

Report 83-34
Stanford -- KSL

Scientific DataLink

Intelligent Presentation: The Generation
Problem for User Interfaces.
Jock Mackinlay,
Mar 1983

card 1 of 1

Heuristic Programming Project
Report No. HPP-83-34

March 1983

**Intelligent Presentation:
The Generation Problem for User Interfaces**

Jock Mackinlay

Thesis Proposal

Heuristic Programming Project
Department of Computer Science
Stanford University

Table of Contents

1. Introduction	1
2. Motivation	3
3. Hypothetical Examples of Presentation	6
3.1. Introduction	6
3.2. Domain	7
3.3. Functional Ontology	8
3.4. Communication Goal Ontology	8
3.5. Descriptive Frames for Presentation Knowledge Sources	9
3.6. First Example: A Function with Degree One	10
3.7. Second Example: Segmenting	13
3.8. Extension to Relations and Their Graphs	18
3.8.1. Relational Ontology	18
3.8.2. Graph Example	18
3.8.3. Dynamic Presentation	20
3.9. Extension to Maps and Diagrams	20
4. Discourse Focus as a Source for Communication Goals	22
5. Related Work	24
5.1. Information Presentation	24
5.2. Synthesis Domains	24
5.3. Knowledge Representation	25
5.4. Human Factors and User Interface Design	26
5.5. Computational Linguistics	27
6. Conclusion and Schedule	28

List of Figures

Figure 3-1: Hypothetical Database Relations	7
Figure 3-2: Description of NumStudent	10
Figure 3-3: Descriptive Frame for the Table Knowledge Source	10
Figure 3-4: Sample Table for NumStudent	10
Figure 3-5: Descriptive Frame for Pie Knowledge Source	11
Figure 3-6: Sample Pie Chart for NumStudent	11
Figure 3-7: Descriptive Frame for Bar Knowledge Source	12
Figure 3-8: Sample Bar Chart for NumStudent	12
Figure 3-9: Descriptive Frame of the Plot Chart Knowledge Source	12
Figure 3-10: Agenda for NumStudent with no Communication Goals	13
Figure 3-11: Hypothetical Data for the Number of Students in Degree Programs	14
Figure 3-12: Description of NumYear	14
Figure 3-13: Descriptive Frame for the Pie-Segment Knowledge Source	14
Figure 3-14: Sample Pie Charts for NumYear	15
Figure 3-15: Descriptive Frame for the Bar-segment Knowledge Source	15
Figure 3-16: Sample Bar Chart for NumYear	15
Figure 3-17: Sample Layered Bar Chart for NumYear	16
Figure 3-18: Descriptive Frame for the Plot-segment Knowledge Source	16
Figure 3-19: Sample Plot Chart for NumYear	17
Figure 3-20: Sample Layered Plot Chart for NumYear	17
Figure 3-21: Hypothetical Data for Prerequisite Relation	18
Figure 3-22: Relational Ontology	19
Figure 3-23: Example Graph	19
Figure 3-24: Additional Data about Class Prerequisites	20

1. Introduction

Effective communication is fundamental to many definitions of intelligence, including Turing's test [23]. The communication between computers and humans, which is only one aspect of this general topic, is the basic concern of this research. Effective communication requires competence in two related areas: interpretation of messages and generation of messages. Effective generation must take into account how messages are going to be interpreted, and successful interpretation must take into account how and why messages are generated. This thesis proposal describes research on the generation of effective communication by a computer system for a human user of that system. The generation problem, a basic problem for user interface designers, has often been neglected because people are very good interpreters and with training can adapt to very unusual linguistic conventions. With the increasing number of computer systems and untrained/unsophisticated users, it is becoming imperative that computer systems make themselves better understood. The proposed research approaches user interface design with the premise that, during communication, computer systems and users should be equal partners. The major goal of the proposed thesis is to build a knowledge-based user interface that generates effective presentations of information.

The difference between this research and similar generation research by Appelt and McKeown in computational linguistics [2, 15] is the focus on graphical languages rather than on natural languages. Natural language, a complex and versatile communication medium, is not always the best medium for user interfaces. Graphical languages or *modalities* allow the user interface to construct messages that are quickly grasped by users. Some examples of graphical modalities for user interfaces are tables, icons, trees, pie charts, and bar charts. Although graphical languages are more stylized than natural language, the user interface that uses graphical languages is still confronted with the basic demands of carrying on a dialogue with the user. Computational linguistics has developed various techniques for carrying on a dialogue. One goal of this proposed research is to adapt these techniques to a domain that has multiple languages. A question the proposed thesis will try to answer is how to choose the appropriate language for communicating some information. User interfaces can be designed to use multiple media like graphics, voice, and pointing devices. The techniques developed here for choosing graphical languages represent the first step towards systems that can construct effective multi-media presentations.

In this thesis proposal, the generation problem is called *information presentation* following the work at BBN [24]. Knowledge-based expert systems for doing generation are called *information presentation systems*. The particular system described here is called the *Intelligent Information Presentation System* or IIPS where the system's "intelligence" is embodied in linguistic notions called

communication goals. The addition of communication goals to the presentation process is a major contribution of the proposed research.

In IIPS there are three types of input for information presentation:

These three types of input are used by IIPS during a heuristically guided search for an effective presentation. The basic architecture of the proposed system is derived from the blackboard model. Each knowledge source is responsible for particular presentation technique. The structure of the information and the communication goals are on the blackboard for the knowledge sources to use.

This proposal describes a sequence of steps for building the proposed knowledge-based user interface. The first step is to solve the problem of *simple presentation*, which is to construct solitary presentations. This step avoids any difficulties associated with carrying on a conversation with the user. The second step is to extend the system to handle *dynamic presentation*, where a presentation must be modified to present new information and achieve new communication goals. In this case, the presentation problems are more difficult but the conversational demands generate additional communication goals that can be used to search for an effective presentation. The final step is to incorporate IIPS into the rest of the application system by developing a mechanism for generating communication goals. Once the system is built, it will be evaluated with an *ablation study*, in which a comparison is done between the complete IIPS and a version missing some of its knowledge.

Section 2 describes the basic motivation for this research. In particular, it touches on computational linguistics, user interface design, expert system explanation facilities and knowledge representation. Section 3 gives some examples of IIPS solving some simple presentation problems for numerical data. Subsections 3.8 and 3.9 sketch the extension of IIPS to symbolic data and dynamic presentation. Section 4 describes a simple mechanism based on computational linguistic techniques for keeping track of the focus of conversation. This mechanism, called a *Dialogue Management System* or DMS, generates communication goals for IIPS. Section 5 reviews related research. The final section gives a proposed research schedule and concludes the discussion.

2. Motivation

Communication is fundamental to many definitions of intelligence. The study of communication is a vast topic with relevant work in many areas, including linguistics, philosophy, and psychology. Most definitions of communication include some notion of purpose. The basic model involves a speaker and a hearer. The speaker has a goal that s/he wants to achieve by communicating with the hearer. S/he uses many different kinds of knowledge, including knowledge about linguistic techniques, the hearer's beliefs, and the preceding conversation to construct a message. When the message is heard, the hearer interprets it using many different kinds of knowledge, including the fact that the speaker constructed the message with some purpose. This model led Austin [3] to formulate the *speech act theory* that accounts for the fact that speakers are doing something with their utterances. Computational linguistics research has used planning technology to build computational models of communication based on this linguistic theory. Cohen [5] and Allen [1] developed models for interpretation that use the fact that the speaker had some purpose. Appelt [2] developed a model for generation that attempted to achieve multiple goals while constructing a message.

Purpose plays an important role in human/computer communication. Users of computer systems are task oriented: they expect that the output of a computer system relates to their task. Human/computer communication can break down due to "purpose related" problems during interpretation and generation. Although the correct interpretation of a user's purpose is a difficult problem for user interfaces, there are a number of techniques that can be used to avoid most of the difficulties. First, command languages can be designed that are easy for the user to use but precise enough for the user interface to interpret without difficulty. Second, user interfaces can obtain information by asking questions. The context of the question makes interpretation easier. Third, good generation facility can explain the system's interpretation of some input so that the user can correct any errors.

On the other hand, the generation of obscure or misleading messages by the computer system is a more difficult problem to avoid. A typical user interface designer must anticipate the information's structure and the presentation goals for every communication situation, and write a user interface that constructs an effective presentation in every case. Most user interface designers do an ad hoc job and rely on the fact that users are very good at interpretation. The difficulty with this approach is the steady increase in the number of unsophisticated or casual users who cannot interpret poorly presented information. The basic premise of this research is that, during communication, computer systems and users should be equal partners.

There are two major steps in generating a presentation: 1) choosing the presentation techniques or

languages in which to encode the presentation and 2) using them in an effective way so the purpose of the presentation is understood. Since different languages have different uses, the user interface designer chooses the most appropriate language for the information, the communication goals, and the hearer's fluency with the language. Simple natural language systems are commonly used when the domain allows a stylized interaction. Bit raster display technology and powerful graphics packages have made it feasible to use graphical languages¹ like pie charts, bar charts, and diagrams. Speech technology is becoming economically feasible for generation, making limited speech recognition justified for applications where the users cannot use their eyes or hands. Often these media can be mixed. For example, a user interface might display a diagram on the screen and explain it verbally. The proposed research deals mainly with the media of the bit raster display because it supports a number of different graphical languages and because choosing between these languages presents many of the same problems as choosing between languages from different media. Model [16] argues that presentations based on graphical (or "analogical") languages are easy for users to interpret. Graphical techniques are rapidly becoming an important part of most state-of-the-art user interfaces for this very reason. The difficulty is that graphical presentations can be misleading or confusing if they are not carefully constructed to portray the purpose of communication.

The goal of the proposed thesis is to build a knowledge-based user interface called IIPS that constructs effective presentations. Recall from Section 1 that IIPS has three inputs: 1) the information to be presented and a description of its structure, 2) a description of the presentation techniques or languages, and 3) the communication goals to be achieved by the presentation. Two kinds of presentation problems will be addressed by the thesis: 1) *simple presentation*, which is the construction of solitary presentations, and 2) *dynamic presentation*, which is the modification of existing presentations to achieve new communication goals or to present new information. Section 3 gives some hypothetical examples of IIPS performance on simple presentation problems for numerical data. These examples demonstrate the feasibility of building IIPS. The major concern of the proposed thesis will be the simple presentation of symbolic data and dynamic presentation in general.

The development of IIPS addresses some interesting issues in knowledge representation theory about graphical representations. Many graphical representations are not very good for representing general information but are very good for information in a particular form. A map, for example, is very effective for encoding the relative distances between cities (i.e., the ratio of their distances) but the precise distance is difficult to calculate. An important unresolved question in knowledge representation theory is whether these special purpose representations can be described in such a

¹These languages are often called *modalities*. I use the terms interchangeably.

way that a knowledge representation system can choose the appropriate one for a given piece of information. Since IIPS uses graphical presentation languages, the proposed research partially addresses this question. To build IIPS, a precise semantics for the graphical representation must be defined that makes it possible to describe a mapping between the internal representation and the graphical one. A graphical representation can be used when the important information is preserved by the mapping.

IIPS will also be a useful environment for conducting human factor experiments to improve user interfaces. Since IIPS is a knowledge-based expert system, its expertise for presenting information is encoded in a declarative form, allowing easy modification and addition. Two different predicates about a particular presentation technique can be added to two copies of the IIPS knowledge base and a comparative experiment can be run. Since the basic theories of the two presentation systems are the same, any difference measured in the experiment must be due to the predicates. Furthermore, the knowledge base contains the latest theory about information presentation in a form accessible to user interface designers and information presentation systems.

A final motivation for the proposed thesis research is to extend the computational linguistic techniques for carrying on a conversation to the general user interface problem. The use of communication goals in information presentation and the development of a dialogue maintenance tool for user interfaces represent a small step in this direction.

3. Hypothetical Examples of Presentation

3.1. Introduction

This section describes IIPS's performance on some simple presentation problems. The choice of the database domain for this example and some of the heuristics were inspired by a short paper describing Gnanamgari thesis research [8], but the approach taken here, which includes a blackboard architecture, formal knowledge representation, and explicit linguistic knowledge, attempts to be more general and extensible.

IIPS is derived from the blackboard model developed by the Hearsay-II project [6]. Each knowledge source (called a *presentation* KS) represents a particular presentation technique and uses the knowledge on the blackboard to construct a presentation based on that technique. Each presentation KS includes a *descriptive frame* that IIPS uses to decide whether to pursue that particular presentation technique. The basic algorithm has the following three steps:

1. *Select*. Select presentation KSs by testing their descriptive frames against the information's structure and the communication goals. When a presentation KS passes the test, it is *instantiated* by constructing a description of the particular presentation. A presentation KS can be instantiated in more than one way for a particular presentation problem. A description is constructed for each instantiation. I call these instantiations *potential presentations*.
2. *Order*. Order the potential presentations on an agenda by using heuristic information in the descriptive frame of the potential presentation's knowledge source. The goal is to have the most promising potential presentation on the top of the agenda.
3. *Activate*. Activate the knowledge source corresponding to the potential presentation on the top of the agenda, which will then attempt to construct that presentation. During its processing it can run into difficulty, post a description of the problem on the blackboard and *deactivate* itself by moving to some other spot on the agenda. The next most promising potential presentation will then activate.

An agenda control architecture was selected because information presentation requires both data-directed and goal-directed reasoning where the information, as data, determines a particular set of presentation techniques and the communication goals decide among these techniques. An agenda of potential presentations might also be useful during dynamic presentation because IIPS can use the other presentations on the agenda as potential targets for modification.

There are many different types of information that might be presented. The examples in this deals with numerical data but I propose to extend IIPS to symbolic data like organizational information, such as plans and hierarchies, and positional information, such as maps and diagrams. I consider such extensions at the end of this section.

3.2. Domain

The domain for the examples of IIPS's performance is the Stanford Computer Science Department's database. This domain was chosen because it is familiar to people at Stanford, and is similar to many other presentation domains. Many current user interfaces could be rebuilt as knowledge-based expert systems. The database presented in this example departs in a number of ways from the current implementation of the CSD database. Most of the departures simply remove implementation details and idiosyncrasies. Significant departures, pointed out in the examples' annotations, fall into the general class of additional information included to describe the structure of the data. IIPS needs this additional description to decide how to present the information.

Student Relations for 1983			
Name	Degree	Office	
Betty Berkeley	PhD	252	
Charles Cmu	PhD	450	
Cathy Cornell	MSCS	252	
Heather Harvard	PhD	408	
Mike Mit	PhD	450	
Nancy Nyu	MSCS	252	
Rob Rochester	MSEE	252	
Ronda Rutgers	MSEE	311	
Sally Stanford	PhD	408	
Upton Udel	MSCS	352	
Yorich Yale	MSCS	352	
Walt Washington	MSEE	311	

Figure 3-1: Hypothetical Database Relations

Figure 3-1 gives a set of hypothetical relations for the database. Each line gives a student's name, degree program, and office number. A formal semantics for a relational tuple like `Student(Mike Mit, PhD, 450)` is that it describes the results for three functions - `Name`, `Degree` and `Office`. A description of these functions will be on IIPS's blackboard. This additional information is used by the IIPS to search for an effective presentation. For example, one could assert that the `Name` function is a *one-to-one* mapping. Then IIPS could decide that `Mike Mit` should be used as an identifying label in a presentation that describes that student. Subsection 3.3 gives the ontology used for describing these functions. Since IIPS is based on the mathematical notion of functions rather than database relations, the approach described in these examples is relevant to a wider class of domains than relational databases.

A practical question is: what is the source for this descriptive information? In a database domain it is reasonable to have the database administrator provide it, but in other less structured domains this approach is less realistic. One possibility is to derive the structure from the information. This interesting learning research can be pursued once IIPS is constructed.

3.3. Functional Ontology

This section defines an ontology for describing functions that is drawn from standard set theory [11].

Describe a function f with a list:

$$(a_1 \dots a_{n+1})$$

where each a_i represents a set over which the values for the function can vary. The first n of these are commonly called either domain sets or arguments, and the last one is called the range set or result.

The legal mappings for the function f are described with tuples:

$$\langle v_1 \dots v_{n+1} \rangle$$

where $f(v_1 \dots v_n) = v_{n+1}$. Given this notation define the following ontology:

3.4. Communication Goal Ontology

One of the contributions of the proposed research is the explicit incorporation of linguistic notions into the presentation process. This example begins to explore how this approach might work. The communication goals are currently captured in the following set of predicates:

emphasize(x)	x can be a domain value or a function result.
compare(a _i)	Compare the function's results for different values of the argument a _i .
contrast(a _i)	Contrast the function's results.
exactvalue(f)	The exact results of the function are important. This is a special case of the emphasize predicate where each result is to be emphasized.
trend(f)	This is also a special case of emphasize where the first derivative of f is emphasized.

Typically, information in a message is not all at the same level of importance. The major objects under discussion are of primary importance, while "contextual" information, included to help the user with interpretation, is of secondary importance. Sometimes information needs to be emphasized to accomplish some other communication goal. Effective communication must capture these levels of importance in the presentation if the user is going to easily discern the purpose of the message.

Communication goals are a major source of importance information for IIPS. **Emphasize**, **compare** and **contrast** are the fundamental communication goals that affect presentation and are called *presentation goals*. The other goals listed above are special cases of these. Although it wasn't required in these examples, these goals can be used to create a partial ordering for the information to be communicated. IIPS will use this information to make choices. An example is the choice of font size to use for titles, labels, and explanatory text.

Computational linguistics research has identified communication goals other than presentation goals. Robinson [19] identifies "domain goals related to the subject area; knowledge-state goals related to

changing the knowledge of one or more of the dialogue participants; and social goals arising from the social context". The realization of these goals can be a major source for presentation goals. For example, to let the hearer know that an item is important (a knowledge-state goal), the system can emphasize that item in the presentation.

Computational linguistic techniques for producing communication goals are too sophisticated for most user interfaces. Appelt [2] used a planning technology and assertions about the hearer's beliefs to generate utterances. The difficulty is that planning techniques and belief calculi are still in the experimental stage, and systems based on these techniques run too slow to be practical for most user interfaces. I believe that this computational linguistics research points to user interface designs of the future, in which a computer system plans its utterances and uses a belief calculus to model the user so that the utterances accomplish the system's purposes.

Section 4 describes a mechanism adapted from computational linguistics to generate presentation goals that is simpler than planning technology and use modeling. This mechanism was inspired by the observation that user interfaces are involved in conversations with user. Typically, conversations have topics and these topics should be emphasized by the presentation system. If the user interface is involved in a mixed-initiative domain, it will want to manage the dialogue by interrupting the user or changing the topic of conversation. These changes in the dialogue are communicated to the user by giving IIPS presentation goals to emphasize the new topic. This trace does not include the generation of presentation goals since the primary goal is to show that they can be used by IIPS to construct effective presentations.

3.5. Descriptive Frames for Presentation Knowledge Sources

Each presentation knowledge source represents a particular presentation technique or language. Rather than trying to instantiate all potential presentations for a particular problem, IIPS uses a description of the presentation KS to decide whether to pursue that presentation approach. Presentation KSs are described with a frame that can include the following three slots:

1. **Required.** The expression must be true before the presentation KS can be applied to the problem.
2. **Positive.** Each expression (separated by commas) "votes" for this particular presentation. The presentation with the most positive votes is put on top of the agenda.
3. **Negative.** Each expression "votes" against this presentation.

There can be slots other than these three. Predicates in these slots are used to test for particular structure in the information to be presented (*structural predicates*) and the existence of particular communication goals (*communication predicates*). Section 3.3 defined some structural predicates and Section 3.4 defined some communication predicates.

3.6. First Example: A Function with Degree One

Consider the function `NumStudent`, based on the student relation depicted in Figure 3-1, that maps a degree program into the number of students in that program. This function describes the result of a typical database request and is simple enough to give the basic presentation process. The additional information needed by IIPS for this function is described in Figure 3-2.

```
functional(NumStudent)
range(NumStudent) = integer
numerical(integer)
domain(NumStudent) = degree
degree = {PhD MSEE MSCS all}
total(degree)
totalval(degree all)
```

Figure 3-2: Description of `NumStudent`

This information is either present in the database or calculated from the tuples found in the database.

The first step in IIPS's algorithm selects a set of potential presentations by testing the *Required* slot in the descriptive frames of the presentation knowledge sources. The following are a number of different presentation techniques that might be available to the IIPS for describing the `NumStudent` function:

1. Table
2. Pie chart
3. Bar chart
4. Plot chart - commonly called a cartesian graph

Figure 3-3 contains the descriptive frame for the Table knowledge source (Table KS).

Figure 3-3: Descriptive Frame for the Table Knowledge Source

IIPS looks at the *Required* slot to decide whether the technique can be used. Tables can encode any functional information so that is the only requirement. If the information had not been functional, the Table KS would have been skipped. Since `NumStudent` passes this test a potential presentation description is constructed.

Number in Program - 1983	
PhD	5
MSCS	4
MSEE	3
ALL	12

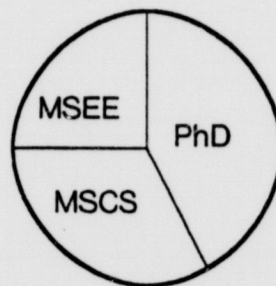
Figure 3-4: Sample Table for `NumStudent`

The second step in the algorithm orders the potential presentations on the agenda by looking at the *Positive* and *Negative* slots of their presentation KSs. Sometimes the structure of the information

can help position an item on the agenda. In this example, tables are moved down in the agenda if there are many values and the chart technique is possible, since plot charts can present many values compactly. However, sometimes the positioning information comes from linguistic knowledge asserted as communication goals. In this case, the Table potential presentation is moved up in the agenda if a current communication goal is to emphasize the exact values of the function. Here we see a mixture of data- and goal-directed reasoning. An actual implementation would probably want to weight the positive and negative votes to emphasize the important votes. In this example, **exact value** should be given a large weight to force the presentation into a tabular form since tables are the best way to present exact values.

Since it satisfied the **Required** slot, the table for the **NumStudent** example is shown in Figure 3-4.

Figure 3-5: Descriptive Frame for Pie Knowledge Source



Number in Program - 1983

Figure 3-6: Sample Pie Chart for **NumStudent**

Figure 3-5 is the descriptive frame for the Pie knowledge source. Since the size of the pie slice denotes a specific numerical result for the function, the second conjunct of the **Required** slot restricts the range to numbers. The third conjunct is the result of the observation that a single pie chart can only describe a function of one argument. Of course, as we will shortly see, a function of degree two can be described with a collection of pie charts. The **Positive** slot defines two independent conditions that move the item up the agenda. The first reflects the fact that pie charts are very good for describing totals. The second condition describes a communication goal, which is the idea of comparing some things. In the first example, we are comparing the number of students in each degree program. The **Negative** slot encodes the heuristic that more than 19 values in a pie chart make it impossible for the user to understand. This heuristic is only hypothetical and the actual value will have to be determined by human factors experimentation. When the correct value is determined, the descriptive frame can be modified rather than changing code inside the Pie

knowledge source. The **NumStudent** function meets the requirements for the Pie KS; the resulting pie chart is shown in Figure 3-6.

Figure 3-7: Descriptive Frame for Bar Knowledge Source

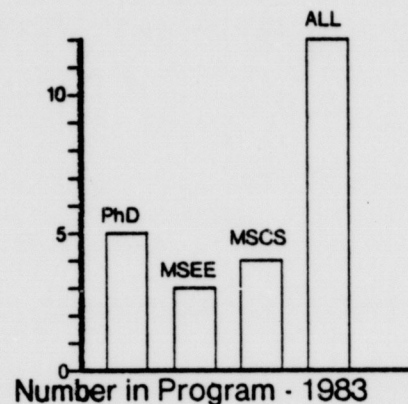


Figure 3-8: Sample Bar Chart for NumStudent

The descriptive frame for the Bar KS is presented in Figure 3-7. Bar charts are very much like pie charts when a function has degree one. The difference is that pies are better for totals. This is reflected by moving the **total** structural predicate to the **Negative** slot. The **ordered** structural predicate in the **Positive** slot reflects the heuristic that the domain axis should be ordered for a bar chart. The bar chart for **NumStudent** is shown in Figure 3-8.

Plot(f)
 Required: $\text{functional}(f) \wedge \text{numerical}(\text{domain}(f)) \wedge$
 $\text{numerical}(\text{range}(f)) \wedge \text{degree}(f) = 1$
 Positive: $\text{continuous}(f), \text{trend}(f)$
 Negative: $\neg \text{continuous}(f)$

Figure 3-9: Descriptive Frame of the Plot Chart Knowledge Source

Figure 3-9 is the descriptive frame for the Plot KS. In pie and bar charts it is possible (and common) for the domain not to be numerical, but in a plot chart the domain must be numerical. This is reflected by the additional condition in the **Required** slot. The **trend** condition embodies the communication goal of emphasizing the trend or first derivative of the function. The Plot chart is not a valid presentation technique for the **NumStudent** example because the domain is not numeric.

Figure 3-10 shows the agenda for the first example when no communication goals are asserted. The **total** structural predicate "cast" the deciding votes.

Potential Presentation	Votes
Pie(NumStudent)	+1
Table(NumStudent	0
Bar(NumStudent)	-1

Figure 3-10: Agenda for NumStudent with no Communication Goals

The next step in the algorithm is to activate the knowledge source for the potential presentation on the top of the agenda. The details of the construction depend on the presentation technique. Although each knowledge source deals with a unique set of problems, there are strong similarities between them since the structure of the information and the communication goals are described in a general language on the blackboard. Furthermore, when a specific knowledge source finds some constraint that makes it undesirable to pursue that particular presentation, it returns control to the `IPS` and describes the constraint so that other knowledge sources can be aware of it. For example, if one result of a function is very small compared to the rest, the pie slice on a pie chart might be invisible. At this point the Pie KS would stop and assert this fact on the blackboard. The Bar KS can use this fact to deactivate itself and a plot chart will be used which is very good at showing small results.

To show the basic operation of a layout knowledge source let's return to the `NumStudent` example and see how the pie chart is constructed. The Pie KS knows that its input is a function from a single domain to a numerical range. It also knows about the idea of "totals". If it is asserted that the function is total, the pie KS uses the total value as its base of calculation (in this case the result of `NumStudent(a11)`, or 12). It knows that it has to distribute the 360° of a circle between the other values. For example, it does the simple ratio calculation to determine that PhD gets a 150° pie slice. If the function is not total then the Pie KS sums the results and uses that for its ratio calculation. Next the Pie KS tries to identify the individual pie slices. In this case there is a one-to-one mapping from the values in the domain of `NumStudent` to names for those values so the `IPS` can decide to use those names as labels. Next the `IPS` must decide on the placement of the labels. In this case it put them in the pie slices, but if there hadn't been enough space a texture could have been used. This is an example of a presentation subproblem that the knowledge source could have recursively called `IPS` to solve.

3.7. Second Example: Segmenting

The previous example demonstrated how `IPS` handles functions of degree one. This example demonstrates a process called segmenting that `IPS` uses to handle functions with degrees larger than one. The basic idea behind segmenting is to restrict a function along one of its domains and produce a set of functions that collectively represent the function and individually have one less domain. All

the functions in this restriction set are constrained to be presented with the same presentation technique. In effect, segmenting produces a set of easier presentation problems. For each presentation technique that allows segmenting, there is a segmentation knowledge source that monitors the presentations of the restriction functions to make sure that they look alike. When a particular function is segmented, a potential presentation is put on the agenda that describes the restriction functions and the constraints that their presentations must satisfy. When the segmentation knowledge source is activated, it recursively calls `IPS` to construct a presentation for each of the restriction functions that satisfies these constraints.

Figure 3-11: Hypothetical Data for the Number of Students in Degree Programs

```
functional(numbers)
range(number)=integer
numerical(integer)
domain(number)={degree year}
degree={PhD MSEE MSCS all}
total(degree)
totalval(degree all)
year=integer
```

Figure 3-12: Description of NumYear

To demonstrate segmentation, consider the function `NumYear` which is just like the `NumStudent` function of the previous section but includes an argument for the year. Figure 3-11 describes some hypothetical data for the number of students in each program for the last three years, and Figure 3-12 describes the `NumYear` function.

Figure 3-13: Descriptive Frame for the Pie-Segment Knowledge Source

The descriptive frame for the Pie-segment KS is shown in Figure 3-13. Segment KSs for functional data restrict the function along some domain to produce functions that are easier to present. The function `segment(f)` refers to the domain that is to be restricted. The `Segment` slot in Figure 3-13 tests this domain to make sure the the total number of pie charts will not be greater than 10. Although this is a hypothetical heuristic, it embodies the observation that too many pie charts are difficult for users to interpret. The `NumYear` function has two domains that both pass the segmentation test so two potential presentations are added to the agenda. When a domain passes the segmentation test, a set of restriction functions are calculated and the presentation KS for each function is determined by the Segment KS. For pie segmentation, the Pie KS is used for all the restriction functions.

For the `NumYear` example there will be two potential presentations for the Pie-segment KS added to the agenda - one restricted on the domain `degree` and one on the domain `year`. The degree agenda

item will have four restriction functions that can be called **PhD**, **MSCS**, **MSEE** and **all** that map a year into the number of students in a specific degree program. The year agenda item has three restriction functions - one for each of the years **1983**, **1982**, and **1981**. Each of these functions looks just like the **NumStudent** function.

The second step determines the placement of the potential presentations on the agenda. In the case of **Segment Ks**, their placement depends on their restriction functions. Each restriction function is evaluated with the **Positive** and **Negative** slots in the descriptive frame of their presentation **Ks**. The three pie segmentation restriction functions for the **year** domain all get a positive vote because they are total functions. The degree program functions don't get any votes unless the **compare** communication predicate for the **degree** domain is true. Thus the **year** segmentation will be higher on the agenda than the **degree** segmentation.

Figure 3-14: Sample Pie Charts for NumYear

Figure 3-14 shows a pie presentation for the **NumYear** function segmented by the **year** domain. This figure demonstrates an important point: presentations can be misleading if they are not carefully constructed. In the 1983 pie the **MSEE** pie slice appears above the **MSCS** slice, while in the 1982 slice it is the other way around. The casual person might not notice this switch and think that the very small slice for 1982 corresponds to the much larger slice for 1983 at the same position. To avoid this kind of problem there will need to be more constraints between the presentation of restriction functions than just their presentation **Ks**.

Figure 3-15: Descriptive Frame for the Bar-segment Knowledge Source

Bar graphs are better for functions of degree two than pie charts because all the data can be represented on one chart. Figure 3-15 is the descriptive frame of the Bar-segment knowledge source. The second conjunct in the **Segment** slot makes sure that there will be a bar for every value so that the bars can be compared. (I have seen bar charts where a few of the bars were missing and found them quite readable. If this is desired then one could write an expression that describes the exact variance that should be allowed.) Figure 3-16 shows the bar chart that results from this segmentation for domain **degree** segmentation.

Figure 3-16: Sample Bar Chart for NumYear

There are four sets of bar graphs that represent the four functions - one for each degree program and one for the total. Each function maps years into numbers of students. The reason that the **degree** segmentation was chosen rather than the **year** segmentation is that the the **Negative** slot for the

bar chart description mentions the total predicate. In the year segmentation the restriction functions are all total functions but in the degree segmentation they are not. The ordering of the two bar segmentations is the inverse of the pie chart's, which shows the flexibility of using declarative techniques like descriptive frames.

The third step of the algorithm is to activate a knowledge source. Segment KSs are more complicated than the basic KSs of the previous example because they must keep track of the presentation of each of the restriction functions. Take for example the bar graph for the NumYear function shown in Figure 3-16. The bar that denotes a particular year in each of the collection of bars must have the same distinguishing texture. This is accomplished by posting these constraints on the blackboard.

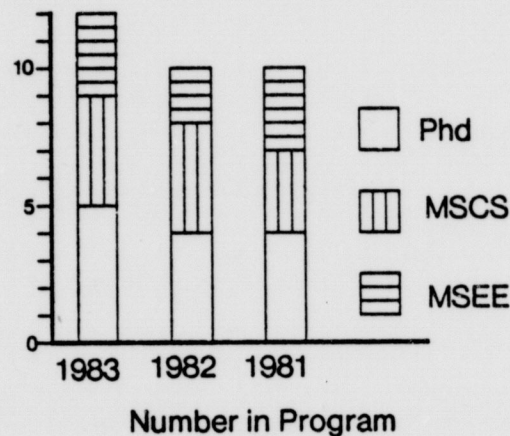


Figure 3-17: Sample Layered Bar Chart for NumYear

Actually, bar charts can effectively present totals if they are used with a special technique called *layering*. Figure 3-17 shows the bar chart for the NumYear function with the use of layering. The Bar KS could decide to use the layering technique when the communication goal `emphasize(a11)` is asserted since `a11` is the total of the other arguments. This is an example of a knowledge source using communication goals while it is doing layout.

```
Plot-segment(f)
  Required: functional(f) ∧ numerical(range(f)) ∧
           v ∈ domain(f) ∧ v ≠ segment(f) ∧
           numerical(v) ∧ degree(f) = 2
  Segment: card(projection(segment(f))) ≤ 6
```

Figure 3-18: Descriptive Frame for the Plot-segment Knowledge Source

Figure 3-18 shows the descriptive frame for the Plot-segment KS. The expression in the Segment slot is not as complicated as in the Bar-segment KS because the values on a plot chart do not have to line

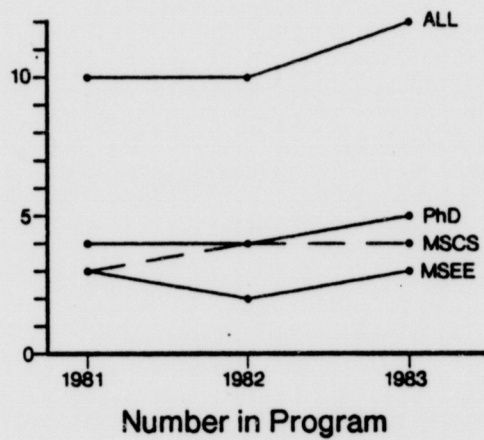


Figure 3-19: Sample Plot Chart for NumYear

up to compare the plotted lines. The `Required` slot includes an additional set of expressions that make sure that the domains of the restriction functions are numerical. Since this is true for function `NumYear` when the restriction domain is `degree`, the knowledge source for the chart in Figure 3-19 is put on the agenda. The `year` segmentation is not instantiated because the range is not numerical.

As Figure 3-20 shows, plot charts can also be layered.

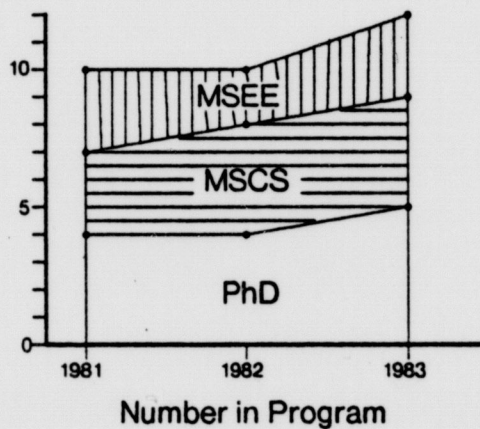


Figure 3-20: Sample Layered Plot Chart for NumYear

There are many different types of plot charts. The plot chart in Figure 3-19 has a linear scale and is called a linear plot chart. An important class of charts are logarithmic plot charts where at least one of the scales is logarithmic. There are also polar coordinate plot charts, triangular plot charts,

perspective plot charts and probability plot charts. Each one of these would be represented with a presentation KS.

3.8. Extension to Relations and Their Graphs

IIPS as it is described so far deals with numeric data and makes extensive use of the set theoretic notion of a function, but in my thesis I plan to deal extensively with the presentation of symbolic, qualitative information. This subsection deals with symbolic information that is based on relations. Relations are used to describe structure and organization. An example from the departmental database domain is the *prerequisite* relation *Pre* for classes. Consider the hypothetical data in Figure 3-21.

```

CS107 = Structured Programming
CS108A = Introduction to Computer Science Part 1
CS108B = Introduction to Computer Science Part 2
CS142 = Programming Languages
CS143 = Compilers
CS161 = Data Structures
CS242 = Programming Language Design
CS243 = Advanced Compilers

Pre(CS108B,CS108A)
Pre(CS142,CS107)
Pre(CS142,CS108B)
Pre(CS161,CS107)
Pre(CS161,CS108B)
Pre(CS143,CS142)
Pre(CS242,CS142)
Pre(CS243,CS143)

```

Figure 3-21: Hypothetical Data for Prerequisite Relation

3.8.1. Relational Ontology

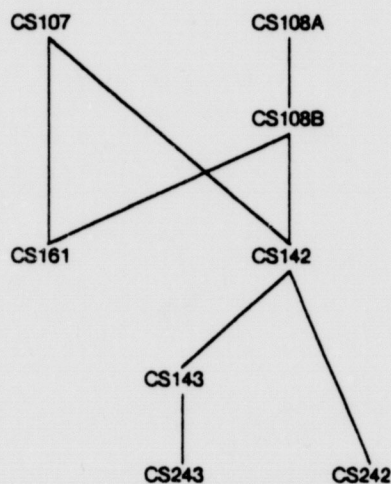
Functions and relations are very similar and since relations also have domains most of the functional ontology is still useful. The list of additional predicates in Figure 3-22 can be used to describe relations. The *Pre* relation is a partial order.

3.8.2. Graph Example

A standard presentation technique for relations is to represent the two values of the relation as *nodes* and represent the relation by drawing an *arc* between them. A collection of nodes and arcs forms a *graph*. Figure 3-23 shows the graph for the *Pre* relation. Asymmetric relations are drawn with a *directed arc*. In the figure, the direction is indicated by the downward slope of the arcs, but arrowheads are often used. Graphs are used extensively in computer science and there are many special kinds of graphs. There are trees, binary trees, directed acyclic graphs, cyclic graphs, and forests. Each of these has a set of presentation conventions and a set of presentation problems. This suggests that they should each have their own presentation knowledge source.

reflexive(r)	$\forall x r(x,x)$
symmetric(r)	$\forall x,y r(x,y) \Rightarrow r(y,x)$
transitive(r)	$\forall x,y,z r(x,y) \wedge r(y,z) \Rightarrow r(x,z)$
irreflexive(r)	$\forall x \neg r(x,x)$
asymmetric(r)	$\forall x,y r(x,y) \Rightarrow \neg r(y,x)$
antisymmetric(r)	$\forall x,y r(x,y) \wedge r(y,x) \Rightarrow x = y$
partial order	(transitive \wedge asymmetric \wedge irreflexive) or (transitive \wedge antisymmetric \wedge reflexive) Less-than "<" is the most common relation that satisfies the first set of requirements and less-than-or-equal " \leq " the second.
equivalence	transitive \wedge symmetric \wedge reflexive Equality "=" is the most common relation that satisfies these requirements.

Figure 3-22: Relational Ontology



Prerequisites Computer Science Classes

Figure 3-23: Example Graph

The placement of an individual arc is fairly easy but when there are many nodes and arcs it can become quite complicated. Part of the difficulty is that there is a lot of freedom for where to put each node and how to draw the arcs. This freedom makes it easy to construct a poor presentation. Good graph presentations encode a lot of information into the placement of nodes. The graph of Prerequisites in Figure 3-23 is very carefully drawn to reflect the additional domain-specific relations shown in Figure 3-24. The distance from CS108A to CS108B is shorter than the distance from CS108B to CS142, reflecting the fact that the first two are in a sequence. This also holds true for CS143 and CS243. CS243 and CS242 are at the same level in the graph because they are both graduate level courses. For IIPS to be able to construct a presentation like this, it will have to know that these additional

```

sequence(CS108A,CS108B)
sequence(CS143,CS243)
graduate(CS243)
graduate(CS242)

```

Figure 3-24: Additional Data about Class Prerequisites

relations are less important than the relations shown in Figure 3-21. Arcs are only used to present important relations while positional techniques can be used to present less important relations. The complexity of this example suggests that the presentation knowledge sources for symbolic information will probably use heuristic search techniques and their own level of the blackboard.

3.8.3. Dynamic Presentation

Graph presentation also raises the important issue of *dynamic presentation*. Consider the *Pre* presentation in Figure 3-23 when the user wants to add the relation *Pre*(CS242,CS161). IIPS is confronted with a new type of problem: it already has a presentation that must be modified to include the new relation. Simply drawing a line from CS161 to CS242 is unsatisfactory because it would come very close to the node for CS143, making the graph difficult to read. A better idea might be to switch the branch for CS242 with the branch for CS243, placing them in correct numerical order. This placement also emphasizes the dual prerequisites for CS242 by making their arcs the same length.

IIPS is confronted with dynamic presentation problems when the user interface is involved in a dialogue with the user. In a dialogue the topic of conversation changes over time and the old objects of interest need to be de-emphasized and finally fade away. The user interface identifies this by tracking the focus of the conversation and noticing when it shifts to a new object. The IIPS constructs a presentation that emphasizes this new object and compares it with the current presentation. If they are similar, a transition is made between them. If they are far apart, the agenda is checked for other presentations that might be closer to the current presentation. I propose to pursue dynamic presentation as part of my thesis research.

Dynamic information presentation might be particularly relevant for the presentation of simulation systems, such as circuit simulators, and for the presentation of monitoring information, such as IC fabrication monitoring systems.

3.9. Extension to Maps and Diagrams

After charts and graphs, maps and diagrams are the most common graphical presentation technique. These techniques are used to show the location of objects and their shape. The most convenient way to describe them is as composites. A map, for example, has a plot chart for the coast line, a graph with cities as nodes and roads as arcs, and a table that describes what the symbols mean. The map

knowledge source will ask the IIPS to solve a number of significant presentation problems. Once IIPS can present relational information fairly well, I plan to work on maps and diagrams.

4. Discourse Focus as a Source for Communication Goals

User interfaces are involved in a *dialogue* with the user. Many user interfaces have relied heavily on the fact the people are very good at carrying on a conversation. These systems have been difficult to deal with because they do not adhere to the usual linguistic conventions by which people manage conversations. Early user interfaces often failed by being too obscure or too inflexible. Obscurity is symbolized by the numeric error codes of FORTRAN compilers. Inflexibility, of course, makes the user interface's job easier by restricting the conversation to one topic. More recent user interfaces have addressed themselves to these problems. Window systems allow users to carry on multiple conversations at once. A recent command language system by Hayes [13] allows the user to switch topics at any time and return to the exact saved context at some later time. User interface designers also create verbose user interfaces to provide enough context for the interpretation of messages. Unfortunately, this solution creates additional problems. Users have difficulty interpreting verbose output unless the presentation is carefully structured. Furthermore, a user returning to a conversation must remember what the topic was. If the information is poorly presented, the user will have to work too hard. This section describes a proposed tool called a *Dialogue Management System* (DMS) that a user interface can use to keep track of conversations with the user so that information is effectively presented. This system will be a major source of communication goals for IIPS.

The DMS design adapts techniques developed in computational linguistics for keeping track of the *focus* of a conversation to the general user interface problem. Grosz [10] used a *global focus* mechanism to interpret task oriented dialogues. Global focus keeps track of the major shifts in topic of a conversation which, in a task oriented dialogue, follows the plan for accomplishing the task. Sidner [20] used a *local focus* mechanism to resolve definite noun phrases during interpretation. Local focus keeps track of the objects mentioned in sentences and how they change over time. McKeown [15] recently used these focus mechanisms for the generation of English descriptions of database structure, but her global focus, unlike the work on interpretation, does not change over time. The DMS also uses both focus mechanisms for generation and the global focus changes over time.

The DMS keeps track of dialogues by allowing the user interface to *register*, *suspend*, and *resume* dialogues. The *active* dialogue is the current one. The user interface registers a dialogue when it realizes that some new topic is being talked about. The previously active dialogue is pushed onto a stack of suspended dialogues and the newly registered dialogue is made active. Dialogues on the stack can be resumed. Activation and resumption of dialogues represent shifts in the global focus. IIPS can look at the representation in the DMS to decide how to present the shift effectively. Each

dialogue represented in the DMS includes information about the major objects and relations of that conversation that IIPS will emphasize. This information represents the local focus and is updated as the dialogue progresses. The user interface can also *close* a dialogue to indicate that the conversation is ended. A closed dialogue is not removed from the DMS since it can be used by the user interface for constructing error messages or for indicating that the particular task described by the dialogue is finished executing.

A major area for further research is how to make IIPS indicate to the user that the topic has changed. Reichman [17] studied discourse samples and identified certain linguistic techniques that were used to indicate the topic shift. Her samples were in natural language and she identified certain "clue words" that indicated these shifts. For graphical languages, there are also techniques that have been used in user interfaces: the screen is flashed, the new item is highlighted, or the old window is "greyed out" as it gets out of date. Each of these techniques can be described and IIPS can incorporate them into a presentation when the DMS indicates that a topic has changed.

5. Related Work

This section discusses the relationship between the proposed research on intelligent information presentation and to other research.

5.1. Information Presentation

I first ran across the phrase "information presentation" in the BBN work by Zdybel, Greenfeld, Yonk and Gibbons [24]. They developed KL-ONE descriptions of presentation techniques (such as maps) that could be connected to other KL-ONE structures. Their work represents the first step in knowledge-based information presentation and I plan to extend their work in many ways. IIPS will cover a larger class of presentation techniques, use a heuristically guided search for presentations, and use communication goals in an effort to construct effective presentations.

The short paper describing Gnanamgari's thesis [8] inspired the domain for the examples in this proposal. She did a simple version of information presentation by using defaults to fill in presentation specifications left out by the user. I deal with functional information in a more systematic way and include communication goals to help construct a presentation. Furthermore, I plan to extend IIPS to non-numerical domains containing qualitative information like relations and position.

Model's thesis work [16] on a monitoring system for MYCIN does information presentation with some graphical techniques but it isn't a knowledge-based like IIPS. However, his thesis is also relevant because it argues for the use of graphical presentation techniques on cognitive grounds. He concludes that graphical presentations are easy for people to interpret and can partially alleviate short-term memory problems.

The STEAMER work at BBN [7] is good example of the power of graphical techniques but their approach to presentation is also not knowledge-based.

There are many other user interfaces, graphic packages and expert system explanation facilities that relate to IIPS. As far as I know, none of them use a knowledge based approach to information presentation.

5.2. Synthesis Domains

Information presentation is a synthesis problem rather than an interpretation or diagnosis problem. Some examples in artificial intelligence of synthesis research are: automatic programming research (such as Green's PSI project [9]), planning research (such as Stefik's MOLGEN system [22]), and expert system research (such as McDermott's R1 system [14]). Most of this work is in a temporal domain, but

R1 and IIPS deal mainly with spatial problems. R1 configured computer systems and it had to worry about the relative locations of the various components - some couldn't be too close for air conditioning reasons. In IIPS there are also constraints between spatial objects, but information presentation has spatial problems that are very different from those solved by R1. As an example, consider the layered bar chart presentation for NumYear in Figure 3-17, where the restriction functions are shown in a single presentation rather than in individual juxtaposed presentations. R1 uses match with almost no backup. Information presentation problems may or may not require backup. It will be interesting to see how the differences in the two domains affect the search.

5.3. Knowledge Representation

Information presentation can also be viewed as a knowledge presentation problem: the translation of information from an internal representation to an external one that is understood by the user. This translation need not preserve information since only some of the information might be relevant to the user. To be able to do this, IIPS must have a description of the translation process. A meta-level description of the internal and external representations is an effective way of describing the translation. For example, a mapping can be described from a description of the internal representation of a function (e.g. a database relation) to a description of the external representation (e.g. the location in a table). The translation of some information can be evaluated by looking at the mappings and determining if the important information is preserved in the external representation.

There has been confusion about graphical representations. Sloman [21] argued that there were certain problem solving domains where "analogical" representation languages should be used rather than "Fregean" languages like predicate calculus. By analogical representation he meant representations like pictures, maps, and scale models². He went on to contrast these representations with formal predicate calculus representations and argue that they were quite different. Hayes [12] responded to this claim:

"Thus Sloman's emphasis on what he calls analogical representations is really a plea for the consideration of a wider class of languages than those in which the only semantic primitive is the application of a function to arguments"

Hayes sketched an approach for giving analogical representations a precise semantics. I intend to expand on this sketch in the thesis. It remains to precisely characterize those cases when an analogical representation is appropriate. Hayes continues:

"This shows, incidentally, that Sloman's arguments for the utility of analogical representations, based on the idea that they are somehow more efficient in use than Fregean representations, are fallacious. For an analogical representation may be

²For further information about analogical representations see Chapter II Section C5 of the AI handbook (page 200) [4]

embedded in a medium which is itself represented in a Fregean way in some other medium. Any discussion of efficiency must take into account the computational properties of the medium."

Thus in information presentation, the appropriate choice of a graphical language depends on the computational abilities of the user. The proposed thesis will show that graphical representations are precise, that information in other representations can be translated into graphical ones, and that particular representations can be chosen because they are useful. This suggests that knowledge representation systems could be designed that could choose special purpose representations for information when the computational uses for the information are precisely described.

5.4. Human Factors and User Interface Design

Given some information to present, IIPS must choose an appropriate language for that information and then construct an effective presentation. In the last subsection this basic process was characterized as a translation from one representation to another. The difficulty with this characterization is that the choice of language and the construction of a presentation depend on *who* is to interpret the presentation. For example, a particular presentation technique might be appropriate for a hearer that is trained to hear it. For an information presentation system that communicates with other computer systems it might be reasonable to have an exact description of the hearer of the message, but it is not yet possible to have a precise description of people. Cognitive science and human factors researchers are working on this description and their work is ultimately relevant to this thesis. The major requirement of the proposed thesis is that IIPS must be flexible enough to use the results of this research. The knowledge-based approach will probably satisfy this requirement. As additional facts are determined and a user model is built, the descriptive frames for the presentation knowledge source can be modified to refer to this additional information. The expressions in the descriptive frames will be a collection of guesses about the heuristics for presentation derived from graphic art and experimentation. The proposed research will not make any claims about having the correct expressions. The goal is to show the feasibility of incorporating the correct expressions when they have been developed by incorporating a plausible set of heuristics in IIPS.

On the other hand, the proposed thesis will be quite relevant to human factors research because IIPS will be a good environment for developing human factors experiments. One of the problems for human factors researchers is interpreting experiments that compare two different systems. Robert's thesis [18] compared a number of text editors using human factors techniques but the editors were so different that it was difficult to draw any general conclusions about text editor design. IIPS can be used to develop human factors experiments by comparing two version with a slightly different

knowledge base. I do not propose to run such an experiment as part of my thesis, but I do plan to demonstrate the feasibility of the approach during the validation phase of the project. The validation of the IIPS will be a comparison of the IIPS to a version of itself with some of the linguistic knowledge removed. This ablation study will demonstrate the importance of including a notion of purpose in the design of user interfaces.

5.5. Computational Linguistics

Since this proposed research deals with human/computer communication and emphasizes the importance of including communication goals in the process of constructing a presentation, it is clearly related with computational linguistics research. References have been included throughout this proposal. The speech act planning research of Cohen [5], Allen [1], and Appelt [2] is important because conversations between people and computers are task oriented. Discourse research such as Grosz's global focus model [10] is a good source for communication goals. Other computational linguistics research relates to user interface design, but most of it deals with command language design and thus is not particularly relevant to the proposed thesis.

6. Conclusion and Schedule

This thesis proposal describes a knowledge-based user interface called an Intelligent Information Presentation System (IIPS) which uses communication goals in a heuristically-guided search for an effective presentation. Several examples of IIPS performance demonstrate the feasibility of the proposed thesis. Section 2 on motivation and Section 5 on related work argue for the importance of this research in relationship to research in knowledge representation, user interface design, and computational linguistics.

The steps in completing the research are:

1. *Implement IIPS for the examples in this proposal.* Although numerical data is not the major concern of this thesis, it does offer a valuable "first case" for a knowledge-based information presentation system because the presentation techniques are very stylized. This implementation does not include any generation of communication goals.
2. *Extend IIPS to qualitative and non-numerical presentation problems.* Organizational trees, procedural networks for plans and other graphical techniques are the target languages for this problem. This extension does not include dynamic presentation.
3. *Extend IIPS to dynamic presentation.* Use the simple presentations that IIPS already produces as a starting point and develop techniques to modify them to include new information or delete old. Generation of communication goals is still not included.
4. *Add the Dialogue Management System (DMS) to the implementation.* The user interface will now be able to carry on a conversation with the user and construct presentations that convey information and manage the conversation in an effective manner.
5. *Perform an evaluation study of IIPS.* An ablation experiment will be run that compares the prototype to a version of itself with some of the linguistic knowledge removed. The goal is to demonstrate that linguistic knowledge is important to the effective presentation of information.
6. *Write a dissertation describing this research.*

References

- [1] Allen, J. F.
A Plan-Based Approach to Speech Act Recognition.
PhD thesis, Computer Science Department, University of Toronto, 1979.
- [2] Appelt, Douglas E.
Planning Natural-Language Utterances to Satisfy Multiple Goals.
PhD thesis, Computer Science Department, Stanford University, 1982.
- [3] Austin, J. L.
How to Do Things with Words.
Oxford University Press, 1962, pages ???.
- [4] Barr, A., Feigenbaum, E.
The Handbook of Artificial Intelligence - Volume I.
William Kaufmann, Inc., 1981.
- [5] Cohen, Philip R.
On Knowing What to Say: Planning Speech Acts.
PhD thesis, Computer Science Department, University of Toronto, 1978.
- [6] Erman, L., Hayes-Roth, F., Lesser, V., and Reddy, D. R.
The Hearsay-II Speech-Understanding System: Integrating Knowledge to Resolve Uncertainty.
A.C.M. Computing Surveys 12(2):213-254, 1980.
- [7] Forbus, K., Stevens, A.
Using Qualitative Simulation to Generate Explanations.
Technical Report 4490, BBN, 19??
- [8] Gnanamgari, S., Badler, N., Morgan, H., Webber, B.
Providing Automatic Graphic Displays Through Defaults.
In Proceeds of the Third Biennial Conference, pages 312-319. CSCSI, 1980.
- [9] Green, C., Gabriel, R., Kant, E., Kedzierski, B., McCune, B., Phillips, J., Tappel, S., and Westfold, S.
Results in Knowledge Based Program Synthesis.
In IJCAI 79 - Tokyo, pages 342-344. IJCAI, 1979.
- [10] Grosz, Barbara J.
Focusing and Description in Natural Language Dialogues.
Joshi et al (eds.), 19??, pages 84-105.
- [11] Halmos, P. R.
Naive Set Theory.
Litton Educational Publishing Inc., 1970.
- [12] Hayes, Patrick J.
Some Problems and Non-Problems in Representation Theory.
In Summer Conference, pages 63-79. AISB, 1974.
- [13] Hayes, Philip.
Cooperative Command Interaction through the COUSIN System.
In International Conference on Man/Machine Systems. 1982.

- [14] McDermott, J.
R1: A Rule-Based Configurer of Computer Systems.
Artificial Intelligence - An International Journal 19(1):39-88, 1982.
- [15] McKeown, Kathleen R.
Generating Natural Language Text in Response to Questions About Database Structure.
PhD thesis, Moore School of Electrical Engineering, University of Pennsylvania, 1982.
- [16] Model, Mitch L.
Monitoring System Behavior in a Complex Computational Environment.
PhD thesis, Computer Science Department, Stanford University, 1979.
- [17] Reichman, Rachel.
Conversational Coherency.
Cognitive Science 2:283-327, 1978.
- [18] Roberts, Terry L.
Evaluation of Computer Text Editors.
PhD thesis, Computer Science Department, Stanford University, 1979.
- [19] Robinson, Ann.
Determining Verb Phrase Referents in Dialogs.
American Journal of Computational Linguistics 17(1):1-16, Jan-Mar, 1981.
- [20] Sidner, Candace L.
Towards a Computational Theory of Definite Anaphora Comprehension in English Discourse.
PhD thesis, M.I.T., 1979.
- [21] Sloman, Aaron.
Interactions Between Philosophy and Artificial Intelligence: The Role of Intuition and Non-Logical Reasoning in Intelligence.
Artificial Intelligence - An International Journal 2:209-225, 1971.
- [22] Stefik, M.
Planning with Constraints.
PhD thesis, Computer Science Department, Stanford University, 1980.
- [23] Turing, A. M.
Computing machinery and the mind.
Mind: A Quarterly Review of Psychology and Philosophy 59:433-460, 1950.
- [24] Zdybel, F., Greenfeld, N., Yonke M. and Gibbons, J.
An Information Presentation System.
in *IJCAI 81 - Vancouver*, pages 978-984. IJCAI, 1981.

FILMED FROM BEST AVAILABLE COPY

**Copyright © 1985 by KSL and
Comtex Scientific Corporation**