP Conference*, Stanford Univ., pages 29-37.

[Miller56] Miller, G.A. 1956. The magical number seven plus or minus two, *Psychological Review*, 63, pages 81-97.

[Neal86] Neal, J.G. & Shapiro, S.C. 1986. Knowledge representation for reasoning about language, in *The Role of Language in Problem Solving*, J.C. Boudreauz, B.W. Hamill, & R. Jernigan (eds.), Springer-Verlag Pub., pages 27-47.

[**Neal87**] Neal, J.G. & Shapiro, S.C. 1987. Knowledge based parsing, in *Natural Language Parsing Systems*, L. Bolc (ed.), Springer-Verlag Pub., pages 49-92.

[**Neal88a**] Neal, J.G., Bettinger, K.E., Byoun, J.S., Dobes, Z., & Thielman, C.Y. 1988. An Intelligent Multi-Media Human-Computer Dialogue System, *Proceedings of the Workshop on Space Operations, Automation, and Robotics (SOAR-88)*, Wright State University, Dayton, OH.

[**Neal88b**] Neal, J.G., Dobes, Z., Bettinger, K.E., & Byoun, J.S. 1988. Multi-Modal References in Human-Computer Dialogue; *Proc. AAAI-88*, pages 819-823.

[**Neches86**] Neches, R. & Kaczmarek, T. 1986. *AAAI-86 Workshop on Intelligence in Interfaces*, USC/Information Sciences Institute, August, 1986.

[**Potash77**] Potash, L.M. 1977. Design of maps and map-related research, *Human Factors 19(2)*, pages 139-150.

[**Press86**] Press, B. 1986. The U.S. Air Force TEMPLAR project status and outlook, *Western Conf, on Knowledge-Based Engineering and Expert Systems*, Anaheim, CA, pages 42-48.

[**Reithinger87**] Reithinger, N. 1987. Generating Referring Expressions and Pointing Gestures, *Natural Language Generation*, G. Kempen (ed.), Dordrecht: Nijhoff, pp. 71-81.

[**Roth88**] Roth, S., Matthis, J., & Mesnard, X. 1988. Graphics and Natural Language as Components of Automated Explanation, *Architectures for Intelligent Interfaces: Elements and Prototypes*, J.W. Sullivan & S.W. Tyler (eds.), Addison-Wesley Pub. Co., pp. 109-128.

[**Shapiro75**] Shapiro, S.C. 1975. Generation as parsing from a network into a linear string, *AJCL*, Microfiche 33, pages 45-62.

[**Shapiro79a**] Shapiro, S.C. 1979a. The SNePS semantic network processing system, in Findler, ed., *Associative Networks - The Representation and Use of Knowledge by Computers*, Academic Press, New York, pages 179-203.

[**Shapiro79b**] Shapiro, S.C. 1979b. Using non-standard connectives & quantifiers for representing deduction rules in a semantic network, invited paper presented at *Current Aspects of AI Research*, a seminar held at the Electrotechnical Laboratory, Tokyo.

[**Shapiro79c**] Shapiro, S.C. 1979c. Numerical quantifiers and their use in reasoning with negative information, *Proc. IJCAI*, pages 791-796.

[**Shapiro89**] Shapiro, S.C. & the SNePS Implementation Group 1989. *SNePS-2.1 User's Manual*, C.S. Dept., SUNY at Buffalo, N.Y.

[**Shapiro82a**] Shapiro, S.C. 1982. Generalized augmented transition network grammars for generation from semantic networks, *AJCL*, Vol. 8, No. 1, pages 12-25.

[**Shapiro82b**] Shapiro, S.C., Martens, J., & McKay, D. 1982. Bi-directional inference, *Proc. of the Cognitive Science Society*, pages 90-93.

[**Shapiro82c**] Shapiro, S.C. & Neal, J.G. 1982. A knowledge engineering approach to natural language understanding, *Proc. ACL*, pages 136-144.

[**Shapiro86**] Shapiro, S.C. & Rapaport, W. 1986. SNePS considered as a fully intensional propositional semantic network, *Proc. AAAI-86*, pages 278-283; in G. McCalla & N. Cercone (eds.) *Knowledge Representation*, Springer-Verlag Pub. *item [[Shapiro87]] Shapiro, S.C. & Rapaport, W. 1987. SNePS considered as a fully intensional propositional semantic network, in* The Knowledge Frontier, Essays in the Representation of Knowledge, *N. Cerone & G. McCalla (eds.), Springer-Verlag, pages 263-315.*

[**Sidner83**] Sidner, C.L. 1983. Focusing in the comprehension of definite anaphora, in M. Brady & R.C. Berwick (eds.), *Computational Models of Discourse*, The MIT Press, pages 267-330.

[**Simmons72**] Simmons, R. & Slocum, J. 1972. Generating English discourse from semantic networks, *CACM*, 15:10, pages 891-905.

[**Smith86**] Smith, S.L. & Mosier, J.N. 1986. Guidelines for Designing User Interface Software, ESD-TR-86-278, August, 1986.

[**Sullivan88**] Sullivan, J.W. & Sherman, W.T. (eds.) 1988. *Architectures for Intelligent Interfaces: Elements and Prototypes*, Addison-Wesley Pub. Co.

[**Thomas85**] Thomas, R.H., Forsdick, H.C., Crowley, T.R., Schaaf, R.W., Tomlinson, R.S., Travers, V.M., & Robertson, G.G. 1985. Diamond: a multimedia message system built on a distributed architecture, *IEEE Computer*, December, pages 65-78.

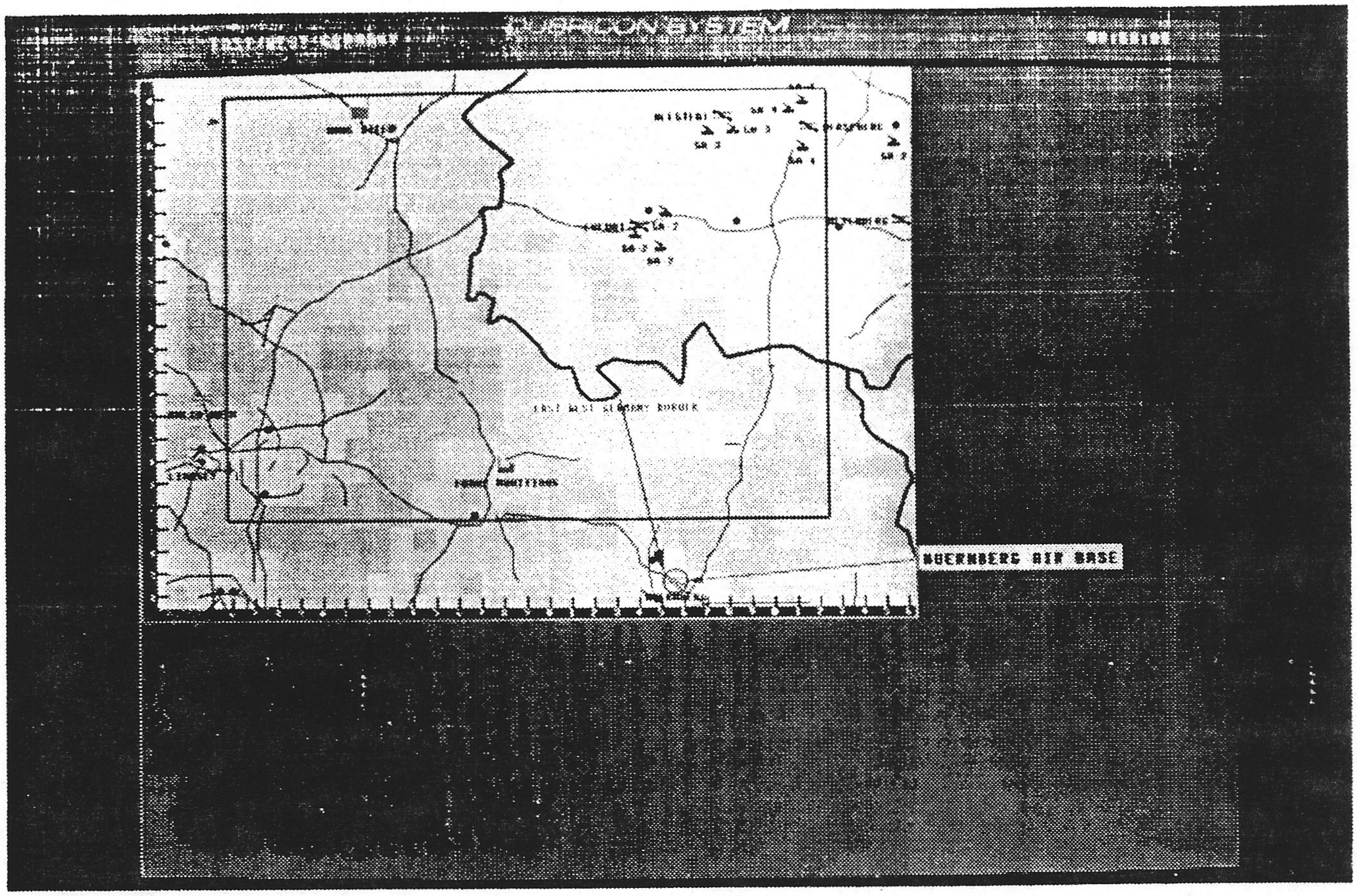
[**Wahlster88**] Wahlster, W. 1988. User and Discourse Models for Multimodal Communication, *Architectures for Intelligent Interfaces: Elements and Prototypes*, J.W. Sullivan & S.W. Tyler (eds.), Addison-Wesley Pub. Co.

[**Young88**] Young, R.L. 1988. A Dialogue Model for User Interfaces, *Architectures for Intelligent Interfaces: Elements and Prototypes*, J.W. Sullivan & S.W. Tyler (eds.), Addison-Wesley Pub. Co., pp. 39-54.

APPENDIX A
EXAMPLE CUBRICON DIALOGUE

This appendix contains examples of dialogue between a human user and CUBRICON. These examples were selected to illustrate key CUBRICON capabilities and include both verbal dialogue and associated graphics.

A more complete and extensive set of examples will be included in the final version of this report.



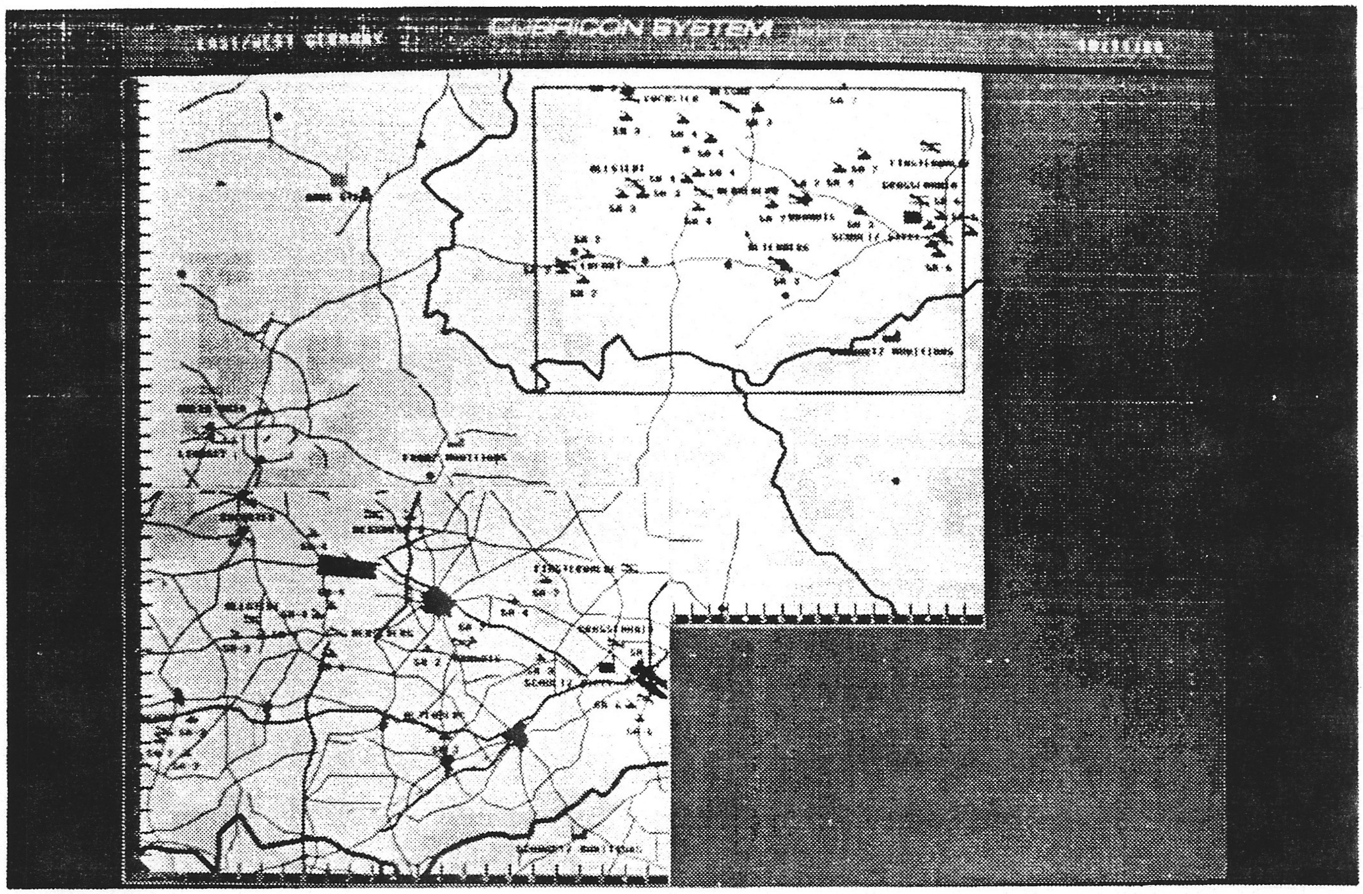
Entities in the Expanded Region

Item	Disposition	Latitude	Longitude	Name	Mobility
air base	friendly	50.300N	8.390E	Rhein Main	--
air base	friendly	50.050N	8.330E	Lindsey	--
air base	enemy	51.400N	11.460E	Allstedt	--
air base	enemy	50.970N	10.960E	Erfurt	--
air base	enemy	51.350N	11.960E	Merseberg	--
air base	enemy	50.980N	12.510E	Altenberg	--
air base	friendly	49.550N	11.150E	Nuernberg	--
SA-2	enemy	50.933N	10.933E	--	low
SA-2	enemy	51.016N	11.116E	--	low
SA-2	enemy	50.883N	11.083E	--	low
SA-2	enemy	51.283N	12.466E	--	low
SA-3	enemy	51.338N	11.516E	--	high
SA-3	enemy	51.335N	11.366E	--	high
SA-4	enemy	51.266N	11.921E	--	low
SA-4	enemy	51.416N	11.638E	--	low
SA-4	enemy	51.450N	11.921E	--	low
plant	friendly	51.421N	9.344E	Hans Steel	--
factory	friendly	49.991N	10.152E	Franz Munitions	--

Its location is 30 miles southwest of the East-West Germany border .
=>> Where is the Nuernberg airbase?
The map on the color graphics screen is being expanded to include the Nuernberg air base.
Its location is 50 miles southeast of the East-West Germany border .

The corresponding table is being generated.

The corresponding table is now on the monochrome screen.
=>>



air base	enemy	50 970N	10.960E	Erfurt	--
air base	friendly	49.550N	11.160E	Nuernberg	--
air base	enemy	61.600N	13.610E	Finsterwalde	--
air base	enemy	61.100N	13.700E	Dresden	--
air base	enemy	61.310N	13.630E	Grossenhain	--
SA-2	enemy	61.916N	12.983E	--	low
SA 2	enemy	51.016N	11.116E	--	low
SA-2	enemy	61.316N	12.716E	--	low
SA-2	enemy	61.900N	11.416E	--	low
SA-2	enemy	60.933N	10.933E	--	low
SA-2	enemy	61.283N	12.466E	--	low
SA-2	enemy	60.883N	11.083E	--	low
SA-2	enemy	61.550N	13.133E	--	low
SA-3	enemy	60.950N	12.566E	--	high
SA-3	enemy	61.760N	11.400E	--	high
SA-3	enemy	61.800N	12.366E	--	high
SA-3	enemy	61.335N	11.366E	--	high
SA-3	enemy	61.338N	11.616E	--	high
SA-3	enemy	61.250N	13.100E	--	high
SA-4	enemy	61.450N	11.921E	--	low
SA-4	enemy	61.266N	11.921E	--	low
SA-4	enemy	61.633N	12.016E	--	low
SA-4	enemy	61.473N	12.956E	--	low
SA-4	enemy	61.416N	11.838E	--	low
			11.816E	--	low
			13.666E	--	high
			13.616E	--	high
			13.905E	--	high
			13.600E	--	high
			13.683E	--	high

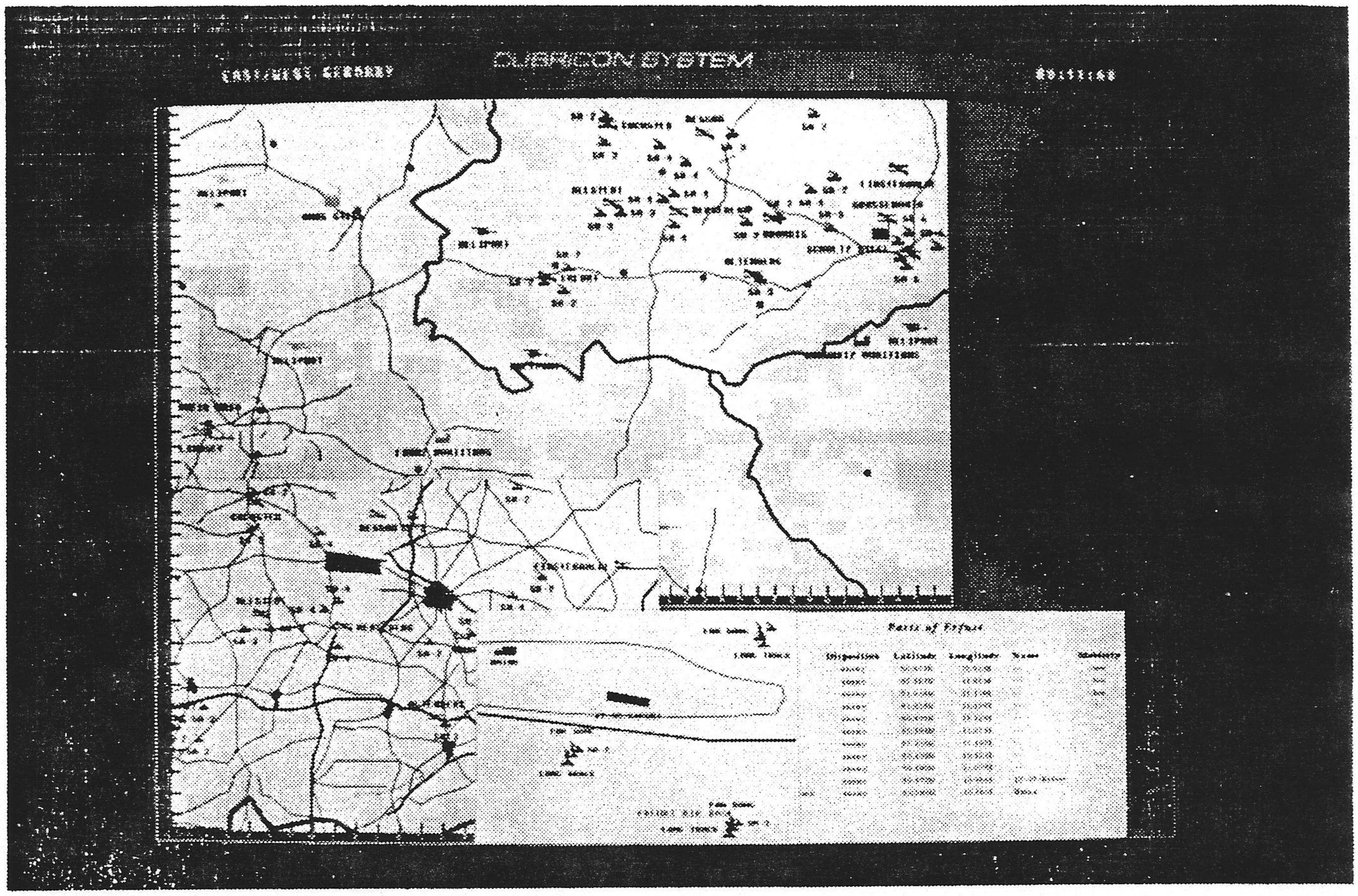
Entities in the Region

Item	Disposition	Latitude	Longitude	Name	Mobility
air base	enemy	61.400N	11.460E	Allstedt	--
air base	enemy	61.310N	13.530E	Grossenhain	--
air base	enemy	61.600N	13.610E	Finsterwalde	--
air base	enemy	61.100N	13.700E	Dresden	--
air base	enemy	60.970N	10.960E	Erfurt	--
air base	enemy	61.350N	11.960E	Merseberg	--
air base	enemy	50.980N	12.510E	Altenberg	--
air base	enemy	61.860N	11.430E	Cochsted	--
air base	enemy	61.310N	12.660E	Brandis	--
air base	enemy	51.810N	12.180E	Dessau	--
SA-2	enemy	61.016N	11.116E	--	low
SA-2	enemy	61.316N	12.716E	--	low
SA-2	enemy	60.883N	11.083E	--	low
SA-2	enemy	60.933N	10.933E	--	low
SA-2	enemy	61.283N	12.466E	--	low
SA-2	enemy	61.550N	13.133E	--	low

A table containing the locations is being presented on the monochrome screen.
=>> Zoomin on this point ↓6(700 184 "Guide Window 3" 1901761).
The zoomed in area is being presented on the color graphics screen.

The corresponding table is being generated.

The corresponding table is now on the monochrome screen.
=>>



PACKAGE WORKSHEET

PKG#	0026	Preparer's Name		Date Prepared		Priority	

OFFENSIVE COUNTER AIR MISSIONS / **PRE-TARGET REFUELING**

Mission	OCA#	Origin	TOD	#AC	AC Type	SCL	AC Pool	SVC#	STN#	Start	Dur.	Disbur.
1	345	Rhein Main Air Base	05:45				49tfw-F-1					
2	445		06:00				45tfw-Ef-					
3												
4												

TARGET STRIKE MISSION / **POST-TARGET REFUELING**

Mission	Aim Point	TOT	SVC#	STN#	Start	Dur.	Disbur.
1	6-24-Merseberg Runway	06:50	345	244	07:25	00:20	21960 lbs
2	3-21-Dresden Runway	07:02	445	244	07:45	00:20	20942 lbs
3							
4							

REFUELING MISSION

RFL#	345	TOD	07:00	AC Type	Kc-135	Load		Origin	Lindsey Air Base

Station	STN#	Start Time	Stop Time	Orbit Location
1	244	07:20	07:55	50.348 N Latitude, 11.692 E Longitude
2				

AIR ESCORT MISSIONS

Mission	AEM#	Origin	TOD	#AC	ACT	SCL	Remarks
1							
2							

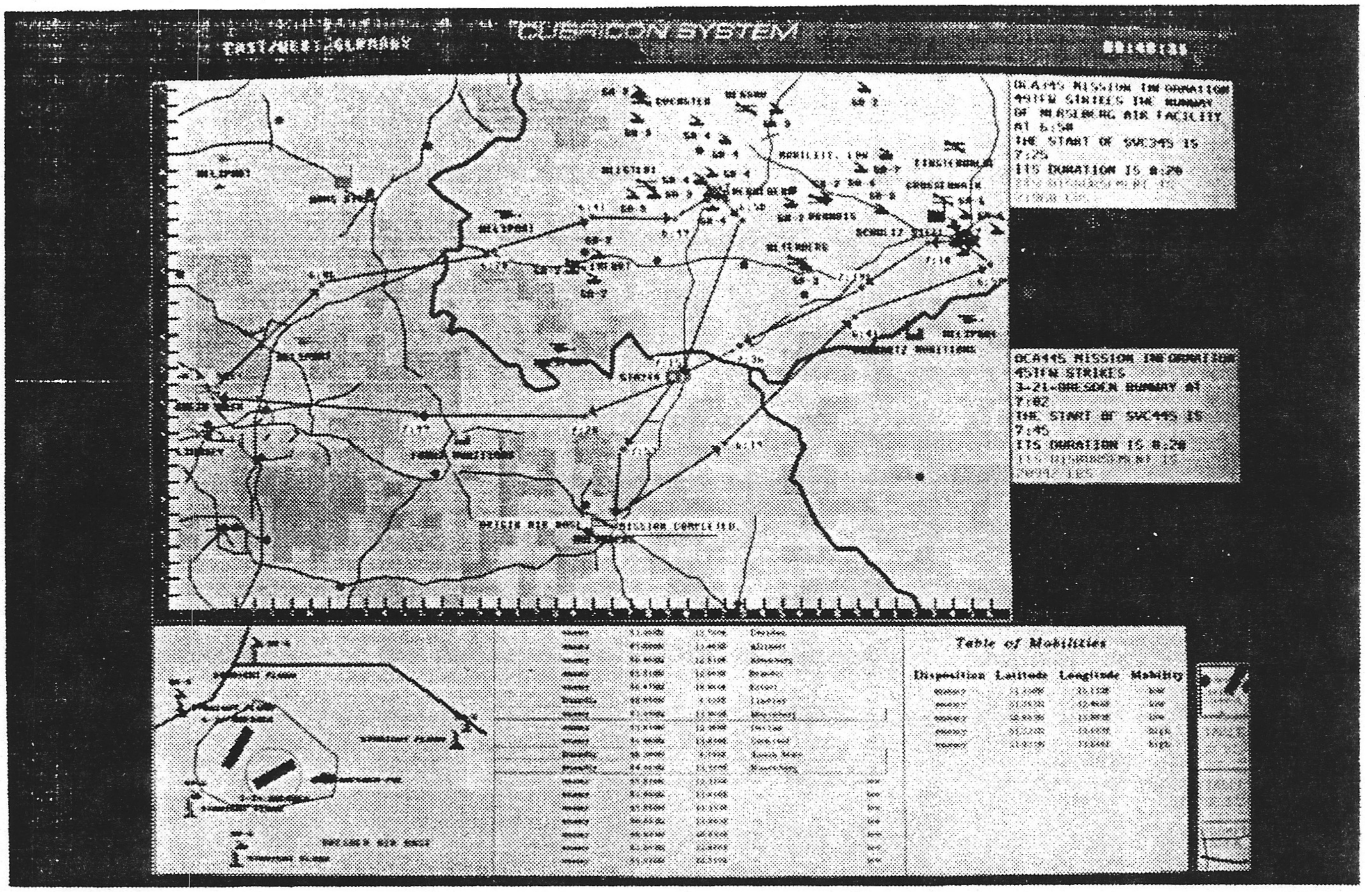
SAM SUPPRESSION MISSIONS

Mission	SSM#	Origin	TOD	#AC	ACT	SCL	Target	TOT
1								
2								

=>> What are the aimpoints within the Erfurt airbase?
Look at the color graphics screen. The aimpoints within Erfurt are being presented.

The corresponding table is being presented on the color screen.
=>> What are the aimpoints within the Dresden airbase?
Look at the color graphics screen. The aimpoints within Dresden are being presented.

The corresponding table is being presented on the color screen.
=>>



PACKAGE WORKSHEET

PKG#	0026	Preparer's Name		Date Prepared		Priority	

OFFENSIVE COUNTER AIR MISSIONS

Mission	OCA#	Origin	TOD	#AC	AC Type	SCL	AC Pool
1	345	Rhein Main Air Base	05:45				49tfw-F-1
2	445	Nuernberg Air Base	06:00				45tfw-Ef-
3							
4							

PRE-TARGET REFUELING

SVC#	STN#	Start	Dur.	Disbur.

TARGET STRIKE MISSION

Mission	Aim Point	TOT
1	6-24-Merseberg Runway	06:50
2	3-21-Dresden Runway	07:02
3		
4		

POST-TARGET REFUELING

SVC#	STN#	Start	Dur.	Disbur.
345	244	07:25	00:20	21960 lbs
445	244	07:45	00:20	20942 lbs

REFUELING MISSION

RFL#	345	TOD	07:00	AC Type	Kc-135	Load		Origin	Lindsey Air Base

Station	STN#	Start Time	Stop Time	Orbit Location
1	244	07:20	07:55	50.348 N Latitude, 11.692 E Longitude
2				

AIR ESCORT MISSIONS

Mission	AEM#	Origin	TOD	#AC	ACT	SCL	Remarks
1							
2							

SAM SUPPRESSION MISSIONS

Mission	SSM#	Origin	TOD	#AC	ACT	SCL	Target	TOT
1								
2								

=>> Present the OCA345 and OCA445 mission plans.
Look at the color graphics screen. The mission plans are being presented.
Oca345 is a submission of PKG0026 .
49tfw departs Rhein Main air facility at 5:45.
Stk345, And Svc345 are submissions of it.
Oca445 is a submission of PKG0026 .
45tfw departs Nuernberg air facility at 6:00.
Stk445, And Svc445 are submissions of it.

APPENDIX B
CUBRICON GRAMMAR AND LEXICON

This section contains a description of the CUBRICON grammar and lexicon. The grammar represents the structure of natural language which must be used when making inputs to CUBRICON. The lexicon includes a listing of the lexicon that is understood by CUBRICON. This lexicon is in two parts. First, is a complete alphabetical list of the lexicon. Second is a listing of CUBRICON vocabulary by word type.

These representations of the CUBRICON grammar and lexicon were developed for, and used to support the CUBRICON evaluation. They are also, included in Appendix E.

I don't understand this sentence.

CUBRICON GRAMMAR

Key:

Notation	Meaning	Examples
[abcdefg]	Type of grammar or vocabulary element	[question], [noun]
(abcdef)	Example of grammar or vocabulary element	(what is this), (airbase)
abcdef	Emglish word that is component of element	what, this
/	Or	

n

Grammar Definition:

Grammar Structure	Definition	Examples
[sentence]	[question] / [imperitive] / [declarative]	(where is the Dresdin Airbase)? (display the FG region). (its name is OCA123).
[question]	where [be-verb] [noun-phrase]	(where) (is) (the Dresden Airbase)?
	/ what [be-verb] [noun-phrase]	(what) (is) (this)?
	/ what [be-verb] [noun-phrase] [locative]	(what) (are) (the aimpoints) (within the Dresden Airbase)?
	/ what [be-verb] [noun-phrase] [prepositional- / complement]	(what) (is) (the mobility) (of this [mouse-click])?
	/ what [noun] [be-verb] [locative]	(what) (ac pools) (are) (at the Nuernberg Airbase)?
	/ [be-verb] [demonstrative] [noun-phrase]	(is) (this [mouse-click]) (a sam)?
[declarative]	[noun phrase] [be-verb] [noun phrase]	(its name) (is) (OCA123).
	/ [noun phrase] [verb-group] [noun-phrase]	(I) (am starting) (a new mission plan).
	/ [prepositional-complement] [noun-phrase] [be-verb] [noun-phrase]	(for OCA123) (the origin) (is) (the Nuernberg Airbase).
[imperitive]	[command-verb] [demonstrative]	(remove) (this window [mouse-click]).
	/ [command-verb] [demonstrative] [locative]	(enter) (this [mouse-click]) (here [mouse-click]).
	/ [command-verb] [demonstrative] [noun-phrase]	(make) (this [mouse-click]) (the current package).
	/ [command-verb] [noun-phrase]	(display) (the FG region).
	/ [command-verb] [noun-phrase] [locative]	(enter) (20 minutes) (here [mouse-click]).
	/ [command-verb] [noun-phrase] [noun-phrase]	(make) (PKG0026) (the current package).
	/ [command-verb] [noun-phrase] [prepositional-complement]	(plan) (a flight path) (for OCA345).
	/ [command-verb] [prepositional-complement]	(zoom-in) (on this point [mouse-click]).

I thought any NP could contain zero or more pointings

Grammar Element Definition:

Grammar Element	Definition	Examples
[noun-phrase]	/ [det] [noun]	(the) (threats)
	/ [det] [proper-name-group] [noun]	(the) (Fulda Gap) (region)
	/ [det] [noun] [proper-name]	(the) (ac-pool) (45TFW-F-111F)
	/ the value [proper-name]	(the) (value) (KC135)
	/ the value [number] [unit]	(the) (value) (21960) (lbs)
	/ [mouse-click-list]	("sequential mouse-clicks on visible objects")
	/ [proper-name]	(STN244)
	/ [quantifier] [noun]	(any) (sams)
	/ [time]	(7:45)
	/ [duration]	(20 minutes)
[proper-name-group]	[proper-name]	(OCA123)
	/ [proper-name] [conjunction] [proper-name-group]	(OCA123) (and) (OCA345)
[demonstrative]	this [mouse-click]	(this) ("mouse-click on visible object")
	/ this [noun] [mouse-click]	(this) (sam) ("mouse-click on visible object")
	/ these [mouse-click-list]	(these) ("sequential mouse-clicks on visible objects")
[locative]	here [mouse-click]	(here) ("mouse-click on location")
	/ [locative preposition] [noun-phrase]	(around) (the Dresdin airbase)
[prepositional-complement]	[preposition] [noun-phrase]	(for) (OCA123 and OCA345)
	[preposition] [demonstrative]	(on) (this point "mouse-click on location")
[verb-group]	[auxiliary] [main-verb]	(will) (depart)
[be-verb]	is / are	
[command-verb]	assign / blink / call / display / enter / expose / highlight / list / make / plan / present / remove / zoom-in	
[auxiliary]	am / will	
[main-verb]	starting / depart / strike	
[det]	a / an / the	
[quantifier]	some / any	
[conjunction]	and	
[locative-preposition]	around / at / in / near / on / within	

spec, pron?

[preposition]	as / by / for / from / of / to	
[mouse-click]	[click any mouse button on any visible object or location on either the monochrome or color display]	("mouse-click on table entry")
[time]	[hours]:[minutes]	(6):(45)
[duration]	[number] minutes	(20) (minutes)
[noun]	- see attached list of nouns and proper names - / [pronoun]	radar
[pronoun]	it	
[proper-name]	- see attached list of nouns and proper names -	Nuernberg

Vocabulary Quick Reference

Threats

airbase
(airbases)
(artillery)
Dresden airbase
Erfurt airbase
sam
sams
(tank battalion(s))
(threat)
threats

Targets (other than threats)

(aimpoint)
aimpoints
factory
(factories)
Franz Munitions
(fuel tank(s))
(fuel storage tank(s))
(Hans Steel)
(heliport)
heliports
(munition factory)
(munition factories)
(population center(s))
(radar(s))
(Schultz Steel)
(Schwartz Munitions)
(steel plant(s))
target
(targets)

Cities

Allstedt
(Altenberg)
(Brandis)
(Cochsted)
(Dessau)
Dresden
Erfurt
(Findsterwalde)
(Leipzig)
Lindsey
Merseberg
Stargard

Mission Descriptors

(aimpoint)
(air escort mission)
aimpoints
(aircraft unit(s))
(current mission plan)
(current package)
(disbursement)
duration
(flight path(s))
mission
(missions)
(mission-plan)
mission plans
OCA
(OCA mission plan(s))
(OCA plan(s))
(orbit location(s))
package
(packages)
(package number)
(PKG@@@)
(PKG@@@@)
(post-target refueling mission(s))
(pre-target refueling mission(s))
(refueling mission(s))
(RFL@@@)
RFL345
(sam suppression mission(s))
(SSM@@@)
STN
(STN number)
(STN mission plans)
(STN plan(s))
(STN@@@)
STN002
STN244
(strike date)
SVC
(SVC mission plan(s))
(SVC@@@)
SVC001
SVC345
SVC445
(SVC@@@@)
(TOD)
(TOT)
(TSM@@@)

Geographic Descriptors

East-West Germany
EW Germany
FG
Fulda Gap
(locations)
map
(orbit location(s))
point
(population center(s))
region

Attributes of Geographic Objects

ceiling
(ceilings)
(disposition)
location
(locations)
(mobilities)
mobility
name
(names)
nationaly
(nationalities)
type

Friendly Assets

(34TFS)
(34TFS-F-15C)
45TFW-EF-111E
(45TFW-F-4G)
(45TFW-F-16C)
(45TFW-F-16D)

49TFW-F-111F
(49TFW-F-16C)
(435TAW)
(45TFW)
49TFW
a
ac-pool
(ac-pools)
airbase
(airbases)
(aircraft)
(aircraft unit(s))
(EF-111E)
(F-4G(s))
(F-16D(s))
(F-111F(s))
(F-16C(s))
(F-15C(s))
fighter
(fighters)
(fighter base(s))
heliport
KC-135
(Leipsig) airbase
(missile(s))
Nuernberg airbase
(orbit location(s))

Preposition

as
by
for
from
of
to

Display Descriptors

color graphics display
form
forms
forms window
map
(mission plan form)

Command

(OCA form(s))
(slot(s))
table
window
(windows)

Auxiliary

am
will

Determiner

a
an
the

Quantifier

some
any Rhein Main

Conjunction

and

Pronoun

it

Be Verb

are
is

Verb

assign
blink
call
display
enter
expose
highlight
list
make
plan
present
remove
zoom-in

Main Verb

starting
depart
strike

Locative

around
at
in
near
on
within

Proper Names

(34TFS)
(34TFS-F-15C)
45TFW-EF-111E
(45TFW-F-4G)
(45TFW-F-16C)
(45TFW-F-16D)
49TFW-F-111F
(49TFW-F-16C)
(435TAW)
(45TFW)
49TFW
(AEM@@@)
Allstedt
(Altenberg)
(Brandis)
(Cochsted)
Dresden
(EF-111E)
East-West Germany
Erfurt
EW Germany (F-4G(s))
(F-16D(s))
(F-111F(s))
(F-16C(s))
(F-15C(s))
FG (Findsterwalde)
Franz Munitions
Fulda Gap
(Grossenhain)
(Hans Steel)
(Hans Steel)
KC-135
(Leipzig)
Lindsey
Merseberg
Nuernberg
(OCA@@@)
OCA345
OCA445
OCA555
(PKG@@@)
(PKG@@@@)
(RFL@@@)
RFL345
Rhein Main
(Schultz Steel)
(Schwartz Munitions)
(SSM@@@)
Stargard
STN
(STN number)
(STN mission plans)
(STN plan(s))
(STN@@@)
STN002
STN244
SVC
(SVC mission plan(s))
(SVC@@@)
SVC001
SVC345
SVC445
(SVC@@@@)
(TSM@@@)

Nouns

ac-pool
(ac-pools)
(ac-type)
(aimpoint)
aimpoints
airbase
(airbases)
(aircraft)
(aircraft unit(s))
(artillery)
battalion
battalions
(color)
color graphics display
(current mission plan)
(current package)
(disbursement)
duration
factory
(factories)
fighter
(fighters)
(fighter base(s))
flight
(flight path(s)) form
forms
forms window
fuel
(fuel tank(s))
(fuel storage tank(s)) (heliport)
heliports
(hour)
location
(locations)
map
(minute)
minutes
(missile(s))
mission
(missions)
(mission-plan)
(mission plan form)
mission plans
(mobilities)
mobility
(munition)
munitions
(munition factory)
(munition factories)
name
(names)
nationality
(nationalities)
(number)
OCA
(OCA form(s))
(OCA mission plan(s))
(OCA plan(s))
(orbit location(s))
(origin)
package
(packages)
(package number)
plan
plans
(plant(s))
point
(population center(s))
(post-target refueling mission(s))
(pre-target refueling mission(s))
(preparer)
(priority)
(radar(s))
(refueling mission(s))
region
regions
sam
(sam suppression mission(s))
sams
(slot(s))
start
(STN number)
(STN mission plans)
(STN plan(s))
(strike date)
(SVC mission plan(s))
table
(tables)
tank
tanks
(tank battalion(s))
target
(targets)
(target strike mission)
(target strike missions)
(task(s))
(threat)
threats
time
(TOD)
(TOT)
type
(types)
unit
units
value
window
(windows)

Vocabulary List

Vobabulary Word	Comment
(34TFS)	
(34TFS-F-15C)	
45TFW-EF-111E	
(45TFW-F-4G)	
(45TFW-F-16C)	
(45TFW-F-16D)	
49TFW-F-111F	
(49TFW-F-16C)	
(435TAW)	
(45TFW)	
49TFW	
a	
ac-pool	
(ac-pools)	
(ac-type)	
(AEM@@@)	
(aimpoint)	
aimpoints	
(air escort mission)	
airbase	
(airbases)	
(aircraft)	
(aircraft unit(s))	
Allstedt	
am	
an	
and	
any	
are	
around	
(artillery)	
(as)	
assign	
at	
(Altenberg)	
battalion	
battalions	
blink	
(Brandis)	
by	
(call)	
ceiling	
(ceilings)	
(Cochsted)	
(color)	
color graphics display	
current	
(current mission plan)	All words available individually in voice system
(current package)	All words available individually in voice system
(depart)	

(Dessau)	
display	
(disposition)	
(disbursement)	
Dresden	
duration	
(EF-111E)	
East-West Germany	Text accompanying this voice input is ew-germany
enter	
expose	
Erfurt	
EW Germany	Short for East-West Germany.
(F-4G(s))	
(F-16D(s))	
(F-111F(s))	
(F-16C(s))	
(F-15C(s))	
factory	
(factories)	
FG	Short for Fulda Gap, a region that can be displayed by CUBRICON.
fighter	
(fighters)	
(fighter base(s))	
(Findsterwalde)	
flight	
(flight path(s))	All words available individually in voice system.
for	
form	
forms	
forms window	All words available individually in voice system.
Franz Munitions	
from	
fuel	
(fuel tank(s))	All words available individually in voice system.
(fuel storage tank(s))	All words available individually in voice system.
Fulda Gap	
(Grossenhain)	
(Hans Steel)	
(heliport)	
heliports	
here	
highlight	
(hour)	
in	
is	
it	
(its)	
KC-135	
(Leipzig)	
Lindsey	
list	
location	
(locations)	
make	

map	
Merseberg	
(minute)	
minutes	
(missile(s))	
mission	
(missions)	
(mission-plan)	All words available individually in vioce system.
(mission plan form)	All words available individually in vioce system.
mission plans	
(mobilities)	
mobility	
(munition)	
munitions	
(munition factory)	All words available individually in voice system.
(munition factories)	
name	
(names)	
nationality	
(nationalities)	
near	
(number)	
Nuernberg	
OCA	
(OCA form(s))	All words available individually in voice system.
(OCA mission plan(s))	
(OCA plan(s))	
(OCA@@@)	Any three digit number is acceptable.
OCA345	
OCA445	
OCA555	
of	
on	
open	
(orbit location(s))	
(origin)	
package	
(packages)	
(package number)	
(PKG@@@)	Any three digit number is acceptable.
(PKG@@@@)	Any three digit number is acceptable.
plan	Must be used as part of phrase (e.g., OCA plan).
plans	Must be used as part of phrase (e.g., OCA plans).
(plant(s))	
point	
(population center(s))	
(post-target refueling mission(s))	
(pre-target refueling mission(s))	
(preparer)	
present	Understood as a verb.
(priority)	
(radar(s))	
refueling	Must be used as part of phrase (e.g., refueling mission).
(refueling mission(s))	All words available individually in voice system.

(RFL@@@)	Any three digit number is acceptable.
RFL345	
region	
regions	
remove	
Rhein Main	
sam	
(sam suppression mission(s))	
sams	
(Schultz Steel)	
(Schwartz Munitions)	
(slot(s))	
(some)	
(SSM@@@)	Any three digit number is acceptable.
Stargard	
start	
starting	
(steel plant(s))	
STN	Must be used as part of a phrase (e.g., STN plan).
(STN number)	
(STN mission plans)	
(STN plan(s))	
(STN@@@)	Any three digit number is acceptable.
STN002	
STN244	
strike	
(strike date)	
SVC	
(SVC mission plan(s))	All words available individually in voice system.
(SVC@@@)	Any three digit number is acceptable.
SVC001	
SVC345	
SVC445	
(SVC@@@@)	Any four digit number is acceptable.
table	
(tables)	
tank	
tanks	
(tank battalion(s))	All words available individually in voice system.
target	
(targets)	
(target strike mission)	All words available individually in voice system.
(target strike missions)	
(task(s))	
the	
these	
this	
(threat)	
threats	
time	
to	
(TOD)	
(TOT)	
(TSM@@@)	Any three digit number is acceptable.

type
(types)
unit
units
value
(was)
what
where
window
(windows)
will
within
zoomin

APPENDIX C
GRAPHIC REPRESENTATION OF THE TASK DOMAIN KNOWLEDGE BASE

This appendix contains a complete graphic representation of the CUBRICON Task Domain Knowledge Base. It shows all resident task domain objects with interlinking arcs which define their semantic structures and relationships.

The final version of this report will include a finalized version of these figures with more complete indexing.

APPENDIX D
SOFTWARE DOCUMENTATION FOR PRIMARY FUNCTIONS

This section will contain descriptions of software developed to implement major system functions. These descriptions will be included in the final version of this report.

APPENDIX E
EVALUATION TRAINING MATERIAL AND DATA

This section contains material used to support the CUBRICON evaluation and also contains data generated during the evaluation. The CUBRICON evaluation support material includes training material and CUBRICON work aids.

E.1 TRAINING MATERIAL AND WORK AIDS

The following material was used to train the Evaluators in the use of CUBRICON, the evaluation goals, and procedures. The following training materials are included (those marked with an astericks also served as work aids during the evaluation itself):

Voice Training List (used to guide Evaluators through voice recognition system training).

Breakdown of Mission Planning Package.

List of Useful Miscellaneous Commands.

Training Outline.

*CUBRICON grammar.

*Vocabulary Quick Reference (CUBRICON vocabulary organized by word type for easy access.

*Vocabulary List (complete alphabetical list of vocabulary words).

*Example of Form (shows areas of the form that can accept input)

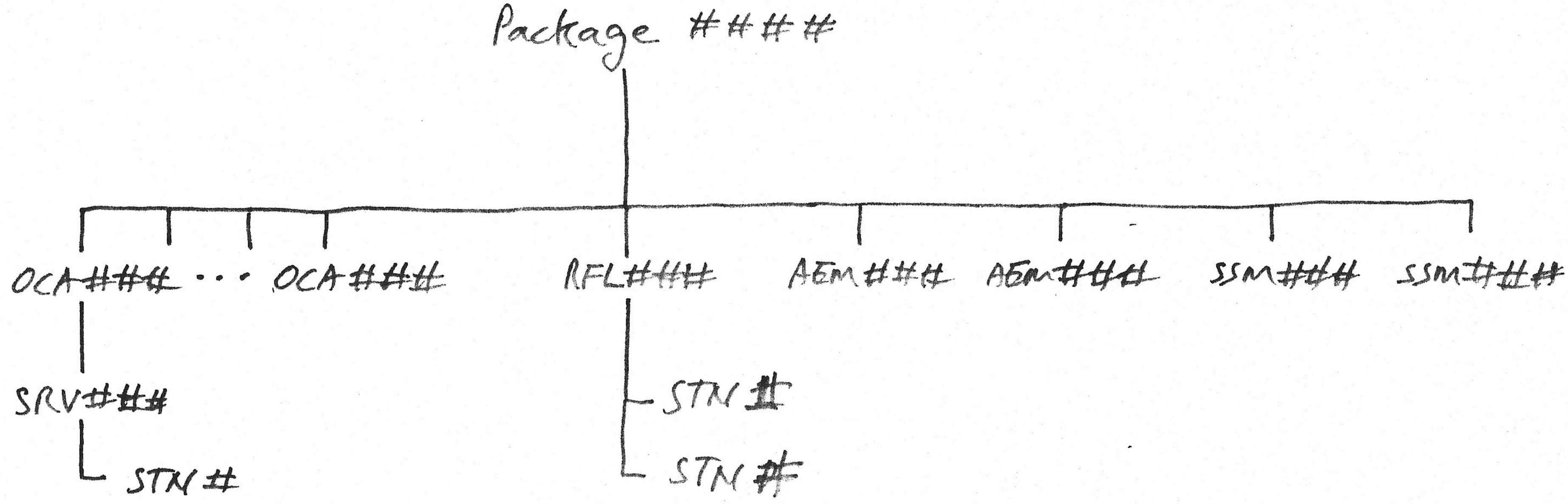
*Evaluator Script (script used for guiding the evaluation which exercised all important CUBRICON features).

Voice Training List

Vocabulary Word for Training	Phonetic Definition	Comment
a	"a"	
ac-pool	"ay-cee-pool"	Aircraft pool or unit at an airbase.
ac-pools	"ay-cee-pools"	Aircraft pools or units at an airbase.
all	"all"	
am	"am"	
an	"an"	
and	"and"	
allstedt	"Allstedt"	Name of city and airbase in E. Germany.
aimpoints	"aimpoints"	
airbase	"airbase"	
any	"any"	
are	"are"	
around	"around"	
assign	"assign"	
at	"at"	
battalion	"battalion"	
battalions	"battalions"	
blink	"blink"	
by	"by"	
cieling	"cieling"	
color-graphics-display	"color graphics display"	
current	"current"	
display	"display"	
dresden	"Dresden"	Name of city and airbase in E. Germany.
duration	"duration"	
enter	"enter"	Enter input for evaluation (at the end of a sentence).
enterq	"enter"	Enter input for evaluation (at the end of a question).
enterverb	"enter"	
erfurt	"erfurt"	
ew-germany	"east-west germany"	Name of region that can be displayed by CUBRICON.
expose	"expose"	
factory	"factory"	
fg	"ef-gee"	Short for Fulda Gap, the name of a region that can be displayed by CUBRICON.
fighter	"fighter"	
flight	"flight"	
fourty-fifth-tfw-ef-111e	"forty fifth tee-ef-dub'l-u ee-ef one-eleven-ee"	45th Tactical Fighter Wing of EF-111Es.
fourty-ninth-tfw	"forty ninth tee-ef-dub'l-u"	49th Tactical Fighter Wing.
fourty-ninth-tfw-f-111f	"forty ninth tee-ef-dub'l-u ef one eleven-ef"	49th Tactical Fighter Wing of F-111Fs.
for	"for"	
form	"form"	
forms	"forms"	
Franz	"Franz"	
from	"from"	
fuel	"fuel"	

fulda-gap	"fulda gap"	Fulda Gap, name of region that can be displayed by CUBRICO
here	"here"	
highlight	"highlight"	
in	"in"	
is	"is"	
it	"it"	
heliport	"heliport"	
heliports	"heliports"	
kc-135	"kay-cee one-thirty-five"	KC-135 aircraft.
lbs	"lbs"	Pounds.
lindsey	"lindsey"	
list	"list"	
location	"location"	
make	"make"	
map	"map"	
merseberg	"Merseberg"	Name of a city and airbase in E. Germany.
minutes	"minutes"	
mission	"mission"	
mobility	"mobility"	
munition	"munition"	
munitions	"munitions"	
name	"name"	
nationality	"nationality"	
near	"near"	
nurenberg	"Nuernberg"	Name of a city and airbase in W. Germany.
oca	"oh-cee-ah"	
oca345	"o-cee-ah three forty five"	Offensive Counter Air Mission Number 345.
oca445	"o-cee-ah four forty five "	Offensive Counter Air Mission Number 445.
oca555	"o-cee-ah five fifty five"	Offensive Counter Air Mission Number 555.
of	"of"	
on	"on"	
package	"package"	
packages	"packages"	
path	"path"	
pause	"pause"	Command that deactivates speech recognition system (see also resume command).
pkg0023	"pee-kay-gee oh-oh-two-three"	Package Number 0023.
pkg0066	"pee-kay-gee oh-oh-two-three"	Package Number 0066.
plan	"plan"	
plans	"plans"	
point	"point"	
present	"present"	
refueling	"refueling"	
region	"region"	
remove	"remove"	
reset	"reset"	Command that starts new word sequence in speech recognition system, deleting previously started word sequence.
resume	"resume"	Command that reactivates speech recognition system (after a pause command)

rfl345	"ar-ef-el three forty five"	Refueling Mission Number 345.
rhein-main	"Rhein Main"	Name of a city and airbase in W. Germany.
sam	"sam"	Surface-to-air-missile.
seven	"seven"	7:00
seven-fifty-five	"seven fifty five"	The time 7:55.
seven-fourty-five	"seven forty five"	The time 7:45.
seven-o-two	"seven oh two"	The time 7:02.
seven-twenty	"seven twenty"	The time 7:20.
six	"six"	The time 6:00.
six-fifty	"six fifty"	The time 6:50.
stargard	"Stargard"	Name of a city and airbase in Poland.
start	"start"	
starting	"starting"	
stn	"es-tee-en"	Short for station.
stn002	"es-tee-en oh-oh-two"	Station Number 002.
stn244	"es-tee-en two-two-" four"	Station Number 224.
storage	"storage"	
strike	"strike"	
svc	"SVC"	
svc001	"SVC001"	
svc345	"SVC345"	
svc445	"SVC445"	
table	"table"	
tank	"tank"	
tanks	"tanks"	
target	"target"	
targets	"targets"	
this	"this"	
the	"the"	
these	"these"	
threat	"threat"	
threats	"threats"	
time	"time"	
to	"to"	
twenty-one-thousand-nine sixty	"twenty one thousand nine sixty"	The number 21960.
twenty	"twenty"	The number 20.
twenty-thousand-nine-fourty-two	twenty thousand nine fourty two	The number 20942.
type	"type"	
types	"types"	
units	"units"	
nit	"uni"	
value	"value"	
what	"what"	
where	"where"	
will	"will"	
window	"window"	
within	"within"	
zoom-in	"zoom in"	Command that causes the system to display more detailed view of a map area.



Package – Group of related missions
OCA – Offensive Counter Air Mission
SRV – Service mission (i.e., refueling)
STN – Station (for refueling mission)
RFL – Refueling mission
AEM – Air Escort Mission
SSM – SAM Suppression mission

Miscellaneous Commands

"reset" <clear input> -- To start speech input over

<control>b -- Move cursor back 1 space
<control>f -- " " forward 1 space
<control>e -- " " to end of line
<rubout> -- Delete previous character
<meta><rubout> -- Delete previous word
"pause/resume -- Ignore voice input/resume

Training Outline

This Appendix contains an outline of training to be provided in support of the CUBRICON evaluations. The training is virtually identical for both stages of the evaluation program. Items that pertain to one or the other stage exclusively are noted parenthetically.

Lesson Number: 1

Title: Introduction

Contents:

- Purpose and design philosophy
  - Application independant.
  - Multi-media input and output.
  - AI-based.
  - Research tool.
  - Integrated system (use desired modality)
- Purpose of evaluation
  - Assess whether CUBRICON meets current requirements.
  - Assess viability of design approach.
  - Make recommendations for improvment (includes suggestions for fine tuning to recommendations for altered design approach).
- Test approach and schedule
  - Two evaluation stages (engineering evaluation and naive user test).
  - Test schedule.
  - Guidelines for applying the evaluation checklist (Stage 1 only).
  - Guidelines for applying evaluation script (Stage 1 only).
  - Guidelines for free-form evaluation (i.e., fully explore evaluation items and stress system to identify weaknesses) (Stage 1 only).
  - Description of the sample problem and guidelines for working its solution (Stage 2 only).
  - System is slow, expect delays.

Lesson Number: 2

Title: The Baseline Application

Handout: Overview description of Application Data/Knowledge Base

Contents:

- Overview of application area
  - Types of tactical planning supported
  - Specific tasks supported (e.g., querying and assessing enemy assets, designating targets, locating and assigning friendly assets, coordinating interdependent missions, etc.)
- The Application Data/Knowledge Base
  - Purpose.
  - Structure and general contents.
  - Specific information available.
  - Review form.
  - Breakdown of Packages.

Lesson Number: 3

Title: Interactive Features

Contents:

- Describe and demonstrate input techniques (with student hands-on).
  - Mouse.
  - Voice.
  - Text.
  - Forms.
  - Combinations of above.
- Describe and demonstrate output features (with student hands-on).

- Automatic windowing approach (hybrid tiled/overlapping layout, used window bin, user override).
- Map presentation system.
  * Symbols and icons.
  * The map area (limits).
  * Color code.
  * Map features (e.g., roads, rivers, cities, etc.).
  * Labeling (e.g., of icons, pointing boxes, text boxes, etc.).
- Text windows.
- Tables and forms.
- Voice outputs.

• Describe and demonstrate interactive control (with student hands-on).
  - Sample commands available (e.g., zoom in, request information about icons, query data base).
  - Range of command techniques (e.g., options for combining media, examples of different approaches to obtain the same information).
  - Flight path generation and subsequent presentation.

Lesson Number: 4

Title: Natural Language (NL) Input with Coordinated Pointing Gestures

Handouts: Grammar specification, catagorized vocabulary list

Contents:

• Overview of NL input via voice recognition system and/or keyboard with mouse point gestures

• Grammar available.

• Vocabulary available.

Lesson Number: 5

Title: Interactive Practice

Contents:

- Supervised practice using vocabulary and grammar together with mouse, keyboard, and various displays.

Lesson Number: 6

Title: Summary/Question and Answer

Contents:

- Present course outline as review.
- Entertain questions.

CUBRICON GRAMMAR

Key:

Notation	Meaning	Examples
[abcdefg]	Type of grammar or vocabulary element	[question], [noun]
(abcdef)	Example of grammar or vocabulary element	(what is this), (airbase)
abcdef	Emglish word that is component of element	what, this
/	Or	

Grammar Definition:

Grammar Structure	Definition	Examples
[sentence]	[question]	(where is the Dresdin Airbase)?
	/ [imperitive]	(display the FG region).
	/ [declarative]	(its name is OCA123).
[question]	where [be-verb] [noun-phrase]	(where) (is) (the Dresden Airbase)?
	/ what [be-verb] [noun-phrase]	(what) (is) (this)?
	/ what [be-verb] [noun-phrase] [locative]	(what) (are) (the aimpoints) (within the Dresden Airbase)?
	/ what [be-verb] [noun-phrase] [prepositional- / complement]	(what) (is) (the mobility) (of this [mouse-click])?
	/ what [noun] [be-verb] [locative]	(what) (ac pools) (are) (at the Nuernberg Airbase)?
	/ [be-verb] [demonstrative] [noun-phrase]	(is) (this [mouse-click]) (a sam)?
[declarative]	[noun phrase] [be-verb] [noun phrase]	(its name) (is) (OCA123).
	/ [noun phrase] [verb-group] [noun-phrase]	(I) (am starting) (a new mission plan).
	/ [prepositional-complement] [noun-phrase] [be-verb] [noun-phrase]	(for OCA123) (the origin) (is) (the Nuernberg Airbase).
[imperitive]	[command-verb] [demonstrative]	(remove) (this window [mouse-click]).
	/ [command-verb] [demonstrative] [locative]	(enter) (this [mouse-click]) (here [mouse-click]).
	/ [command-verb] [demonstrative] [noun-phrase]	(make) (this [mouse-click]) (the current package).
	/ [command-verb] [noun-phrase]	(display) (the FG region).
	/ [command-verb] [noun-phrase] [locative]	(enter) (20 minutes) (here [mouse-click]).
	/ [command-verb] [noun-phrase] [noun-phrase]	(make) (PKG0026) (the current package).
	/ [command-verb] [noun-phrase] [prepositional-complement]	(plan) (a flight path) (for OCA345).
	/ [command-verb] [prepositional-complement]	(zoom-in) (on this point [mouse-click]).

Grammar Element Definition:

Grammar Element	Definition	Examples
[noun-phrase]	/ [det] [noun]	(the) (threats)
	/ [det] [proper-name-group] [noun]	(the) (Fulda Gap) (region)
	/ [det] [noun] [proper-name]	(the) (ac-pool) (45TFW-F-111F)
	/ the value [proper-name]	(the) (value) (KC135)
	/ the value [number] [unit]	(the) (value) (21960) (lbs)
	/ [mouse-click-list]	("sequential mouse-clicks on visible objects")
	/ [proper-name]	(STN244)
	/ [quantifier] [noun]	(any) (sams)
	/ [time]	(7:45)
	/ [duration]	(20 minutes)
[proper-name-group]	[proper-name]	(OCA123)
	/ [proper-name] [conjunction] [proper-name-group]	(OCA123) (and) (OCA345)
[demonstrative]	this [mouse-click]	(this) ("mouse-click on visible object")
	/ this [noun] [mouse-click]	(this) (sam) ("mouse-click on visible object")
	/ these [mouse-click-list]	(these) ("sequential mouse-clicks on visible objects")
[locative]	here [mouse-click]	(here) ("mouse-click on location")
	/ [locative preposition] [noun-phrase]	(around) (the Dresdin airbase)
[prepositional-complement]	[preposition] [noun-phrase]	(for) (OCA123 and OCA345)
	[preposition] [demonstrative]	(on) (this point "mouse-click on location")
[verb-group]	[auxiliary] [main-verb]	(will) (depart)
[be-verb]	is / are	
[command-verb]	assign / blink / call / display / enter / expose / highlight / list / make / plan / present / remove / zoom-in	
[auxiliary]	am / will	
[main-verb]	starting / depart / strike	
[det]	a / an / the	
[quantifier]	some / any	
[conjunction]	and	
[locative-preposition]	around / at / in / near / on / within	

[preposition]	as / by / for / from / of / to	
[mouse-click]	[click any mouse button on any visible object or location on either the monochrome or color display]	("mouse-click on table entry")
[time]	[hours]:[minutes]	(6):(45)
[duration]	[number] minutes	(20) (minutes)
[noun]	- see attached list of nouns and proper names - / [pronoun]	radar
[pronoun]	it	
[proper-name]	- see attached list of nouns and proper names -	Nuernberg

Vocabulary Quick Reference

Threats

airbase
(airbases)
(artillery)
Dresden airbase
Erfurt airbase
sam
sams
(tank battalion(s))
(threat)
threats

Targets (other than threats)

(aimpoint)
aimpoints
factory
(factories)
Franz Munitions
(fuel tank(s))
(fuel storage tank(s))
(Hans Steel)
(heliport)
heliports
(munition factory)
(munition factories)
(population center(s))
(radar(s))
(Schultz Steel)
(Schwartz Munitions)
(steel plant(s))
target
(targets)

Cities

Allstedt
(Altenberg)
(Brandis)
(Cochsted)
(Dessau)
Dresden
Erfurt
(Findsterwalde)
(Leipzig)
Lindsey
Merseberg
Stargard

Geographic Descriptors

East-West Germany
EW Germany
FG
Fulda Gap
(locations)
map
(orbit location(s))
point
(population center(s))
region

Attributes of Geographic Objects

ceiling
(ceilings)
(disposition)
location
(locations)
(mobilities)
mobility
name
(names)
nationaly
(nationalities)
type

Mission Descriptors

(aimpoint)
(air escort mission)
aimpoints
(aircraft unit(s))
(current mission plan)
(current package)
(disbursement)
duration
(flight path(s))
mission
(missions)
(mission-plan)
mission plans
OCA
(OCA mission plan(s))
(OCA plan(s))
(orbit location(s))
package
(packages)
(package number)
(PKG@@@)
(PKG@@@@)
(post-target refueling mission(s))
(pre-target refueling mission(s))
(refueling mission(s))
(RFL@@@)
RFL345
(sam suppression mission(s))
(SSM@@@)
STN
(STN number)
(STN mission plans)
(STN plan(s))
(STN@@@)
STN002
STN244
(strike date)
SVC
(SVC mission plan(s))
(SVC@@@)
SVC001
SVC345
SVC445
(SVC@@@@)
(TOD)
(TOT)
(TSM@@@)

Friendly Assets

(34TFS)
(34TFS-F-15C)
45TFW-EF-111E
(45TFW-F-4G)
(45TFW-F-16C)
(45TFW-F-16D)

49TFW-F-111F
(49TFW-F-16C)
(435TAW)
(45TFW)
49TFW
a
ac-pool
(ac-pools)
airbase
(airbases)
(aircraft)
(aircraft unit(s))
(EF-111E)
(F-4G(s))
(F-16D(s))
(F-111F(s))
(F-16C(s))
(F-15C(s))
fighter
(fighters)
(fighter base(s))
heliport
KC-135
(Leipsig) airbase
(missile(s))
Nuernberg airbase
(orbit location(s))

Preposition

as
by
for
from
of
to

Display Descriptors

color graphics display
form
forms
forms window
map
(mission plan form)

Command

(OCA form(s))
(slot(s))
table
window
(windows)

Auxiliary

am
will

Determiner

a
an
the

Quantifier

some
any Rhein Main

Conjunction

and

Pronoun

it

Be Verb

are
is

Verb

assign
blink
call
display
enter
expose
highlight
list
make
plan
present
remove
zoom-in

Main Verb

starting
depart
strike

Locative

around
at
in
near
on
within

Nouns

ac-pool
(ac-pools)
(ac-type)
(aimpoint)
aimpoints
airbase
(airbases)
(aircraft)
(aircraft unit(s))
(artillery)
battalion
battalions
(color)
color graphics display
(current mission plan)
(current package)
(disbursement)
duration
factory
(factories)
fighter
(fighters)
(fighter base(s))
flight
(flight path(s)) form
forms
forms window
fuel
(fuel tank(s))
(fuel storage tank(s)) (heliport)
heliports
(hour)
location
(locations)
map
(minute)
minutes
(missile(s))
mission
(missions)
(mission-plan)
(mission plan form)
mission plans
(mobilities)
mobility
(munition)
munitions
(munition factory)
(munition factories)
name
(names)
nationality
(nationalities)
(number)
OCA
(OCA form(s))
(OCA mission plan(s))
(OCA plan(s))
(orbit location(s))
(origin)
package
(packages)
(package number)
plan
plans
(plant(s))
point
(population center(s))
(post-target refueling mission(s))
(pre-target refueling mission(s))
(preparer)
(priority)
(radar(s))
(refueling mission(s))
region
regions
sam
(sam suppression mission(s))
sams
(slot(s))
start
(STN number)
(STN mission plans)
(STN plan(s))
(strike date)
(SVC mission plan(s))
table
(tables)
tank
tanks
(tank battalion(s))
target
(targets)
(target strike mission)
(target strike missions)
(task(s))
(threat)
threats
time
(TOD)
(TOT)
type
(types)
unit
units
value
window
(windows)

Proper Names

(34TFS)
(34TFS-F-15C)
45TFW-EF-111E
(45TFW-F-4G)
(45TFW-F-16C)
(45TFW-F-16D)
49TFW-F-111F
(49TFW-F-16C)
(435TAW)
(45TFW)
49TFW
(AEM@@@)
Allstedt
(Altenberg)
(Brandis)
(Cochsted)
Dresden
(EF-111E)
East-West Germany
Erfurt
EW Germany (F-4G(s))
(F-16D(s))
(F-111F(s))
(F-16C(s))
(F-15C(s))
FG (Findsterwalde)
Franz Munitions
Fulda Gap
(Grossenhain)
(Hans Steel)
(Hans Steel)
KC-135
(Leipzig)
Lindsey
Merseberg
Nuernberg
(OCA@@@)
OCA345
OCA445
OCA555
(PKG@@@)
(PKG@@@@)
(RFL@@@)
RFL345
Rhein Main
(Schultz Steel)
(Schwartz Munitions)
(SSM@@@)
Stargard
STN
(STN number)
(STN mission plans)
(STN plan(s))
(STN@@@)
STN002
STN244
SVC
(SVC mission plan(s))
(SVC@@@)
SVC001
SVC345
SVC445
(SVC@@@@)
(TSM@@@)

Vocabulary List

Vobabulary Word	Comment
(34TFS)	
(34TFS-F-15C)	
45TFW-EF-111E	
(45TFW-F-4G)	
(45TFW-F-16C)	
(45TFW-F-16D)	
49TFW-F-111F	
(49TFW-F-16C)	
(435TAW)	
(45TFW)	
49TFW	
a	
ac-pool	
(ac-pools)	
(ac-type)	
(AEM@@@)	
(aimpoint)	
aimpoints	
(air escort mission)	
airbase	
(airbases)	
(aircraft)	
(aircraft unit(s))	
Allstedt	
am	
an	
and	
any	
are	
around	
(artillery)	
(as)	
assign	
at	
(Altenberg)	
battalion	
battalions	
blink	
(Brandis)	
by	
(call)	
ceiling	
(ceilings)	
(Cochsted)	
(color)	
color graphics display	
current	
(current mission plan)	All words available individually in voice system
(current package)	All words available individually in voice system
(depart)	

(Dessau)	
display	
(disposition)	
(disbursement)	
Dresden	
duration	
(EF-111E)	
East-West Germany	Text accompanying this voice input is ew-germany
enter	
expose	
Erfurt	
EW Germany	Short for East-West Germany.
(F-4G(s))	
(F-16D(s))	
(F-111F(s))	
(F-16C(s))	
(F-15C(s))	
factory	
(factories)	
FG	Short for Fulda Gap, a region that can be displayed by CUBRICON.
fighter	
(fighters)	
(fighter base(s))	
(Findsterwalde)	
flight	
(flight path(s))	All words available individually in voice system.
for	
form	
forms	
forms window	All words available individually in voice system.
Franz Munitions	
from	
fuel	
(fuel tank(s))	All words available individually in voice system.
(fuel storage tank(s))	All words available individually in voice system.
Fulda Gap	
(Grossenhain)	
(Hans Steel)	
(heliport)	
heliports	
here	
highlight	
(hour)	
in	
is	
it	
(its)	
KC-135	
(Leipzig)	
Lindsey	
list	
location	
(locations)	
make	

map	
Merseberg	
(minute)	
minutes	
(missile(s))	
mission	
(missions)	
(mission-plan)	All words available individually in vioce system.
(mission plan form)	All words available individually in vioce system.
mission plans	
(mobilities)	
mobility	
(munition)	
munitions	
(munition factory)	All words available individually in voice system.
(munition factories)	
name	
(names)	
nationality	
(nationalities)	
near	
(number)	
Nuernberg	
OCA	
(OCA form(s))	All words available individually in voice system.
(OCA mission plan(s))	
(OCA plan(s))	
(OCA@@@)	Any three digit number is acceptable.
OCA345	
OCA445	
OCA555	
of	
on	
open	
(orbit location(s))	
(origin)	
package	
(packages)	
(package number)	
(PKG@@@)	Any three digit number is acceptable.
(PKG@@@@)	Any three digit number is acceptable.
plan	Must be used as part of phrase (e.g., OCA plan).
plans	Must be used as part of phrase (e.g., OCA plans).
(plant(s))	
point	
(population center(s))	
(post-target refueling mission(s))	
(pre-target refueling mission(s))	
(preparer)	
present	Understood as a verb.
(priority)	
(radar(s))	
refueling	Must be used as part of phrase (e.g., refueling mission).
(refueling mission(s))	All words available individually in voice system.

(RFL@@@)	Any three digit number is acceptable.
RFL345	
region	
regions	
remove	
Rhein Main	
sam	
(sam suppression mission(s))	
sams	
(Schultz Steel)	
(Schwartz Munitions)	
(slot(s))	
(some)	
(SSM@@@)	Any three digit number is acceptable.
Stargard	
start	
starting	
(steel plant(s))	
STN	Must be used as part of a phrase (e.g., STN plan).
(STN number)	
(STN mission plans)	
(STN plan(s))	
(STN@@@)	Any three digit number is acceptable.
STN002	
STN244	
strike	
(strike date)	
SVC	
(SVC mission plan(s))	All words available individually in voice system.
(SVC@@@)	Any three digit number is acceptable.
SVC001	
SVC345	
SVC445	
(SVC@@@@)	Any four digit number is acceptable.
table	
(tables)	
tank	
tanks	
(tank battalion(s))	All words available individually in voice system.
target	
(targets)	
(target strike mission)	All words available individually in voice system.
(target strike missions)	
(task(s))	
the	
these	
this	
(threat)	
threats	
time	
to	
(TOD)	
(TOT)	
(TSM@@@)	Any three digit number is acceptable.

type
(types)
unit
units
value
(was)
what
where
window
(windows)
will
within
zoomin

Associated Service Mission (Pre-Target)

Mission Start Time

Distrobution (lbs fuel)

OCA Mission Number

Origen Airbase

Time of Departure

Aircraft Unit (e.g. Fighter Wing)

Station Number

Duration (minutes)

PACKAGE WORKSHEET

PKG#	0025	Preparer's Name		Date Prepared		Priority	

OFFENSIVE COUNTER AIR MISSIONS

Mission	OCA#	Origin	TOD	#AC	AC Type	SCL	AC Pool
1	345	Nurnberg City	06:00				49tfw-F-1
2	445	Rhein Main Fighter Bas	05:45				45tfw-Ef-
3							
4							

PRE-TARGET REFUELING

SVC#	STN#	Start	Dur.	Disbur.

Time on Target

Target (aimpoint)

TARGET STRIKE MISSION

Mission	Aim Point	TOT
1	6-24-Dresden Runway	07:02
2	6-24-Merseberg Runway	06:50
3		
4		

POST-TARGET REFUELING

SVC#	STN#	Start	Dur.	Disbur.
345	244	07:45	00:10	20942 lbs
445	244	07:25	00:10	21960 lbs

Associated Service Mission (Post-Target) (same breakdown as for Pre-mission)

Refueling Mission

REFUELING MISSION

RFL#	345	TOD	07:00	AC Type	Kc-135 Air	Load		Origin	Rhein Main Fighter Base

Station	STN#	Start Time	Stop Time	Orbit Location
1	244	07:20	07:55	50.75 N Latitude, 13.55 E Longitude
2				

AIR ESCORT MISSIONS

Mission	AEM#	Origin	TOD	#AC	ACT	SCL	Remarks
1							
2							

Mission	SSM#	Origin	TOD	#AC	ACT	SCL	Target	TOT
1								
2								

Look at the color graphics screen. The aimpoints within dresden are being presented.

The system is deciding whether to generate a monochrome table.

The corresponding table is being presented on the monochrome screen.
=>> Make PKG0025 the current package.
I understand, assigning PKG0025 as the current package.
=>> Display the form.
=>>

Mouse-R: Menu.
To see other commands, press Shift, Control, Meta-Shift, or Super.

[Fri 11 Aug 5:33:49] steveg CL SNEPS: User Input GC: Scanning resident pages

Script

(1) Display the Fulda Gap region.

(2) Where is the Erfurt airbase?

(3) Where is the Nuernberg airbase?

(4) Where is the Stargard airbase?

(5) Blink the heliports.

(6) What are the threats around the Dresden airbase?

(7) Zoom in on this point <point>.

(8) List the packages.

(9) Make PKG0023 the current package.
(NOTE: in our demo we will want to start with a half blank form and fill in enough info about a mission plan in order to be able to present it via the multi-modal presentation capability)
or
Make this <point-on-package-table> the current package.

(10) Display the forms window.

(11) Enter OCA445 here <oca-#2>.

(12) Expose this window <pt-main-map>.

(13) Enter <pt-nurnberg> here <oca-origin2>.

(14) Enter 6:00 here <oca-tod2>.

(15) What are the aimpoints within the Dresden airbase?

(16) What are the aimpoints within the Dresden airbase?

(17) What is the mobility of these <point-more-4-sams>?

(18) Enter this <pnt-to-sam> here <pnt-form-aimpt-slot>.

(19) What is the mobility of these <point-less-5-sams>?

(20) Is this <point-on-Dresden-aimpoint-window> a sam?

(21) What are the aimpoints within the Erfurt airbase?

(22) Highlight this <point> on the table.

(23) Highlight this <point> on the {map/color-graphics display}.

(24) Enter this <pnt-to-Dresden-runway> here <pnt-form-aimpt-slot>.

(25) Enter 7:02 here <strike-TOT>.

(26) What ac-pools are at the Nuernberg airbase?

(27) Enter this <point-ac-pool> here <oca-ac-pool2>.
or

Enter the ac-pool 45TFW-EF-111E here <oca-ac-pool2>.

(28) Enter SVC445 here <svc-#2>.

(29) Enter 7:45 here <svc-start>.

(30) Enter 20 minutes here <svc-durat2>.

(31) Enter the value 20942 lbs here <svc-disbur2>.

(32) Enter STN244 here <svc-stn-#1>.

(33) Enter STN244 here <svc-snt-#2>.

(34) Enter this location <click on map> here <stn-orbit>.

(35) Plan a flight path for OCA345.

(36) Plan a flight path for OCA445.
(In the near future, the user should be able to enter the way points of the flight path via mouse-points on the map.)

(37) Present the OCA345 and OCA445 mission plans.

(38) Remove the flight paths.

(39) Make PKG0066 the current package.

(40) Zoom in on this point <point-on-map>.

E.2 COMPLETED HUMAN ENGINEERING EVALUATION CHECKLIST

This sections contains the completed Human Engineering Evaluation Checklist. It includes both the top-level summary portions as well as the more detailed and supporting checklist item responses.

Interface Engineering Evaluation Checklist

Rate CUBRICON's performance with respect to the evaluation categories. The numbered items (-1, -2, etc.) within each category will help in making your assessments. These numbered items are not intended to serve as the sole basis upon which to make your assessments. All observations you believe are relevant should be considered. State the rationale on which you base yoúr ratings.

Refer to Smith and Mosier (1986) to guide your evaluation. Many of the numbered items include references to Smith and Mosier. These references are listed within parentheses at the end of the items. Bear in mind that CUBRICON is built using new technology. Its approach to user-interface design is new and inovative. The guidelines in Smith and Mosier were developed for conventional interfaces. If CUBRICON violates any of the Smith and Mosier guidelines, ask yourself whether the violations could represent an improvement over conventional user-interface approaches, or whether they are the result of poor design.

Finally, be critical! Don't be afraid to tell us what you think (good and bad). Stress the system. Find out where its weak points are and tell us how we can make it better. If you need more time, take it. The results of this evaluation will guide future design efforts.

Note: The numbered items within the evaluation categories are also cross-referenced to the top-level CUBRICON goals that were stated in the SOW. These are noted using the *number* format. These references are not meant to be used during the hands-on portion of the CUBRICON evaluation but will be used during subsequent analysis and reporting.

1. The Efficiency and Effectiveness of Making INPUTS to CUBRICON.

1.1 Rate the general ease, naturalness, and effectiveness of making inputs to CUBICON:

Rating		Comments
Excellent	.....	*Frequent misinterpretations of speech input was inefficient. However, a better speech recognition system and increased training and practice could alleviate this problem. Allocation of mouse to screen (color graphics or monochrome) via the keyboard was cumbersome. This could be improved by using the right and left buttons on the mouse to select the desired screen. The use of specific command verbs to initiate specific actions was difficult to remember, particularly when the verbs have similar*
Very Good	.....	
Adequate	..X..	
Poor	.....	
Extremely Poor	.....	

meanings (e.g., display and present). A more generalized use of verb commands would lessen memory load.
The option of input media (speech, text, pointing) or combinations of media made the system enjoyable to use. It accomodates differences in task demands and user preferences.

-1 Inputs to CUBRICON could be made using the most convenient and desired media/modalities and in a manner that seemed natural and efficient. *1*. Circle one: Always, *Usually*, Sometimes, Rarely, Never. Comments:

Some limitations include: no pointing at form except for input; allocation of mouse to screen was cumbersome (Assignment of mouse, i.e., cursor, to screens with mouse button rather than keyboard would be more efficient); only mouse click could be used to specify location (not speech) with zoom-in command (speech or text didn't work); it is inconvienient when speech is misinterpreted requiring reset (text has to be cleared and the statement must be repeated from the beginning).

-2 Inputs to CUBRICON were correctly understood the first time without clarification or reformating. *1*. Circle one: Always, Usually, Sometimes, *Rarely*, Never. Comments:

Frequent misinterpretation of speech required clarification.

-3 Verbal reference to objects within the CUBRICON data base could be made using desired and natural terminology (3.1.6.5, 3.1.7.1). *1*. Circle one: Always, Usually, *Sometimes*, Rarely, Never. Comments:

Structure of command sentences was somewhat rigid (e.g., "zoom-in on this -mouseclick- point" was acceptable, "zoom-in on this -mouseclick- location" and "zoom-in on this -mouseclick- " were not). The system would tolerate ommissions of "the", however. Also, command verbs tied to specific actions were difficult to remember.

-4 CUBRICON provided for efficient specification and input of spatial/geographic information (e.g., flightpaths, putting objects at desired locations) (1.6.2 - 1.6.9). *1*. Circle one: Always, *Usually*, Sometimes, Rarely, Never. Comments:

Did not put objects at locations. No feedback for first location specified for flightpath was provided until second location was specified.

-5 The use of data entry forms was straightforward and not prone to errors (e.g., areas for data entry were clearly delineated and movement between them was natural and efficient) (1.0.6, 1.4.all, 2.2.all, 3.1.2.1 - 3.1.2.4). *1*. Circle one: Always, *Usually*, Sometimes, Rarely, Never. Comments:

It was not clear which areas were required to be completed or whether there was a hierarchical order to fill them. Format, and movement between areas was straightforward.

-6 Pointing at desired objects could be accomplished equally well on the various types of windows displayed (e.g., tables, maps) and on the monochrome display as well as the color display. *2*. Circle one: Always, *Usually*, Sometimes, Rarely, Never. Comments:

Pointing worked well for maps and for text. Pointing on the form was only enabled for input. When numerous icons were displayed in close proximity, more than one icon was picked-up by point and the system crashed.

Reference: Also consider items: 1.3-4.

1.2 Rate the ability of CUBRICON to accept, integrate, and understand inputs that were made using multiple media/modalities:

Rating		Comments
Excellent	.....	*Being able to point at several objects as part of an input, the combined use of speech and pointing for an input, and the use of multiple windows for an input are all excellent features that made the system easy to use.*
Very Good	..X..	
Adequate	.....	
Poor	.....	
Extremely Poor	.....	

-1 The ability to point and speak at the same time was helpful in making inputs. *1*. Circle one: *Always*, Usually, Sometimes, Rarely, Never. Comments:

Easy to use.

-2 Mouse points were correctly related to the intended objects described via natural language. *6*. Circle one: Always, *Usually*, Sometimes, Rarely, Never. Comments:

-3 It was possible to point at multiple objects as part of an input, and these were correctly integrated and understood within the dialogue by CUBRICON. *6*. Circle one: Always, *Usually*, Sometimes, Rarely, Never. Comments:

When the system was querried on the mobility of several icons, one of which had no mobility, the voice response called it a miss while the text said it had no mobility.

-4 It was possible to make inputs efficiently using multiple windows (e.g., pointing at objects in different windows when defining a target list). *1*. Circle one: Always, *Usually*, Sometimes, Rarely, Never. Comments:

Unable to point at the form except for input.

-5 The ability to define inputs on one window by pointing at objects on another window (e.g., in completing forms) was efficient and easily accomplished. *1*. Circle one: Always, *Usually*, Sometimes, Rarely, Never. Comments:

Process was easy, except for allocating mouse to screen.

Reference: Also consider items: 1.1-1.

1.3 Rate the ability of CUBRICON to understand inputs based on the dialogue context:

Rating		Comments
Excellent	.....	*Formatting of speech input was somewhat*
Very Good	.....	*rigid. The system was not very tolerant*
Adequate	..X..	*of deviation from this structure. Allowing*
Poor	.....	*verb commands with similar meanings (e.g.,*
Extremely Poor	.....	*display and present) to be used inter-*

changably would be helpful.

-1 The formulation of inputs to CUBRICON flowed naturally from the context of the displays and dialogue and did not require translation in in order to achieve acceptable structure and formats (e.g., the terminology acceptable for data control and input was consistent with the style, terminology, and format used for output) (2.0.7, 3.1.8.5). *1*. Circle one: Always, Usually, *Sometimes*, Rarely, Never. Comments:

Structure of commands requires memorization. Formulation of data input to the form was automatically structured by the system (e.g., "arrival time is six" formatted as 6:00) making it convienient to use.

-2 Inputs to CUBRICON could be made within the ongoing dialogue without invoking special procedures or calling special displays (1.0.2). *1*. Circle one: Always, *Usually*, Sometimes, Rarely, Never. Comments:

Most input done on one screen containing form, command scroll area, and system.

-3 Inputs that are illogical based on the task and data context, were noted by CUBRICON and communicated (1.6.19, 1.7.1). *5*. Circle one: Always, Usually, Sometimes, *Rarely*, Never. Comments:

Very little error trapping. The system would proceed with next command without recognition that required information was omitted. Error feedback messages were not informative.

-4 CUBRICON provided prompts or reminders based on the task being performed (e.g., guides for accomplishing complicated procedures were available when needed)(3.1.8.6, 3.1.8.7, 3.2.4, 3.2.5). *7*. Circle one: Always, Usually, Sometimes, *Rarely*, Never. Comments:

No prompts were available for filling out form (e.g., what areas were required to plan a flight path, guidance for possible hierarchy of form entries). Also, no prompts or error checking.

-5 CUBRICON was able to correctly relate pronouns and indefinate references to their proper referent (based on the preceding dialogue). *8*. Circle one: Always, *Usually*, Sometimes, Rarely, Never. Comments:

Voice output used pronouns correctly. Pronouns not accepted as input.

-6 CUBRICON was able to correctly interpret inputs based on the context of the dialogue (e.g., requests for information produced outputs relevant to the dialogue; requests that made no sense based on the context were questioned). *8*. Circle one: Always, *Usually*, Sometimes, Rarely, Never. Comments:

-7 Ambiguous mouse points were correctly resolved by CUBRICON based on the context of the dialogue. *5*. Circle one: Always, *Usually*, Sometimes, Rarely, Never.

- Inaccurate points were correctly resolved.
- Incorrect points that made no sense were corrected or questioned.

Comments:

Had problem when selected icon was in close proximity to other icons.

-8 The CUBRICON vocabulary and grammar was sufficient for expressing desired concepts and data. *1* Circle one: Always, *Usually*, Sometimes, Rarely, Never. Comments:

The vocabulary and grammar seemed appropriate the the application.

Reference: Also consider items: 1.2-2, 1.2-3.

2. The Efficiency and Effectiveness of CUBRICON OUTPUTS.

2.1 Rate the general understandability, effectiveness, and smoothness of CUBRICON outputs:

Rating		Comments
Excellent	.....	*Voice output was sometimes difficult to*
Very Good	..X..	*interpret. Additional labelling of maps and*
Adequate	.....	*tables would be helpful. Outputs were*
Poor	.....	*generally clean and easy to understand.*
Extremely Poor	.....	

-1 CUBRICON outputs were clear and understandable without requests for clarification. Information was presented in a form that could be clearly and unambiguously understood and could be related to the task being performed (2.0.3, 2.4.9). *4*. Circle one: Always, Usually, *Sometimes*, Rarely, Never. Comments:

Voice was difficult to interpret on many occasions, however, this problem could disappear with increased training and usage.

-2 Information needed for interpreting displays was readily available (e.g., a key defining the meaning of symbols used on a map, appropriate supplementary information presented via an appropriate media) (2.0.1). *4*. Circle one:

Always, Usually, *Sometimes*, Rarely, Never. Comments:

Maps and tables were rarely uniquely identified. A key wasn't available but I'm not sure it is necessary.

-3 CUBRICON displays employed labels that were clear, consistent, and helpful. This included labels within displays as well as labels identifying the display itself (2.2.3 - 2.2.10, 2.3.7 - 2.3.9, 2.3.11, 2.4.11, 2.4.1.1 - 2.4.1.9, 2.7.1.2 - 2.7.1.4, 2.7.5.6). *4*. Circle one: Always, Usually, *Sometimes*, Rarely, Never. Comments:

Maps and tables are not uniquely identified. Labels within the form would probably be more meaningful to the mission planner.

-4 CUBRICON displays employ coding schemes that were clear, consistent, and adequately captured the important distinctions among display elements (2.6.3 - 2.6.38). *4*. Circle one: Always, Usually, *Sometimes*, Rarely, Never. Comments:

Need to code or label maps and tables. Map icons were easy to distinguish from each other. Boxing of highlighted items on tables were difficult to distinguish. Bold face type would be easier to see. Also, red arrow pointer was difficult to see in enemy territory. Use of a distinct color would make it easier to distinguish.

-5 When items were selected (by the user or the system) this was clearly conveyed to the user (1.6.7, 3.4.6). *5*. Circle one: Always, *Usually*, Sometimes, Rarely, Never. Comments:

The first selection when drawing a flightpath isn't indicated until second paint is selected. Selected items on tables were boxed on color graphics display. This was hard to decifer. Bold face would stand out better.

-6 CUBRICON clearly communicated its activities especially when processes were not immediate (2.7.1.7, 3.0.14, 3.0.15). *7*. Circle one: Always, Usually, *Sometimes*, Rarely, Never. Comments:

System status line would generally say "run" or "user input". Specific activities weren't that were ongoing were not identified.

-7 Spatial relationships among graphic elements (e.g., elements on a map) were clearly presented (e.g., it was possible to accurately judge distances or query for exact distances (1.6.1.5, 1.2.2.4, 2.4.1.11, 2.4.1.12, 2.4.8.3, 2.4.8.3). *5*.

Circle one: Always, Usually, Sometimes, Rarely, Never. Comments:

Unable to querry for exact distances. Option to impose a grid on map would be helpful. Part of scale was obscured at the origin.

-8 The general organization and layout of windows was efficient for the tasks at hand. *5*. Circle one: Always, *Usually*, Sometimes, Rarely, Never.

- The relative location of related windows allowed necessary comparisons and efficient accomplishment of tasks involving multiple windows.
- Window sizes were sufficient for presentation of the information that each needed to present.

Comments:

Window size seemed appropriate. Location of windows was sufficient for task completion except when automatic deletion removed a map still in use (Note: this occurred following a permanent zoom-in when the context map was removed).

2.2 Rate the appropriateness and effectiveness of CUBRICON media/modality selection and integration:

Rating		Comments
Excellent	.....	*I think the integration of media was effective. Its effectiveness would probably be more apparent with a faster system response time and heavier workload conditions. Voice messages about map display changes were convienient since it allows the user to remain fixated on the display while changes are described.*
Very Good	.....	
Adequate	..X..	
Poor	.....	
Extremely Poor	.....	

-1 Speech, graphic, and textual outputs were used appropriately and in the right proportion to clearly, concisely, and efficiently accomplish the necessary communications (4.0.26 - 4.0.29). *6*. Circle one: Always, *Usually*, Sometimes, Rarely, Never. Comments:

On some occasions presentation of both voice and text message seemed overly redundant. Processing of voice and text message seemed to increase information processing load. Graphic output was clear and easy to use. An exception to this was the flashing of data in the text window after a flight path was *presented.*

-2 Outputs were presented using media/modalities that were appropriate for the content and context of the communication (2.4.1 - 2.4.3, 2.4.6.1, 2.4.6.2, 2.4.8.1). *4*. Circle one: Always, *Usually*, Sometimes, Rarely, Never. Comments:

The ability to obtain a hard copy would enhance the system.

-3 Information that was presented for comparative purposes was displayed in a manner suited for such comparisons (e.g., side by side in a table, highlighted on a map using clear distinguishable codes, etc.) (2.3.1, 2.3.5, 2.4.2, 2.5.13). *4*. Circle one: Always, *Usually*, Sometimes, Rarely, Never. Comments:

Comparison of maps could be made easier by allowing the user to pull a map up from storage. Also, two different forms cannot be displayed side-by-side for comparison, but this may be necessary.

-4 Tables presented information in a manner that facilitated efficient use (e.g., tables were organized by the parameters with which the information was to be accessed or it was a simple matter to reorganize the table to meet this requirement) (2.3.3, 2.3.4, 2.3.12 - 2.3.17, and to a lesser degree all of 2.3). *1*. Circle one: Always, *Usually*, Sometimes, Rarely, Never. Comments:

Ordering of column information may need to be rearranged. Feedback from military type would be helpful (e.g., name of item column imbedded between other columns and was the 5th of 6 columns).

-5 Maps were presented in a way that facilitated their effective and understanding (2.4.8.2 - 2.4.8.9, 2.4.8.15, 2.4.8.17, and to a lesser degree all of 2.4.8). *4*. Circle one: Always, Usually, *Sometimes*, Rarely, Never. Comments:

Occasionally areas of the maps were overly cluttered with icons that couldn't be differentiated. Due to lack of labels, there was no way to quickly see which tables were asociated with which maps. Also the map scale was ambiguous.

-6 CUBRICON made unambiguously clear, which graphically displayed objects were referred to via an associated media/modality. (e.g., verbal outputs were related to associated displayed items in a clear and unambiguous fashion). *6*. Circle one: Always, *Usually*, Sometimes, Rarely, Never. Comments:

Identification was promarily made by change in color code and flashing on graphics display.

-7 It was possible to relate items in tables or on forms to their graphic representations (e.g., on a map). *4*. Circle one: Always, Usually, *Sometimes*, Rarely, Never. Comments:

Maps and tables need to be uniquely identified and the association between them demonstrated (e.g., a line connecting them or coded in some way.

-8 CUBRICON speech output was helpful in providing orientation to other system outputs (e.g., created or modified maps, tables, etc.). *6*. Circle one: Always, Usually, *Sometimes*, Rarely, Never. Comments:

Voice output may be more useful when system response time is faster or workload increases.

-9 Auditory and voice coding was employed effectively to communicate important distinctions among auditory displays (2.6.39 - 2.6.42). *6*. Circle one: Always, Usually, Sometimes, *Rarely*, Never. Comments:

Chimes used to indicate declutter of graphic display as opposed to voice. This was the only auditory coding used.

-10 When relations among information components are important, integrated displays (individual or multi-media) that show those relations were provided (2.5.7, 2.7.2.1). *6*. Circle one: Always, Usually, *Sometimes*, Rarely, Never. Comments:

The relationship between map and entity tables wasn't demonstrated.

-11 Information that was needed temporarily was made available on a temporary basis (rather than cluttering displays with such information) (2.7.5.1). *7*. Circle one: Always, *Usually*, Sometimes, Rarely, Never. Comments:

The use of window overlays served this purpose. The ability to display and remove windows as desired, would enhance the system.

Reference: Also consider items: 2.1-1, 2.1-2.

2.3 Rate CUBRICON's effectiveness at selecting and controlling output quantity, level of detail, and resolution:

Rating Comments

Excellent	.....	*Need to indicate scrolling option on tables. On the whole, maps were easy to read and use. Text in tables on graphics display was difficult to read.*
Very Good	.....	
Adequate	..X..	
Poor	.....	
Extremely Poor	.....	

-1 Map displays contained an appropriate amount of area at an appropriate scale (without resizing) for task accomplishment (e.g., zoomed-in or out to correct amount of detail and area coverage) (1.6.5). *3*. Circle one: Always, Usually, *Sometimes*, Rarely, Never. Comments:

User control of zoomed area size would enhance system, along with addition of zoom-out feature.

-2 Map and other graphic displays, and symbols used within them, were large enough to provide the resolution needed to resolve objects and determine necessary relationships among objects. *3*. Circle one: Always, *Usually*, Sometimes, Rarely, Never. Comments:

Cluster of icons couldn't be deciphered, otherwise, maps and symbology were easy to read. Removable grid would be useful for distance relationships. Tables on graphic display were difficult to read due to text size.

-3 CUBRICON responses to requests for information provided the information in a level of detail consistent with the request and the context of the request (e.g., only necessary information was displayed, yet sufficient detail was provided for the task) (2.0.2, 2.4.5). *3*. Circle one: Always, *Usually*, Sometimes, Rarely, Never. Comments:

-4 When a request for information resulted in a large volume of information, CUBRICON provided a means for dealing with the information in an organized and efficient manner, and/or helped the user rescope the request (2.2.14, 2.4.6.3, 2.5.4, 2.7.2.2 - 2.7.2.6, 2.7.2.10). *3*. (see below for examples). Circle one: Always, Usually, *Sometimes*, Rarely, Never.

- Displayed scrolling window with scroll bar (and perhaps a slider). **No, not completed**
- Presented an indication of the percentage of the information included in and above the displayed window. **No, not provided**
- Indicated the number of items that satisfied a query, and perhaps provided an opportunity to focus the request. **No**
- Presented summary or top-level map of the information. **No**

- And so forth . . .

Comments:

It wasn't clear that information on tables could be scrolled or how scrolling would be accomplished.

-5 An appropriate means for highlighting critical information was used (considering the nature of the critical information, the task context, and other coding schemes in use) (2.4.0.6, 2.4.0.8, 2.4.0.19, 2.4.6.4, 2.6.1). *3*. Circle one: Always, Usually, *Sometimes*, Rarely, Never. Comments:

Required information on the form (for planning a flight path) wasn't indicated. Highlighting on the map was effective. Flashing of items in some instances was excessive and inappropriate.

-6 It seemed that the information being displayed was well controlled (e.g., it was never overwhelming). *3*. Circle one: Always, *Usually*, Sometimes, Rarely, Never. Comments:

Automatic declutter prevented this.

Reference: Also consider items: 2.2-10.

2.4 Rate how well CUBRICON maintained context clarity:

Rating		Comments
Excellent	.....	*Marking of original zoomed-in area*
Very Good	.....	*boundary on original map was helpful.*
Adequate	..X..	*The relationship between maps and*
Poor	.....	*tables needs to be made explicit.*
Extremely Poor	.....	*Labels on maps and tables to uniquely*

identify contents are needed.

-1 When displayed information is relevant only in a certain context, this is clearly communicated (e.g., dynamic information includes time stamp, available weapons indicate compatible platforms). *5*. Circle one: Always, *Usually*, Sometimes, Rarely, Never. Comments:

Not all pertinent time stamps were given on the map (e.g., target strike time) (Note: this occurred during flight path presentation).

-2 Adequate contextual information was available for the proper interpretation and use of displayed information (2.0.11, 2.4.18, 3.4.1, 3.4.7). *5*. Circle one: Always, *Usually*, Sometimes, Rarely, Never. Comments:

-3 When information was best interpreted relative to some significant level or critical value, this comparison was clear from the display (2.4.7, 2.4.8.18). *5*. Circle one: Always, *Usually*, Sometimes, Rarely, Never. Comments:

Some critical times (departure time, strike time) missing from flight path map display.

-4 CUBRICON communicated information in a manner in which the structure of and relationships among the data being entered or displayed was clear (e.g., hierarchical relationships) (1.0.31, 1.6.18, 1.8.12, 2.2.1, 2.3.10, 3.1.6.3) *5*. Circle one: Always, Usually, *Sometimes*, Rarely, Never. Comments:

Hierarchical relationships among data on the form wasn't clear.

-5 When displays are changed (e.g., removing windows or information, zoom-in or out, panning, scrolling), adequate cues are provided for maintaining orientation to the larger context (2.0.11, 2.4.16, 2.4.17, 2.4.18, 2.4.8.2, 2.4.8.11, 2.4.8.16, 2.6.2, 2.7.2.14 - 2.7.2.17, 2.7.3.2, 2.7.3.4, 3.3.5). *5*. Circle one: Always, Usually, *Sometimes*, Rarely, Never. Comments:

Zoom-in area noted with box on original map. No cues provided for scrolling context. For example, when tables have been scrolled, table and column labels scroll off the display and context is lost.

-6 Output formats were consistent with expectations based on the preceding dialogue and the context of pre-existing displays (3.0.16, 3.1.1). *2*. Circle one: Always, *Usually*, Sometimes, Rarely, Never. Comments:

-7 Standard displays used standard formats that were readily identifiable and useable (e.g., standard information was contained and consistently organized in display headings) (2.0.6, 2.0.13, 2.0.14, 2.0.15, 2.1.3, 2.2.13, 2.4.4, 2.4.10, 2.4.12, 2.5.1). *2*. Circle one: Always, *Usually*, Sometimes, Rarely, Never. Comments:

(Windows) not uniquely identified with labels.

-8 CUBRICON provided prompts to help in making standard or required inputs or when omissions were inadvertantly made (1.0.24). *2*. Circle one: Always, Usually, Sometimes, Rarely, *Never*. Comments:

Error feedback was not informative/diagnostic.

-9 Windows were managed in a way that minimized and disruption to display context. *5*. Circle one: Always, *Usually*, Sometimes, Rarely, Never.

- The most important windows were kept on the screen. Windows that were removed were less important and not critical to the ongoing task.
- The largest windows were used for the most important information or when large amounts of detail had to be presented.

Comments:

Reference: Also consider items: 2.1-2, 2.1-3, 2.1-4, 2.1-5, 2.4-8, 2.3-5.

2.5 Rate the appropriateness and effectiveness of voice output as used within CUBRICON integrated outputs:

Rating		Comments
Excellent	.....	*Evaluating the use of voice output is*
Very Good	.....	*problematic because system response time*
Adequate	..X..	*was slow and this magnified the feeling*
Poor	.....	*that voice output was extrinsic to the*
Extremely Poor	.....	*task. A frequent user of the system would*
		not require as much voice feedback as is
		currently provided and a means of adjusting
		the level/amount of voice feedback should
		be addressed in future enhancements of the

system.

-1 Rate the appropriateness of CUBRICON's decisions about when to use speech output. Relate your answer within the following categories *7* : **Adequate**

- CUBRICON speech output did not interrupt user inputs (especially speech inputs) and allowed the user to interrupt if desired or necessary.

  User was unable to interrupt the speech generator. Most of the time speech occurred when processing was taking place (and user could not input data) so speech didn't interrupt user activity. Future enhancements could include a method for stopping voice output when user input occurs.

- Speech output was used when there was a requirement for rapid two-way exchanges of information.

  This was hard to evaluate since system response was slow.

- Speech output was used when the information to be presented dealt with a future time requiring some preparation, and especially when it was intended for immediate use.

  Speech was used to inform the user about display events that were about to happen, and to present information about displayed items.
- Speech was used when it was important to elicit attention from other tasks or activities.

  Speech was used to draw attention to the appropriate display.
- Speech was used when information needed to be presented independant of head or eye movement.

  Speech presentation allows user to fixate on map display activity while receiving information about the display via voice output. At other times the voice output seemed extraneous and overly redundant.

-2 Rate how well CUBRICON constructed speech outputs (i.e., were speech outputs constructed in a manner that maximized overall communication efficiency and understandability?) Relate your answer within the following categories *7* : **Very good**

- Standard and consistent terminology was used for expressing common concepts.

  The use of terminology was standardized and consistent.
- Terminology that was meaningful to the user population was used.

  The terminology seemed appropriate to the application.
- Consistent phraseology was employed throughout all parts of the interface.

  Phraseology was consistent.
- Speech output vocabulary was coordinated and consistent with speech recognition capabilities.

  Vocabulary/terminology of speech input and output was similar.

-3 Voice outputs were constructed in a manner that facilitated accurate perception and understanding. Relate your answer within the following categories *4*: **Sometimes**

- Important words were placed near the end of messages so that surrounding sentence structure would provide context and facilitate intelligibility.

This was the case in most instances. However, when voice output was given regarding mission duration the message, "the duration is this", was given, thus failing to provide the usre with critical information.

- Multi-syllable words were selected when possible to provide linguistic redundancy and reduce phonemic uncertainty within any given word.

  Múltiple syllable words were used when appropriate. It may be the case that frequent usres would want to abbreviate multi-syllable words or multi-word inputs.

- Voice outputs were kept as short as possible.

  Voice messages were generally shorter than text messages. It may be possible to shorten them some more.

-4 CUBRICON speech outputs were coordinated with ongoing tasks and related outputs using other modalities *6*. circle one: Always, Usually, *Sometimes*, Rarely, Never. Comments:

This was a problem because of slow system response. See comments in 2.5.

-5 It was possible to have messages repeated when needed *1*.

Users didn't have the option to have speech messages repeated (in speech) but all speech messages were represented in text messages (this provided for this function).

3. Sequence and System Control Issues

3.1 Rate the efficiency and effectiveness with which the CUBRICON user-interface was controlled:

Rating		Comments
Excellent	.....	*User control of windowing was limited,*
Very Good	.....	*although some features that would*
Adequate	..X..	*improve user control (e.g., zoom-out)*
Poor	.....	*were not enabled yet.*
Extremely Poor	.....	

-1 It was possible to clearly and easily specify desires for control and transformation of maps (e.g., specify area for zoom-in and zoom-out, and panning) (1.6.5, 1.6.6, 1.6.8, 2.4.8.10, 3.0.4, 3.2.1) *1*. Circle one: Always, Usually, *Sometimes*, Rarely, Never. Comments:

Zoom-out, recall of stored maps/tables weren't enabled. User option to selectively declutter or redisplay maps would enhance the system.

-2 It was possible to customize displays to meet personal preferences (e.g., reorganize table columns, redefine area, displayed on a map, redefine window layout). *2*. Circle one: Always, Usually, Sometimes, Rarely, *Never.* Comments:

Features not available.

-3 The automatic management of windows (e.g., positioning, sizing, and removal) was accomplished in a way that facilitated their use while allowing user intervention to achieve alternative window organizations when desired (3.0.1, 3.0.2, 3.0.4, 3.0.5, 3.2.1) *2*. Circle one: Always, Usually, *Sometimes*, Rarely, Never. Comments:

User intervention was limited.

-4 Display changes did not disrupt the ongoing dialogue (e.g., did not remove needed windows, display changes were consistent with expectations) (3.0.7) *4*. Circle one: Always, *Usually*, Sometimes, Rarely, Never. Comments:

Unable to input data while display changes in progress. Occasional E-W Germany map would be removed when I would have liked it available for reference.

-5 Feedback about CUBRICON's acceptance and understanding of inputs was sufficiently quick and clear (3.0.9, 3.0.14, 3.0.15). *5*. Circle one: Always, *Usually*, Sometimes, Rarely, Never. Comments:

System response was slow. Acceptance of first point on flight path wasn't clear.

-6 There was a simple means for indicating to CUBRICON when verbal inputs were meant for CUBRICON and when they were not (e.g., ignore and continue) (1.0.37). *5*. Circle one: *Always*, Usually, Sometimes, Rarely, Never. Comments:

-7 When not all windows were active, CUBRICON clearly indicated which were active and provided an efficient means for selecting desired windows

(2.7.5.7, 2.7.5.8). *6*. Circle one: Always, Usually, Sometimes, Rarely, Never. Comments:

Recall of inactive windows not enabled.

-8 It was possible to cancel partially completed inputs (including voice inputs) and ongoing CUBRICON processes by invoking an explicit CANCEL command (1.0.11, 1.0.35, 3.3.1, 3.3.3). *7*. Circle one: *Always*, Usually, Sometimes, Rarely, Never. Comments:

-9 Control of the CUBRICON interface was handled effectively. Updating of displays was efficient and did not require excessive effort, while at the same time the ultimate control of the interface was available to the user (2.0.8, 2.0.9, 2.7.1, 2.7.1.1, 2.7.1.5, 2.7.1.8, 2.7.1.9, 2.7.2.13, 2.7.3.5 - 2.7.3.8, 2.7.5.1, 2.7.5.2, 3.0.4, 3.2.1). *7*. Circle one: Always, Usually, *Sometimes*, Rarely, Never. Comments:

Very little control of windowing operations.

-10 System operations logically reflected user inputs inputs and desires (3.0.16). *2*. Circle one: Always, *Usually*, Sometimes, Rarely, Never. Comments:

-11 Issuance of commands to CUBRICON was efficient and easy (3.1.5.2, 3.1.5.21, 3.1.5.22). *1*. Circle one: Always, Usually, *Sometimes*, Rarely, Never. Comments:

Speech system difficult to use due to frequency of recognition errors.

-12 It was possible to maintain control over dynamic displays (e.g., PAUSE and CONTINUE commands) (3.3.8 - 3.3.11). *7*. Circle one: Always, Usually, Sometimes, *Rarely*, Never. Comments:

No real-time control over flight path presentation (the only truly dynamic display).

-13 The response time for voice, text, and graphics inputs was sufficiently fast to ensure efficient, continuous dialogues 3.0.18, 3.1.2). *4*. Circle one: Always, Usually, Sometimes, *Rarely*, Never. Comments:

System response time was too slow.

-14 Measure and note typical response times for the following inputs:

- Voice input requesting highlighting of currently displayed information. Note the following: Feedback that request was understood – *No feedback*; Response complete – *1-2 seconds.*
- Voice input requesting a new map to be created (e.g., zoom-in). Note the following: Feedback that request was understood – *3 seconds*; Response complete – *45 seconds (map) plus 20 seconds (table).*
- Selection of a menu item (e.g., select a package from list of available packages).Note the following: Feedback that request was understood – *60 seconds*; Response complete – *62 seconds.*
- Issuance of typed command requesting a standard display (e.g., request display of the forms menu). Note the following: Feedback that request was understood – *70 seconds*; Response complete – *75 seconds.*
- Typical graphic interaction (e.g., point at graphically displayed item for selection). Note the following: Feedback that request was understood – *4 seconds*; Response complete – *4 seconds.*

Reference: Also consider items: 1.1-7, 1.3-4, 2.1-6, 2.4-5, 2.4-8.

3.2 Rate the efficiency and effectiveness of error management and control within the CUBRICON user-interface:

Rating		Comments
Excellent	.....	*Making corrections to text input may*
Very Good	.....	*be easier if user has knowledge of*
Adequate	.....	*EMACS. Without this knowledge, retyping*
Poor	..X..	*of whole lines is required as text can't*
Extremely Poor	.....	*be inserted. This may be, in part, a*
		function of current changes being made
		to the system.

-1 The process of making corrections and "on-the-fly" changes during input was straightforward and efficient (1.0.7, 3.1.5.23, 3.5.12). *1*. Circle one: Always, Usually, Sometimes, *Rarely*, Never. Comments:

It was extremely difficult or impossible to change or undo something, such as adjusting points on the flight path. Also, incorrect text due to speech recognition errors was tiresome to correct.

-2 A requirement for an explicit ENTER action prior to CUBRICON processing of user inputs was imposed when necessary to permit user review or reconsideration (1.0.9, 1.4.1, 3.0.5, 3.1.5.25, 3.5.7,). *2*. Circle one: Always, *Usually*, Sometimes, Rarely, Never. Comments:

Text input didn't did not require "enter". Punctuation served as enter. This is not a typical method. Most users will probably be accustomed to the use of the ENTER key.

3.3 Rate how well CUBRICON performs the functions of data protection:

Rating		Comments
Excellent	.....	
Very Good	.....	
Adequate	.....	
Poor	.....	
Extremely Poor	..X..	

-1 There was ample protection against actions that result in the deletion or significant altering of information (e.g., warnings, undo capability, feedback about results of change prior to action, etc.) (1.3.12, 1.3.13, 2.0.10, 3.5.7, 3.5.8, 3.5.10). *7*. Circle one: Always, Usually, Sometimes, Rarely, *Never*. Comments:

APPENDIX F
WORKING PAPER ON HUMAN FACTORS ISSUES RELATED TO THE USE OF COMPUTER SPEECH GENERATION

The following working paper describes literature relating to human factors issues of using computer generated speech. It contains general guidelines on when to use, and how to construct computer generated speech. These guidelines are related to types of messages to be generated by CUBRICON. Specific CUBRICON issues are also discussed.

This paper was delivered to RADC and DARPA earlier in this program. It is included here for completeness.

CUBRICON Working Paper on Computer Speech Generation
19 August 1988

D. Funke

1. Introduction

This working paper summarizes the results of a review of the literature on computer speech generation. This review was conducted specifically to support the CUBRICON development effort, and the results are reported within the context of CUBRICON needs.

2. General Guidelines for Deciding When and How to Use Speech

This section presents general guidelines for using computer speech generation which applies across all categories of speech output (described below). The literature upon which these guidelines are based are cited. A complete reference and a summary of the articles are given in Appendix A.

2.1 When to Use Speech

The following guidelines relate to deciding when to use speech generation.

- Don't interrupt the human user (McCauley, 1984), but allow the user to interrupt the speech generator (Chapanis, 1975).
- Use speech when there is a requirement for rapid two-way exchanges of information (Simpson et al., 1985; Deatherage, 1972).
- Use speech when the information deals with a future time requiring some preparation (Simpson et al., 1985; Deatherage, 1972), and especially when it is intended for immediate use (Simpson et al., 1985).
- Use speech when the message must elicit attention from other tasks or activities (Simpson et al., 1985)*.
- Use speech when eyes and hands are occupied and unavailable for communicating (Simpson et al.,1985; McCauley, 1984), or information must be presented independent of head movement (Simpson et al., 1985)*.

2.2 How to Construct Speech Output

These guidelines relate to the construction of speech output in a way that maximizes efficiency of use and understandability.

- Standardize vocabulary.
  - Use standard and consistent terminology for expressing common concepts (Cooper, 1987; Bucher et al., 1984; Miller et al., 1951).

----

* Also applies to auditory signals in general.

Select terminology that is meaningful to the user population (Smith and Mosier, 1986).

- Use consistent phraseology throughout all parts of the interface (Smith and Mosier, 1986).

- Coordinate speech output vocabulary with voice recognition capabilities. Users will adopt terminology used by the computer (Zoltan-Ford, 1984).

- Provide supporting context.

  - Put important words near the end of messages. This allows the surrounding sentence structure to provide context to improve intelligibility (Merva and Williges, 1987; McCauley, 1984; Miller et al. 1951).

  - Use multiple syllable words (they have more linguistic redundancy and reduce sequential phonemic uncertainty present in a word) (Bucher et al., 1984; Simpson, 1976).

- Make messages as short as possible (Simpson et al., 1985; Zoltan-Ford, 1984).

- Messages of highest priority (i.e., warnings) should be preceded with a beep or other alerting sound (unless computer generated speech is used exclusively for warnings) (Simpson et al., 1985; Simpson and Williams, 1980; Hakkinen and Williges, 1984).

- Communicate message category with (or before) the message (Merva and Williges, 1987). This can be accomplished through the use of different voices (or other form of auditory coding) (Smith and Goodwin, 1970; Simpson and Williams, 1980), or by selecting message structure in ways that define message category early in the message. (e.g., messages describing how to identify information of interest might use a standard form such as: "Displaying mobile SAMS blinking"). This aids in cueing the listener to the importance of messages, and limits the context of the message to improve understandability.

- ~~Present only information that is intended for immediate use (Simpson et al., 1985).~~

- Coordinate speech output with ongoing tasks (McCauley, 1984).

- Use only one modality when information must be integrated by the user (i.e., use only speech if speech is selected as the best media). Do not use a multi-modal expression under this condition (Wickens and Goettle, 1984). For example, if speed and altitude must be used together within a decision-making process, they should be presented in the same modality.

- Provide straightforward method for users to have message repeated. (Smith and Goodwin, 1987).

- Make sure that spoken messages are highly reliable (Simpson et al., 1985).

3. Computer Generated Speech and the CUBRICON Application

3.1 Categories of Generated Speech

Several categories of speech have been defined for use by the CUBRICON speech generation system. Rules for generating speech are being defined for each of these categories drawing from applicable literature. The categories are:

- Warnings. Computer initiated presentation of information that requires immediate action and which, if not acted upon, could lead to serious consequences.

- Advisories. Computer initiated presentation of information that is important ~~to~~ for the user to know but does not necessarily require immediate action.

- Dialogue. Presentation of information that is part of rapid two way exchanges. The category of "Dialogue" is further defined by the following subcategories:

    - Query response. Presentation of information in direct response to a user request. Query responses are always application specific (e.g., How many SAMS are there in the area of interest?).

      Note that it is possible for users to request dialogue guidance (defined below). Although, in this case this information is "requested", and therefore a response to a user query, it is treated under the category of "Dialogue guidance".

    - Dialogue guidance. Presentation of information to assist in the dialogue itself. Dialogue guidance can be of four types:

        - Feedback. Informing the user that user inputs have been accepted, understood, and so forth.

        - Focus. Providing information about the interpretation or use of presented information (e.g., telling the user where to look).

        - Prompt. Specific direction to the user about user inputs to facilitate efficient dialogue (e.g., system request for information).

        - Status. Informing the user about the status of the system/dialogue (e.g., "generating new map", "defining new task").

These generated speech outputs can be independent of other media/modalities, or as part of multimedia output, where generated speech provides explanation or enhancement of other outputs. In fact, all CUBRICON outputs can be classed within these categories. It should be noted that dialogue output can fall into more than one of the subcategories.

3.2 A Framework for Defining CUBRICON Speech Generation Logic

This section presents guidelines for deciding when and how to use computer generated speech within the context of the CUBRICON system. These guidelines are defined for each of the speech categories defined above. They are expressed in rule-like fashion to facilitate further specification, and incorporation within the CUBRICON system. References to literature upon which each implementation approach is based are also given. References are cited by number with a key given at the end of the section. A complete summary of each article reviewed as well as the complete reference is given in Appendix A. The following table presents these guidelines:

Summary of Rules for Using Generated Speech

Category	When to Use	How to Implement
Warnings	- Need to communicate information that requires immediate action which if not taken will lead to serious consequences.	- Precede message with alerting sound (e.g., beep). (Ref. 1,2,3) - Use high priority voice (Ref. 2,6). - State message twice (Ref. 4,5,7,9). - Keep message short Ref. 1,8) (4-5 syllables) (Ref. 1). - Refer to additional information (presented separately from warning). - Interrupt ongoing processes. If voice I/O is underway, break at logical breaking point (tbd). - Augment with visual display (Ref. 1).
Advisories	- Need to communicate information that is important to the user, but does not necessarily require immediate action.	- Use high priority voice. (Ref. 2,6). - Keep message as short as possible (Ref. 1,8) (no more than 10 words) - Refer to additional information (presented separately from warning). - Present at end of ongoing communication (most recent user request is satisfied) (i.e., don't interrupt, Ref. 10).

Category	When to Use	How to Implement
Dialogue	-	-
Query Response	- User has requested information which is best expressed, at least in part, using speech.	- Use low priority voice (Ref. 2.6). - Keep message as short as possible (Ref. 1,8). - Provide clear reference to related information on the screen (Ref. 10).
Dialogue Guidance	-	-
Feedback	- User has made inputs to the system, and, - There is no immediate system response that is perceivable by the user (e.g., requires time for processing, no response required).	- Use low priority voice (Ref. 2,6). - Keep message as short as possible (Ref. 1,8) (no more than 4 or 5 words).
Focus	- Expressions require amplifying or orientational information to assure proper understanding or use.	- Use low priority voice. (Ref. 2,6). - Keep message as short as possible (Ref. 1,8) (no more than 10 words). - Provide clear reference to expression being amplified (Ref. 10).
Prompt	- It is necessary to solicit information from the user (e.g., to allow completion of process, to clarify user request).	- Use low priority voice (Ref. 2,6). - Keep message as short as possible (Ref. 1,8) (no more than 4 or 5 words).
Status	- It is necessary to inform the user about the system or dialogue status (e.g., that a new task is being defined, system is performing a task).	

Key to References:

1) Simpson et al., 1985
2) Simpson and Williams, 1980
3) Hakkinen and Williges, 1984
4) Davis and Stockton, 1982
5) MIL-STD-1472C
6) Smith and Goodwin, 1970
7) Merva and Williges, 1987
8) Zoltan-Ford, 1984
9) Miller et al., 1951
10) McCauley, 1984

The following table summarizes the approach being implemented by CUBRICON for presenting messages of differing types/priority. Two voices are being used to distinguish between high and low priority messages. An auditory beep will be used to alert users when warnings are to be issued. Other presentation characteristics that distinguish the three message categories are also summarized in the following table.

CUBRICON Approach to Distinguishing Speech Categories

Category	Priority	Voice	Beep	State Message Twice	Augment with Visual	Interrupt Ongoing Process
Warnings	1	A	Y	Y	Y	Y
Advisories	2	A	-	-	D	-
Dialogue	3	B	-	-	D	-

KEY: A = Voice A (for messages of high relative importance)
B = Voice B (for messages of low relative importance)
1, 2, 3 = High, medium, and low priority, respectively
Y = Yes
- = No
D = Depends on specifics of message

3.3 Special CUBRICON Issues

This section addresses specific design issues related to CUBRICON, that are not specifically addressed in the literature. Each issue is presented with our rationale for the solution approach. This section will grow as we address new issues that come up as part of the CUBRICON design process (in fact we are presently confronting new speech related issues that will eventually be documented in this section).

3.3.1 The Use of Generated Speech to Support CUBRICON Introduction The capability to provide an introduction to mission plans is planned as part of CUBRICON. This introduction will fulfill two functions: 1) Provide an overview of the specific mission plan; and 2) Provide an introduction to the

CUBRICON interface itself (assuming those most likely to need an introduction will be infrequent CUBRICON users, such as unit commanders or higher-level Generals).

A mini-analysis of media/modalities that can best satisfy the first function of this mission plan introduction was performed. It was decided that the media for providing this function will need to be able to express hypothetical concepts and a broad overview of complex information, the combinatorial specifics of which cannot be predicted. The following conclusions concerning media to support this presentation were drawn:

1. Graphic portrayal at any level would provide too detailed a view and would need explanation and extrapolation to the larger view.

2. High-level tabular displays would be very useful since broad categories that define military planning can be pre-specified (and tailored for particular plans if necessary).

3. Speech is flexible but not structured. It may be useful in presenting high-level information about a mission plan that cannot be pre-specified or structured (e.g., as in a table). Speech may need to be supported by more structured media/modalities.

Consideration of the second function of the CUBRICON Introduction, to introduce the CUBRICON interface approach, has led to some slightly different conclusions:

1. A multi-media approach to presenting the introduction would best suit this function, since the CUBRICON interface itself integrates multi-media displays.

2. The literature indicates that training and practice significantly improves the ability to use synthetic speech (Cooper, 1987; Schwab et al., 1985). In fact, Cooper found that synthetic speech training that is coupled with visual feedback is best. The inclusion of a multi-media approach which includes synthetic speech would provide just this kind of embedded practice each time an introduction was presented (i.e., embedded, recurring training with visual feedback).

The above analysis provides strong rationale for using synthetic speech together with a tabular presentation for providing the CUBRICON Introduction to mission plans.

4. References

Bucher, N. M., Voorhees, J., and Werner, E. (1984). "Alerting prefixes for speech warning messages. In Proceedings of the National Aerospace and Electronics Conference (NAECON). New York: IEEE Press, 924-931.

Chapanis, A. (1975). Interactive human communication. Scientific American, 232 (3), 36-42.

Cooper, M. (1987). Human factor aspects of voice input/ output. Speech Technology, March/April 1987.

Davis, G. and Stockton, G. (1982). F-16 voice message system study. In NAECON Proceedings. New York: IEEE Press, 324-331

Deatherage, B. H. (1972). Auditory and other sensory forms of information presentation. In H. P. VanCott and R. G. Kinkade (eds.), Human Engineering Guide to Equipment Design. Washington, DC: U. S. Government Printing Office.

Hakkinen, M. T. and Williges, B. H. (1984). Synthesized warning messages: effects of an alerting cue in single- and multiple-function voice synthesis systems. Human Factors, 26 (2), 185-195.

Harvey, D. S. (1988). Talking with airplanes. Air Force Magazine. 88-96.

Luce, P. A., Feustel, T. C., and Pisoni, D. B. (1983). Capacity demands in short-term memory for synthetic and natural speech. Human Factors, 25 (1), 17-32.

~~Merva, M. A. and Williges, B. H. (1987). Context, repetition and synthesized speech intelligibility. In Proceedings of the Human Factors Society - 31st Annual Meeting. 961-965.~~

McCauley, M. E. (1984). Human factors in voice technology. In F. A. Muckler, A. S. Neal, and L. Strother (eds.) Human Factors Review: 1984. Santa Monica, CA: The Human Factors Society, Inc.

Merva, M. A. and Williges, B. H. (1987). Context, repetition and synthesized speech intelligibility. In Proceedings of the Human Factors Society - 31st Annual Meeting. 961-965.

MIL-STD-1472C. (1981). Military Standard: Human Engineering Design Criteria for Military Systems, Equipment and Facilities.

Miller, G. A., Heise, G. A., and Lichten, W. (1951). The intelligibility of speech as a function of the context of the test materials. Journal of Experimental Psychology, 41, 329-335.

Schwab, E. C., Nusbaum, H. C., Pisoni, D. B. (1985). Some effects of training on the perception of synthetic speech. Human Factors, 27 (4), 395-408.

Simpson, C. A. (1976). Effects of linguistic redundancy on pilots' comprehension of synthesized speech. In Proceedings of the Twelfth Annual Conference on Manual Control (TMX-73170). Moffett Field, CA: NASA, 294-308.

Simpson, C. A., and Marchionda-Frost, K. (1984). Synthesized speech rate and pitch effects on intelligibility of warning messages for pilots. Human Factors, 26 (5), 509-517.

Simpson, C. A., McCauley, M. E., Roland, E. F., Ruth, J. C., and Williges, B. H. (1985). System design for speech recognition and generation. Human Factors, 27 (2), 115-141.

Simpson, C. A. and Williams, D. H. (1980). Response time effects of alerting tone and semantic context for synthesized voice cockpit warnings. Human Factors, 22 (3), 319-330.

Smith, S. L. and Goodwin, N. C. (1970). Computer-generated speech and man-computer interaction. Human Factors, 12 (2), 215-223.

Smith, S. L. and Mosier, J. N. (1986). Guidelines for Designing User Interface Software. ESD-TR-86-278.

Wickens, C. D. and Goettle, B. (1984). Multiple resources and display formatting: The implications of task integration. In Proceedings of the Human Factors Society 28th Annual Meeting. 722-726.

Zoltan-Ford, E. (1984). Reducing variability in natural-language interactions with computers. In Proceedings of the Human Factors Society - 28th Annual Meeting. 768-772.

Summary of Literature on Speech Generation

Chapanis, A. (1975). Interactive human communication. *Scientific American*, 232 (3), 36-42.

Agency: Johns Hopkins University

Keywords: Problem solving (PS)
Voice communication (VC)

Notes: Describes experiments in human problem solving with various communication modes available including voice.

Results/I3 Design Implications:

The freedom to interrupt in voice and typed communications shortens problem solving time.

Human speech is characterized by mispronunciations, errors, and violations of format.

Problem solving requiring communication between humans is significantly enhanced when voice communication is permitted (with or without being able to see eachother). Video, handwriting, and typewriting combinations were all slower.

Cooper, M. (1987). Human factor aspects of voice input/ output. Speech Technology, March/April 1987.

Agency: British Telecom Technology, UK

Keywords: Speech generation (SG)
Speech recognition (SR)

Notes: Very brief review of speech generation (and recognition) literature . Focuses on intelligibility and naturalness.

Results/I3 Design Implications:

Synthetic speech more readily understood when context limits possible words.

Gramatically correct pauses improves performance with synthetic speech.

Performance with "surface" questions (questions about the text itself such as "was the word 'fairy' used?) was better with synthetic speech than natural speech, but worse with "higher level" questions. Synthetic speech may draw attention to actual words spoken.

Practice with synthetic speech with visual feedback improves performance.

Davis, G. and Stockton, G. (1982). F-16 voice message system study. In NAECON Proceedings. New York: IEEE Press, 324-331

Agency:

Keywords:

Notes: Recommendations based on analysis of interview data with operational pilots.

Article refered to in McCauley (1984), and is currently oredered.

Results/I3 Design Implication:

Use voice for warnings when: 1. immediate corrective action required; 2. conditions that could become critical exist.

Express warnings twice (for both above categories).

Deatherage, B. H. (1972). Auditory and other sensory forms of information presentation. In H. P. VanCott and R. G. Kinkade (eds.), _Human Engineering Guide to Equipment Design_. Washington, DC: U. S. Government Printing Office.

Agency: Joint Army, Navy, Air Force Steering Committee/ American Institutes for Research

Keywords: Auditory communication (AC)

Notes: General guidelines for design of auditory displays.

Results/I3 Design Implications:

Attached table summarizes recommendations. See article for more specific data.

----

4. Auditory and Other Sensory Forms of Information Presentation

4.1 The Use of Auditory Presentation

Some signals are better suited for auditory than for visual presentation and vice versa. The choice of which sense to use depends on: (a) the nature of the signal, (b) the conditions under which it must be received, and (c) the characteristics of the person involved. Table 4-1 summarizes situations in which one form is preferred over another.

TABLE 4-1. WHEN TO USE THE AUDITORY OR VISUAL FORM OF PRESENTATION

Use auditory presentation if:	Use visual presentation if:
1. The message is simple.	1. The message is complex.
2. The message is short.	2. The message is long.
3. The message will not be referred to later.	3. The message will be referred to later.
4. The message deals with events in time.	4. The message deals with location in space.
5. The message calls for immediate action.	5. The message does not call for immediate action.
6. The visual system of the person is overburdened.	6. The auditory system of the person is overburdened.
7. The receiving location is too bright or dark-adaptation integrity is necessary.	7. The receiving location is too noisy.
8. The person's job requires him to move about continually.	8. The person's job allows him to remain in one position.

4.1.1 Signals Suited to Auditory Presentation

Auditory presentation is preferred over visual presentation:

1. For signals of acoustic origin. Ingenious visual displays for speech have been devised, but none is likely to supplant hearing except when deafness or intense noise conditions render hearing useless.

2. For warning signals. A visual warning signal must be seen in order to warn. On the other hand, hearing is omnidirectional and cannot be involuntarily shut off. It is therefore best for calling attention to imminent or potential danger.

3. For situations when too many displays are visually presented—in piloting an airplane, for example.

4. For presenting information independently of head orientation—as when duties require body movement or head turning.

5. For situations when darkness limits vision or makes seeing impossible.

6. For conditions of anoxia in high altitudes or high positive g forces; auditory sensitivity is more resistant to anoxia than is visual sensitivity. A man suffering from oxygen deficiency may have his vision seriously impaired, but he can still hear signals.

7. When signals must be distinguished from noise. The ear acts as a frequency analyzer, making it an effective detector of periodic signals in noise. Even when it is considerably weaker than the background noise, if the signal is a sinusoid (pure tone) or a combination of sinusoids (complex tones), the ear can detect it. The ear also efficiently detects periodic modulation in the very-low-frequency range and responds to variations in intensity or frequency.

4.1.2 Choosing the Form of Auditory Presentation

Use tonal or noise signals, rather than speech:

1. For simplicity.

2. When listeners are trained to understand coded signals.

3. For designating a point in time that has no absolute value.

4. When immediate action is desired.
5. In conditions unfavorable for receiving speech messages. (Tonal signals can be heard at noise levels where speech is unintelligible.)
6. When security of the message is desired; coded tonal or noise signals may be used.
7. If speech communication channels are overloaded.
8. If speech will mask other speech signals or annoy listeners for whom the message is not intended.

Use speech rather than tonal or noise signals under these conditions:

1. For flexibility.
2. To identify a message source.
3. When listeners are without special training in coded signals.
4. There is a necessity for rapid two-way exchanges of information.
5 The message deals with a future time requiring some preparation. (Example: The countdown preparatory to firing a missile—tonal signals could be miscounted.)
6. Situations of stress might cause the listener to "forget" the meaning of a code.

4.1.3 Some Common Uses for Auditory Presentation

Speech is best for transmitting urgent messages, since the maximum transmission rate of speech is 250 words per minute (wpm). Morse code, on the other hand, is intelligible under low signal-to-noise ratios, but the maximum rate of transmission is about 30 wpm.

Noise signals are best for sea and air navigation. Lighthouse diaphones that pulse messages in Morse code to indicate position are audible for long distances; whistling and bell buoys can locate channels and shoals; radio-range signals and fan-marker radio beacons can mark airways. Other types of signals, detecting, echo ranging, warning, and alarm signals, are commonly auditory.

Table 4–2 summarizes the principal characteristics and special features of different types of auditory alarm and warning signals. Examples given are for horns, whistles, sirens, bells, buzzers, chimes, gongs, and oscillators.

Design recommendations. The following principles for selection and design of alarm and warning signals should be observed (see also Table 4-3):

1. At a minimum, use sounds having frequencies between 200 and 5000 Hz, and if possible, between 500 and 3000 Hz, because the human ear is most sensitive to this middle range.
2. Use sounds having frequencies below 1000 Hz when signals must travel long distances (over 1000 ft.) because high frequencies are absorbed in passage and hence cannot travel as far. Figure 4–1 shows attenuation of sounds of various frequencies in calm air for distances from 10 to 10,000 ft. under conditions free from the effects of reflecting surfaces and obstacles.
3. Use frequencies below 500 Hz when signals must bend around obstacles or pass through partitions. (See Figure 4-2.)
4. In noise, signal frequencies different from those most intense frequencies of the noise are best in order to reduce masking of the signal.
5. Use a modulated signal to demand attention. Intermittent beeps repeated at rates of one to eight beeps per second, or warbling sounds that rise and fall in pitch are seldom encountered, and are therefore different enough to get immediate attention. If speech is necessary during an alarm, use an intermittent, pure-tone signal of relatively high frequency.
6. Use complex tones rather than pure sinusoidal waves, because few pure tones can be positively identified but each complex sound is noticeably different from other sounds.

4.1.4 When to Use Auditory Displays

Certain considerations are helpful in deciding when to use an auditory display for spatial information:

1. Use auditory displays to relieve the eyes. Although the eye is better for spatial discrimination, it can look in only one direction at a time. In general, auditory spatial displays are recommended only when the eyes are fully engaged and additional spatial information is needed.
2. Use auditory displays (other than speech) to present restricted information, such as the following:

Hakkinen, M. T. and Williges, B. H. (1984). Synthesized warning messages: effects of an alerting cue in single- and multiple-function voice synthesis systems. Human Factors, 26 (2), 185-195.

Agency: Virginia Polytechnic Institute; State University, Blacksburg, VA

Keywords: Speech generation (SG)
Warnings (W)

Notes: Laboratory experiment
Simplified air traffic control task

Results/I3 Design Implications:

When synthesized speech used for emergency messages and non-emergency messages, an alerting tone preceding the emergency messages improved detection.

When synthesized speech used for only emergency messages, an alerting tone preceding the emergency messages lengenthed response time to the message.

Harvey, D. S. (1988). Talking with airplanes. Air Force Magazine. 88-96.

Agency: Based on interviews with Air Force researchers (e.g., Dr. Tom Furness at AAMRL).

Keywords: Speech recognition (SR)

Notes: Describes applied experience with in-cockpit voice recognition systems.

Results/I3 Design Implications:

Provides list of 656 words sufficient for voice control in air-to-air and air-to-ground tactical situations.

Luce, P. A., Feustel, T. C., and Pisoni, D. B. (1983). Capacity demands in short-term memory for synthetic and natural speech. Human Factors, 25 (1), 17-32.

Agency: Speech Research Laboratory, Indianna University

Keywords: Speech generation (SG)

Notes: Laboratory experiments.
Recall of word lists using synthetic and natural speech.

Results/I3 Design Implications:

Shows evidence that there is a short-term memory penalty for synthetic speech when compared to natural speech.

Merva, M. A. and Williges, B. H. (1987). Context, repetition and synthesized speech intelligibility. In Proceedings of the Human Factors Society - 31st Annual Meeting. 961-965.

Agency: AT&T Consumer Products; Virginia Polytechnic Institute

Keywords: Speech generation (SG)

Notes: Laboratory experiment.
Subjects were asked to transcribe what they heard using synthetic speech.

Results/I3 Design Implications:

Transcription accuracy best when the context of the message was known prior to the message (e.g., when subjects knew the message would be about the weather).

Transcription accuracy best for 150 to 180 words per minute. However, experienced users were able to use faster rates accurately. Perhaps use faster speech rates for experienced users.

Transcription accuracy best for messages repeated twice. Repeating a message three times did not further improve performance. Without repetitions perceptions were correct 95% of of the time. Recommend a user-initiated repeat function (since subjects were aware when they made a mistake).

Words at end of messages were transcribed more accurately than words at the beginning of messages. Put most important information near the end of messages.

Accuracy transcribing numerical information was better than for verbal (i.e., non-numerical).

McCauley, M. E. (1984). Human factors in voice technology. In F. A. Muckler, A. S. Neal, and L. Strother (eds.) Human Factors Review: 1984. Santa Monica, CA: The Human Factors Society, Inc.

Agency: Essex Corp.

Keywords: Speech generation (SG)
Speech recognition (SR)

Notes: Provides broad review of speech generation and recognition technology. Includes human factors considerations for its use. Major focus is on speech recognition.

Results/I3 Design Implications:

When arranged in meaningful sentences, differences in synthetic natural speech understanding are minimal. However, under task loading synthetic speech understanding degrades more quickly than natural speech.

Don't interupt. System speech should not be used when human user is speaking. Care should be taken in temporal coordination of speech synthesis with ongoing tasks (p. 148).

Useful when eyes (and hands) are occupied and unavailable for communicating.

MIL-STD-1472C. (1981). Military Standard: Human Engineering Design Criteria for Military Systems, Equipment and Facilities.

Agency: Department of Defense

Keywords: Speech generation (SG)

Notes: General standard covering broad range of human factors issues applied to system design.

Results/I3 Design Implications:

See attached pages from MIL-STD-1472C.

----

f. Signals that resemble random noise, periodic pulses, steady or frequency modulated simple tones, or any other signals generated by standard countermeasure devices (e.g., "bagpipes").

g. Signals similar to random noise generated by air conditioning or any other equipment.

5.3.4.4 Compatibility.

5.3.4.4.1 Existing Signals. The meaning of audio warning signals selected for a system should be consistent with warning signal meanings already established for that function.

5.3.4.4.2 Acoustic Environment. Established signals shall be used, provided they are compatible with the acoustic environment and the requirements specified herein for the voice communication system. Standard signals shall not be used to convey new meanings.

5.3.4.5 Masking.

5.3.4.5.1 Other Critical Channels. Audio warning signals shall not interfere with any other critical functions or warning signals, or mask any other critical audio signals.

5.3.4.5.2 Separate Channels. Where a warning signal delivered to a headset might mask another essential audio signal, separate channels may be provided to direct the warning signal to one ear and the other essential audio signal to the other ear. In such a situation and when required by operating conditions, this dichotic presentation may further provide for alternation of the two signals from ear to ear.

5.3.5 Verbal Warning Signals.

5.3.5.1 Nature of Signals. Verbal warning signals shall consist of:

a. An initial alerting signal (nonspeech) to attract attention and to designate the general problem.

b. A brief standardized speech signal (verbal message) which identifies the specific condition and suggests appropriate action.

5.3.5.2 Intensity. Verbal alarms for critical functions shall be at least 20 dB above the speech interference level at the operating position of the intended receiver.

5.3.5.3 Vocal Criteria.

5.3.5.3.1 Type of Voice. The voice used in recording verbal warning signals shall be distinctive and mature.

5.3.5.3.2 Delivery Style. Verbal warning signals shall be presented in a formal, impersonal manner.

5.3.5.4 Speech Processing. Verbal warning signals shall be processed only when necessary to increase or preserve intelligibility, such as by increasing the strength of consonant sounds relative to vowel strength. Where a signal must be relatively intense because of high ambient noise, peak-clipping (see 3.24) may be used to protect the listener against auditory overload.

5.3.5.5 Message Content. In selecting words to be used in audio warning signals, priority shall be given to intelligibility, aptness, and conciseness in that order.

5.3.5.6 Message Categories.

5.3.5.6.1 Critical Warning Signals. Critical warning signals shall be repeated with not more than a 3-second pause between messages until the condition is corrected or overridden by the crew.

5.3.5.6.2 Message Priorities. A message priority system shall be established and more critical messages shall override the presentation of any message occurring below it on the priority list. If two or more incidents or malfunctions occur simultaneously, the message having the higher priority shall be given first. The remaining messages shall follow in order of priority. In the event of a complete subsystem failure, the system shall integrate previous messages via electronic gathering and report the system rather than the component failure.

5.3.6 Controls for Audio Warning Devices.

5.3.6.1 Automatic or Manual Shut-off. When an audio signal is designed to persist as long as it contributes useful information, a shut-off switch controllable by the operator, the sensing mechanism, or both, shall be provided, depending on the operational situation and personnel safety factors.

5.3.6.2 Automatic Reset. Whether audio warning signals are designed to be terminated automatically, by manual control, or both, an automatic reset function shall be provided. The automatic reset function shall be controlled by the sensing mechanism which shall recycle the signal system to a specified condition as a function of time or the state of the signaling system.

5.3.6.3 Volume Control.

5.3.10.2 Squelch Control. Where communication channels are to be continuously monitored, each channel shall be provided with a signal-activated switching device (squelch control) to suppress channel noise during no-signal periods. A manually operated, on-off switch, to de-activate the squelch when receiving weak signals, shall be provided.

5.3.10.3 Foot-operated Controls. When normal working conditions will permit the operator to remain seated at the working position and access to "talk-listen" or "send-receive" control switches is required for normal operation or if console operation requires the use of both hands, foot-operated controls shall be provided. Hand-operated controls for the same functions shall be provided for emergency use and for use when the operator may need to move from one position to another.

5.3.11 Speaker/Side Tone. The speaker's verbal input shall be in phase with its reproduction as heard on the headset. This side tone should not be filtered or modified before it is received in the headset.

5.3.12 Speech Intelligibility.

5.3.12.1 General. When information concerning the speech intelligibility of a system is required, three recommended methods are available, with the appropriate selection being dependent upon the requirements of the test:

a. The ANSI standard method of measurement of phonetically balanced (PB) monosyllabic word intelligibility, S3.2-1960, should be used when a high degree of test sensitivity and accuracy is required.

b. The modified rhyme test (MRT) (see Human Engineering Guide to Equipment Design) should be used if the test requirements are not as stringent or if time and training do not permit the use of the ANSI method.

c. The articulation index (AI) calculations should be used for estimations, comparisons and predictions of system intelligibility based upon ANSI S3.5-1969.

5.3.12.2 Criteria. The intelligibility criteria shown in Table VI shall be used for voice communication. The efficiency of communications needed and the type material to be transmitted shall determine which of the three communication requirements of Table VI is to be selected.

TABLE VI. INTELLIGIBILITY CRITERIA FOR VOICE COMMUNICATIONS SYSTEMS

COMMUNICATION REQUIREMENT	SCORE		
	PB	MRT	AI
Exceptionally high intelligibility; separate syllables understood	90%	97%	0.7
Normally acceptable intelligibility; about 98% of sentences correctly heard; single digits understood	75%	91%	0.5
Minimally acceptable intelligibility; limited standardized phrases understood; about 90% sentences correctly heard (not acceptable for operational equipment)	43%	75%	0.3

Miller, G. A., Heise, G. A., and Lichten, W. (1951). The intelligibility of speech as a function of the context of the test materials. Journal of Experimental Psychology, 41, 329-335.

Agency: Harvard University

Keywords: Voice communication (VC)

Notes: Laboratory experiment
Studied the effect of various types of context on speech intelligibility. Speech was generated by humans speeking through a sound system.

Results/I3 Design Implications:

The effect of context on intelligibility was to make speech intelligible with lower signal to noise ratios. Types of context: limited vocabulary; surrounding sentence structure; and repetition (repeating the word or message).

Schwab, E. C., Nusbaum, H. C., Pisoni, D. B. (1985). Some effects of training on the perception of synthetic speech. Human Factors, 27 (4), 395-408.

Agency: Speech Research Laboratory, Department of Psychology, Indiana University, Bloomington, IN

Keywords: Speech Generation (SG)
Training (T)

Notes: Laboratory study

Results/I3 Design Implications:

Training in understanding synthetic speech had significant, positive effect on performance in use of synthetic speech. Effects of training apparent even after six months.

Simpson, C. A., and Marchionda-Frost, K. (1984). Synthesized speech rate and pitch effects on intelligibility of warning messages for pilots. Human Factors, 26 (5), 509-517.

Agency: Psycho-Linguistic Research Associates

Keywords: Speech generation (SG)

Notes: Laboratory experiment
Simulated helicoptor tasks with synthesized warning messages.

Results/I3 Design Implications:

156 words per minute speech rate yields high intelligibility and response time, and is preferred by pilots. Higher rates yielded faster response times but may have required higher cognitive load.

Voice pitch in the range of 90 to 120 Hz will be judged as alerting. There was no effect of pitch on intelligibility.

Simpson, C. A., McCauley, M. E., Roland, E. F., Ruth, J. C., and Williges, B. H. (1985). System design for speech recognition and generation. Human Factors, 27 (2), 115-141.

Agency: Psycho-Linguistic Research Associates; Monterey Technologies; Rolands and Associates; McDonnell-Douglas; Virginia Polytechnic Institute and State University, Blacksburg, Virginia

Keywords: Speech generation (SG)
Speech recognition (SR)

Notes: Reviews literature and offers general guidelines.

Results/I3 Design Implications:

Definitions:

- Warnings -- Presentation of information which requires immediate action and which if not communicated could seriously degrade subsequent efforts or lead to major problems.

- Advisories -- Presentation of information that is not specifically requested but which is important for properly managing the ongoing efforts.

- Responses to Queries -- Presentation of information that is specifically requested by the human user.

- Feedback from Control Inputs -- Presentation of information to the human user that informs him or her that preceding inputs have been accepted, understood, etc.

- Prompts -- Short messages that aid in data entry. These are very similar to advisories but are distinguished by their routineness with respect to predefined procedures or data entry sequences.

When to Use Speech:

- Flexibility is required, i.e., message content cannot be predicted.

- N/A

- Listeners will have no special training of coded signals.

- When rapid two-way exchanges are required.

- When the message deals with a future time, requiring preparation.

- Situations of stress which might cause the operator to forget the meaning of coded signals.

When to Use Auditory Signals:

- When signal must ellicit attention (e.g., warnings).
- There are too many visual displays.
- Information must be presented independantly of head movement (e.g., the human user cannot look away from the current task to see information presented visually).

DESIGN OF SPEECH DISPLAYS

- Spoken information should be highly reliable.
- Information should be intended for immediate use.
- Speech messages should be kept as short as possible.
- Give voice a distinct sound (e.g., machine voice identifies the machine as the source, different voices indicate different functions or priorities or sources.

Speech Displays for Specific Purposes:

- Warnings:
  - Word in short phrases (e.g., 4-5 syllables).
  - Precede with beep (or other alerting sound) when voice is also used for other message types.
  - Augment with visual (for best results).
- Advisories:
  - Little research available. If used with warnings, consider distinguishing the two (e.g., with a beep).
- Responses to user queries:
  - When conditions for using voice requirements are met (and the human user request itself does not specifically ask for non-voice output).

  Feedback:
  - Can also use prompting as feedback.
  - When conditions for using voice requirements are met.
- Prompts:
  - Use short message.
  - Use terse style.

Simpson, C. A. and Williams, D. H. (1980). Response time effects of alerting tone and semantic context for synthesized voice cockpit warnings. Human Factors, 22 (3), 319-330.

Agency: Psycho-Linguistic Research Associates

Keywords: Speech generation (SR)

Notes: Experiment with voice warnings in a cockpit environment. A fixed-base simulator was used.

Results/I3 Design Implications:

Cites literature showing that voice warning system with a visual annunciator produces faster response times and reduced workload compared to either a visual system alone or a visual system with an alerting tone.

An alerting tone preceding voice warnings slowed response time. The lenghtening of the voice warning with additional context words did not slow response time.

Humans are sensitive to changes in voice quality (e.g., natural to synthetic, male to female). They are not sensitive to changes in language (English to German).

Recommendations: 1) Voice warnings should be in a vioce that is qualitatively different from other voices in the environment; 2) If synthesized speech is used exclusively for warnings, there should be no alerting tone; 3) If synthesized voice is used for more than just warnings, some means of elliciting attention to the warnings is needed.

Smith, S. L. and Goodwin, N. C. (1970). Computer-generated speech and man-computer interaction. Human Factors, 12 (2), 215-223.

Agency: MITRE Corporation

Keywords: Speech Generation (SG)

Notes: Abstract attached.

Results/I3 Design Implications:

Allow for user pacing and optional repetition of speech output.

Use different voices (or other auditory coding) to distinguish types of output

----

ning self-taught. *Computers and*
16(10), 30-35.
achine communication. In Carl-
Annual review of information
logy. New York: John Wiley &
54.
The TINT users' guide. Santa
System Development Corpora-
'03, July 30, 1965.
W. P., Jr., and Alexander, S. N.
ieering handbook. New York:
criptions and scheduling in the
ring system. Santa Monica, Cali-
Development Corporation,
1966.
ental investigation of user per-
ared computing systems. Santa
System Development Corpora-
6, 1967.
d, M. M. Time-sharing versus
n experimental inquiry into hu-
ig. System Development Cor-
gie-Mellon University, SP-3110,

HUMAN FACTORS, 1970, 12(2), 215–223

Computer-Generated Speech and Man-Computer Interaction

SIDNEY L. SMITH *and* NANCY C. GOODWIN, *The MITRE Corporation, Bedford, Massachusetts*

Large-scale use of talking computers may be anticipated for reasons of general convenience and special effectiveness. A telephone link to a computer, with touch-tone inputs and stored speech outputs, poses a number of design problems of interest for the human factors specialist. Recommended features include (1) user pacing and optional repetition of speech output in a transactional sequence, (2) use of different voices and other auditory coding to distinguish types of output, and (3) use of tone codes to indicate required input.

A physician is seated at his desk. He has just placed a telephone call. But the friendly female voice that answered his call no longer belongs to the woman who once spoke these words. It is now the voice of a computer speaking on the telephone:

> This is the acidosis program. Do you want descriptive information about this program? (dah-beep)

The opening speech, although it sounds natural enough, ends with an interrogatory tone-code that sounds like *dah-beep*. The computer is asking the physician to reply.

Because the physician is unfamiliar with the acidosis program, he replies affirmatively to the computer's question, and the program continues:

> The acidosis program was written originally by Doctors William Schwartz, David Goldman, and Jerome Kassirer of the New England Medical Center, Boston, Massachusetts. (beep) [1]

The computer concludes its speech with a declarative *beep*, and then waits. The physician could wait, too, if he were interrupted by other work, but now he signals the computer to go on:

> The program has been adapted in its present form by members of the MITRE Corporation. This program is intended to assist in the diagnosis of acidosis. The program consists of three general steps. (beep)

Still more speeches follow, as the physician continues his interaction with the talking computer:

> The first step is the entry of CO_2 and pH data to compute partial pressure of CO_2 and to determine the presence of an acidosis condition. (beep)

> The second step is the entry of serum sodium and chloride data to compute anion gap. (beep)

> The third step is the entry of blood sugar, serum ketones, salicylate, and creatinine data to analyze acidosis conditions associated with a widened anion gap. (beep)

Although no physician is using this acidosis program in his practice today, the interaction process dramatized above represents a fully operable demonstration of the future potential of computer aids in medicine.[2] The acidosis program also demonstrates access to a computer using a simple terminal device, the telephone.

---

[1] The attribution of credit for design of the acidosis program is valid; this program represents just one aspect of Dr. Schwartz' interest in the potential application of new technology in medicine.

[2] The speech output version of the acidosis program has been demonstrated at a number of recent professional meetings (Smith and Goodwin, 1968; 1969a; 1969b; 1969c).

Smith, S. L. and Mosier, J. N. (1986). Guidelines for Designing User Interface Software. ESD-TR-86-278.

Agency: MITRE Coorporation, Bedford, MA 01730

Keywords: General human-computer interface (GHCI)

Notes: Literature review

Results/I3 Design Implications:

Contains general human-computer interface design guidelines, very little specifically dealing with computer generated speech.

Wickens, C. D. and Goettle, B. (1984). Multiple resources and display formatting: The implications of task integration. In _Proceedings of the Human Factors Society 28th Annual Meeting_. 722-726.

Agency: University of Illinois

Keywords: Speech generation (SG)
Information display (ID)

Notes: Laboratory experiment
Compared performance on air traffic control task when presentation and pilot request modality were varied.
Study used few subjects and offered limited power.
More research was recommended.

Results/I3 Design Implications:

When information must be integrated, it should be presented using a single modality.

The competition for processing capacity may provide a counter-tendancy when a single modality is used (but the positive effects seem to outweigh this one negative).

Zoltan-Ford, E. (1984). Reducing variability in natural-language interactions with computers. In _Proceedings of the Human Factors Society - 28th Annual Meeting_. 768-772.

Agency:

Keywords: Speech Generation (SG)
Speech Recognition (SR)

Notes: Laboratory experiment.
Varied allowable voice (or typed) input and computer voice (or CRT) output.
Measured user satisfaction and input efficiency.

Results/I3 Design Implications:

Users voice inputs will conform in style and vocabulary to the style and vocabulary used by the computer in forming voice outputs.

Provide consistently worded program output: users will model it.

Design the program to communicate tersly (e.g., verb-noun) rather than more conversational styles (e.g., pronoun-modal auxiliary-verb-determiner-noun).

Provide nonthreatening error messages that reiterate those vocabulary and/or phrases the the processor can understand.

Speech is natural to humans and requires less training than other forms of interaction, but some training is still required. "Just because people know how to drive a car does not mean that they can drive cross-country without a map."

APPENDIX G
WORKING PAPER DESCRIBING LOCATIVE REFERENCING FOR MAP-BASED SYSTEMS

This appendix contains a paper describing research conducted in association with CUBRICON. This research resulted in the implementation of a definitive referencing capability for CUBRICON. While this parallel effort was funded by another agency, it was closely coordinated with CUBRICON development efforts, and in fact used CUBRICON as an implementation vehicle.

(forthcoming)

APPENDIX H
REFERENCES TO PUBLISHED TECHNICAL PAPERS DESCRIBING CUBRICON AND THE RESEARCH CONDUCTED UNDER THE INTELLIGENT INTEGRATED INTERFACES PROJECT

This appendix contains a list of CUBRICON related technical papers that were published in technical journals and conference proceedings. These papers were written and presented through the support of the Intelligent Integrated Interfaces Project.

- Technical papers/publications/presentations:

— "Intelligent Integrated Interface Technology," by J.G. Neal, S.C. Shapiro, Y. Smith. In the *Proceedings of the 1987 Tri-Service Data Fusion Symposium,* JHU-APL, Laurel, MD

— "Intelligent Multi-Media Interface Technology," by J.G. Neal and S.C. Shapiro. In the *Proceedings of the AAAI Workshop on Architectures for Intelligent Interfaces,* March 1988. To appear in *Architectures for Intelligent Interfaces: Element and Prototypes,* J. Sullivan & S. Tyler (Eds.), Addison-Wesley Publishing Co.

— "An Intelligent Multi-Media Human--Computer Dialogue System," by J.G. Neal, K. Bettinger, J.S. Byoun, Z. Dobes, C.Y. Thielman. In the *Proceedings of SOAR-88,* sponsored by USAF, NASA, and Wright State University, Dayton, OH, July, 1988.

— "Multi-Modal References in Human-Computer Dialogue," by J.G. Neal, Z. Dobes, K.E. Bettinger, and J.S. Byoun. In the *Proceedings of AAAI-88,* St. Paul, MN, August 1988

— "Multi-Modal Output Composition for Human-Computer Dialogues," by J.G. Neal, C.Y. Thielman, D.J. Funke, and J.S. Byoun. In the *Proceedings of the 1989 Artificial Intelligence Systems in Government Conference,* Washington, DC, March 1989

— "Natural Language with Integrated Deictic and Graphic Gestures," by J.G. Neal, C.Y. Thielman, Z. Dobes, S.M. Haller, and S.C. Shapiro. In the *Proceedings of the 1989 DARPA Workshop on Speech and Natural Language,* Harwich Port, MA, October 1989

— "CUBRICON: A Multi-Modal User Interface," by J.G. Neal, C.Y. Thielman, Z. Dobes, S.M. Haller, S. Glanowski, and S.C. Shapiro. In the *Proceedings of the GIS/LIS '89 Conference,* Orlando, FL, November 1989

— *The Intelligent Multi-Media Interfaces Project: Final Report,* by G.J. Neal et. al., in preparation.

— "CUBRICON: A Knowledge-Based Multi-Modal Interface System," by J.G. Neal, C.Y. Thielman, J. Lamnens, S.M. Haller, S.C. Shapiro. Journal paper in preparation.