

USER MODELS FOR INTENT-BASED AUTHORIZING

By

Andrew Csinger

B.Eng. McGill University
M.Sc. (Computer Science) University of British Columbia

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

in

THE FACULTY OF GRADUATE STUDIES
DEPARTMENT OF COMPUTER SCIENCE

We accept this thesis as conforming
to the required standard

.....
.....
.....
.....
.....

THE UNIVERSITY OF BRITISH COLUMBIA

November 1995

© Andrew Csinger, 1995

In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of Computer Science
The University of British Columbia
Vancouver, Canada

Date:

Abstract

Authoring is the collection, selection, preparation and presentation of information to one or more readers by an author. The thesis takes a new, critical look at traditional approaches to authoring, by asking what knowledge is required and at which stages of the process. From this perspective, traditional authoring is seen to entrench an early commitment to both form and content.

Although the late binding of form is now commonplace in structured document preparation systems, a similar delay in the binding of content is necessary to achieve user-tailored interaction. The authoring paradigm we have developed to service this goal is called *intent-based authoring*, because the author supplies at compile-time a communicative goal, or *intent*. Just as SGML editors and HTML browsers defer rendering decisions until run-time by referring to a local style-sheet, intent-based authoring systems defer content-selection decisions until run-time when they refer to models of both author and reader(s).

This thesis shows that techniques from artificial intelligence can be developed and used to acquire, represent and exploit such models. Probabilistic abduction is used to recognize user models, and cost-based abduction to design tailored presentations. These techniques are combined in a single framework for best-first recognition and design.

These reasoning techniques are further allied with an interaction paradigm we call *scrutability*, whereby users critique the model in pursuit of better presentations; users see a critical subset of the model determined by sensitivity analysis and can change values through a graphical user interface. The interactivity is modelled to ensure that representations of the user model to the user are made in the most *perceptually salient* manner.

A prototype for intent-based video authoring is described. Video is used as a test medium because it is a “worst case” temporally linear medium; a viable solution to video authoring problems should apply easily to more tractable traditional media.

The primary contribution of this dissertation is to the field of applied artificial intelligence, specifically to the emerging field of user modelling. The central contribution is the intent-based authoring framework for separating intent from content.

Contents

Abstract	i
Table of Contents	ii
List of Tables	v
List of Figures	vi
Acknowledgements	vii
I Introduction	1
1 Systems have it Easy	2
1.1 Introduction	4
1.2 Traditional Approaches	5
1.3 Intent-based Authoring	8
1.4 <i>Valhalla</i> and the Departmental Hyperbrochure	11
1.4.1 Sample Session	11
1.5 Overview	13
II Literature Survey and Formal Background	15
2 User Modelling	17
2.1 Overview	18
2.2 Issues: Acquisition and Exploitation.	23
2.2.1 Acquisition	23

2.2.2	Exploitation	24
2.2.3	Correction: Situations requiring correction.	26
2.2.4	Scope: What is Represented?	27
2.2.5	Extent and Adaptivity	28
2.3	Multiple Agents, Multiple Models	29
2.4	User Modelling Shells	29
2.5	User Modelling: A Definition by Consensus	31
3	Authoring	35
3.1	Dimensions of Authoring	36
3.2	Graph Generation	41
3.3	Data Analysis and Visualization	46
3.4	Psychophysical Research	50
3.5	Multimedia	52
3.5.1	Video Authoring	58
3.6	Computer-Supported Cooperative Work	63
3.7	Structured Documents: Content from Form	69
3.8	Hypertext and Hypermedia	72
3.9	Putting it all Together: Cyberspace?	75
4	Formal Background	80
4.1	Symbolic Logic and Default Reasoning	80
4.1.1	Default–Programming with Theorist	82
4.1.2	Summary and Conclusions	85
4.2	Decision Making Under Uncertainty	86
4.2.1	Bayesian Decision Theory	86
4.2.2	Example	87
4.2.3	Decision Analysis	89
4.3	Speech Acts	89
III	Contribution	92
5	User Models for Intent-based Authoring	96
5.1	Overview	96
5.2	Motivation	97
5.3	Components of the Theory	98

5.3.1	Summary: Inputs, Outputs and Roles	105
5.4	An Abductive Framework for Recognition and Design	106
5.4.1	Recognizing User Models	107
5.4.2	Designing Presentations	119
5.5	Interactivity and Scrutable Models	128
5.6	Example	132
5.7	Alternative Approaches	137
5.7.1	Decision Theory for Multimedia Presentation Design . . .	137
5.7.2	Costs and Utilities	139
5.7.3	Other similar approaches	141
5.8	Conclusions	142
6	Implementation	144
6.1	Architecture	145
6.1.1	The Reasoner	152
6.2	Functionality	154
IV	Conclusions	159
7	Conclusions	160
7.1	Implications	162
7.2	Future Work	162
7.2.1	Learning: Updating Prior Probabilities	163
7.2.2	Privacy	167
7.2.3	Future Development	169
	Bibliography	173

List of Tables

2.1	Errors in user models.	26
3.1	Mackinlay and related work.	44
3.2	Data Analysis and Visualization.	49
3.3	Intermedia.	57
3.4	CSCW time and space diagram.	64
3.5	KMS.	66
4.1	Domain–Formulation	84
5.1	Priors and normalized posteriors.	119
5.2	Myopic Tradeoff Table	122
5.3	Results of Sensitivity Analysis before user action	135
5.4	Results of Sensitivity Analysis after user action	136

List of Figures

1.1	The traditional approach to authoring	4
1.2	The intent-based approach to authoring	8
1.3	The <i>Valhalla</i> User Model Window	12
1.4	The <i>Valhalla</i> Control Window	13
2.1	User and system models	25
3.1	EXVIS five-dimensional stick figures	47
3.2	Iconograph	48
5.1	A partially elaborated presentation	104
5.2	Roles, Inputs and Outputs	105
5.3	Representative recognition assumables	107
5.4	Example facts and rules	108
5.5	Rules for perceptual salience	112
5.6	The <i>Valhalla</i> User Model Window	113
5.7	Assumables for perceptual salience calculations	114
5.8	Show: an example intent	127
6.1	Valhalla's Distributed Architecture	145
6.2	The hierarchy of interfaces to the video server	147
6.3	The <i>Valhalla</i> Control Window	149
6.4	A frame from the Departmental Hyperbrochure	151
6.5	The <i>Valhalla</i> User Model Window	152
6.6	Valhalla implementation	153
6.7	The <i>Valhalla</i> Agents	153
6.8	Authorial Intent as a Prolog query	154
6.9	Knowledge-based Video Annotation and Presentation	155

Acknowledgements

*And further, by these, my son, be admonished:
of the making of many books there is no end;
and much study is a weariness of the flesh.*

—Ecclesiastes 12:12

Sue Rathie, who kept me on the path where I would have wandered, oh, so many times. For reading my thesis more often than anyone else. For helping me with all those details. For listening. For being there, *all the time*...

My parents, Helen and Joseph, who got me to where I could do the rest on my own. There *were* times over the last few years when I think the only reason I kept going was because I promised my mother I would “get a PhD.” I was six years old when I made that promise...

Kelly Booth has been the source of much good advice, and a partner in many a stimulating discussion. When I went to Germany for a summer of study, he left me with the sage words: “take lots of toilet paper and don’t drink the water.” It later turned out that the water supply at the German Center for Artificial Intelligence was tainted, and I remain forever grateful...

Dave Poole loves to argue. He won’t admit it, but he does. And he is infuriatingly good at it, as all of his graduates agree. The strange thing is, that after years of bickering, and endless iterations of frustrating non-felicitous utterances and failed mutual beliefs with unintended perlocutionary effects, his students still like him. Valerie McRae *has* looked in on these often boisterous meetings from time to time, wondering if it wouldn’t be prudent to invite the RCMP. Very strange, indeed...

Steve Gribble, for implementing the Next interface. It was a pleasure to work with him. He’s the best undergraduate I’ve ever met, with technical and interpersonal skills far beyond his years. Mike Horsch, who taught me everything I know. Steve Mason, for taking just as long to finish his PhD. Scott Flinn and Chris Roehrig, who know everything about Next.

Imouttahere.

Thanks

Thanks to my co-supervisors, Kellogg S. Booth and David Poole, and to the other members of the thesis advisory committee:

- Tom Calvert (Computer Science, Simon Fraser University),
- Bob Goldstein (MIS, Faculty of Commerce, UBC),
- Richard Rosenberg (Computer Science, UBC).

The examiners were:

- Jim Little (Departmental Examiner, Computer Science, UBC),
- Ricki Goldman-Segal (University Examiner, Education, UBC),
- David Chin (External Examiner, Computer Science, University of Hawaii).

The student reader was Michael Horsch.

I am grateful for all their contributions to this work.

Part I
Introduction

Chapter 1

Systems have it Easy

Each of the shots must be physically spliced with cement or tape to the shots that precede and follow it...

—James Monaco
How to Read a Film, page 117

Do not be afraid to seize whatever you have written and cut it to ribbons; it can always be restored to its original condition in the morning...

—*The Elements of Style*, page 72

The work reported in this dissertation is rooted in the belief that the best way to make progress towards cooperative computational systems¹ is to take stock of human capabilities and limitations, and then to pursue human-computer relationships which exploit these capabilities and overcome these limitations. A human-computer symbiosis is warranted, where each participant in the relationship does what it does best. Advances in the foreseeable future will likely revolve about the

¹“Instead of the passive-aggressive error messages that are currently given in response to incorrect or incomplete specifications, intelligent agents should collaborate with the user to build an acceptable request.” [1]

design of this symbiosis, rather than the embodiment of intelligence in some computational artifact. There are at least two complementary approaches to achieving this goal.

One way is to build superior interfaces with better affordances that clearly advertise their function to the human user, and that cater to known human psychophysics [151]. Research in Human Computer Interaction (HCI) pursues this approach in many directions—some of it is surveyed in Chapter 3 of this dissertation. Central to the HCI approach is the desire to make the system easier for the user to understand [122, 123]; *the user should be able to acquire and exploit a model of the system.*

The other, complementary way stems from the realization that until now, it has been the user that does all of the explicit modelling, and that perhaps we have reached a stage where the computer can be made to bear at least part of the burden of representation. Research in User Modelling [115, 116] takes this approach; *the system should be able to acquire and exploit a model of the user.*

This dissertation focusses on the second approach.

User-guided theorem-proving systems and diagnostic expert systems are obvious examples of tasks in which humans and computers collaborate to achieve a goal. In environments like these, where the goals of the user are known by the system *a priori*, the job of acquiring a restricted model of the user is reasonably well-defined. Even in relatively open-ended application environments, the fact that the human user has chosen, say, a word-processing program rather than a drawing program, provides some grounds for model building. In contrast, vague or implicit user-goals in broader areas like decision-support systems (See, *e.g.*, Goldstein [84]) or desktop publishing and production environments makes acquiring and updating models of users very difficult.

This dissertation develops a particular approach to acquiring and exploiting models of users. The approach is applied to authoring and a prototype applica-

tion for video authoring and browsing is described. The introduction first defines authoring, and then offers a new perspective on traditional approaches. This perspective is one of the contributions of this thesis, and yields insight into the heretofore unregarded limitation of the traditional approaches, as well as insight into a new strategy called intent-based authoring which overcomes these limitations and is developed in this thesis. The goal of this thesis is to provide the intellectual and logical foundations upon which intent-based authoring systems can be developed.

1.1 Introduction

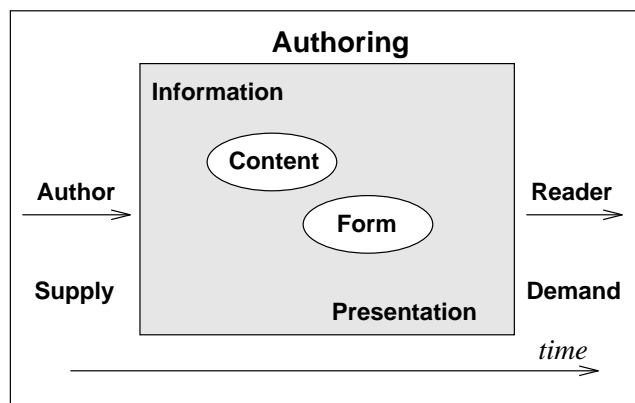


Figure 1.1: **The traditional approach to authoring**

Authoring is the honorable tradition of collecting, structuring and presenting information in the form of a “document” rendered in some medium or media. Until recently, the document has been static, in the sense that once rendered, it is fixed for all time and for all readers. Promising new technologies have recently come into existence that could alleviate some of the limitations of this difficult, knowledge-intensive undertaking.

1.2 Traditional Approaches

In the traditional model of authoring, the task of an author² is to collect a coherent body of information, structure it in a meaningful and interesting way, and present it in an appropriate fashion to a set of readers (or viewers) of the eventual work. This traditional notion of authoring commits the author to the form as well as to the contents of the work, well in advance of the actual time at which it is presented. Figure 1.1 emphasizes that there is no clear separation of information from presentation, and authors are committed to both the form and content of their message.

Structured-document approaches separate form and content, but user-tailored presentation is still not possible; reader “demand” only indirectly affects the authoring process. The familiar book format conveys the force of the general problem; once printed, there is no way—short of second-editions and published errata—to change the presentation for the particular needs and desires of individual readers, or groups of readers.³ The author must both select and order in advance the information to be presented. Presentations tailored to the needs of particular audiences are not possible in the traditional approach to authoring, with its “compile-time” commitment to form as well as to content.

The traditional approach to authoring when applied to non-traditional media like film, results in the same limitations. As an offshoot of his semiological analyses of the cinema, Metz [134, p.45] wrote that “the spectatorial demand cannot mould the particular content of each film ...” Metz is pointing out that when viewers sit in their theatre seats munching popcorn, it is too late in the traditional model

²The on-line copy of Webster’s 7th Dictionary offers the following definition:

1: the writer of a literary work (as a book)

2a: one that originates or gives existence: SOURCE <trying to track down the author of the rumor><the author of a theory>.

³That some books are published in multiple editions—the *Windows* versus the *Macintosh* edition of a manual, or the *Prolog* versus the *Lisp* edition of a programming text, for instance, does not address the general problem. Both these groups were anticipated by an author at compile-time. Not all individual readers can be anticipated in this way.

for their goals and desires to influence the content of the celluloid images being projected before them.

Such statements—though accurate in 1974—are representative now of what should be considered out-dated, traditional approaches that take a technologically imposed “supply-side” view of the authoring process, in which authors and publishers join to decide both the form and the content of a document before readers ever make their wishes known. The principal limitation of these traditional approaches is the resulting “one-size-fits-all” static document, exemplified by the venerable book format that we have been using since well before Gutenberg, when scribes laboriously and meticulously copied manuscripts; identical replication was the *sine qua non* of these technologies. Most approaches to authoring are even today just bigger and faster versions of the printing press, and do nothing to overcome this early binding problem.⁴

Today we can do better. We now have fast graphics, powerful reasoning engines and other technology, and rather than just add horsepower to traditional techniques, we can harness these new technologies to change the way authoring activities are conducted. Before continuing the exposition of this new, non-traditional authoring paradigm, we argue that at least two “new” strategies fall within the traditional model and still suffer from its limitations.

Hypermedia: (See Section 3.8) is media which can be accessed non-linearly, or non-sequentially, and it is nothing new. The terminology became common when non-linear documents became computerized, but hypermedia has been with us for a long time in the form of indexed documents (*e.g.*, encyclopaedia), footnotes that reference other parts of a document or other documents, and so on. Although tables of contents and elaborate indexes are intended as remedies to the static document format, the burden of this approach to overcoming the “one size fits all”

⁴“Binding” is used here in the computer scientist’s sense of associating values with variables. The pun was originally unintended.

problem falls heavily upon the reader. For instance, an encyclopaedia is a hyperdocument that can be browsed using the indices and cross-references as navigational links. The browsing activity completes the selection and ordering functions normally performed by the author and brings with it an inherent overhead that must be assumed by every reader. The viewer completes the job of the author by selecting and ordering the information to be viewed through the process of navigating the links established by the author. This not only pushes aspects of the problem from one person (the author) to another person (the viewer), it also dramatically increases the demands on the author who must provide explicit navigation cues in addition to the traditional authoring tasks. Reducing the amount of human effort required from the author and viewer is still a significant problem with current approaches to (hyper-)authoring. These effects can be mitigated by the knowledge-based approach advocated in this dissertation. See Section 3.8 for more exploration.

Form versus Content: An author chooses not only the information to be presented (the content) but also the order and style in which it will be presented (the form). Both contribute to the effectiveness of a presentation, yet few people are highly skilled in all aspects of these processes. This problem is at least partially addressed by the structured document paradigm, which attempts to separate the specification of the content of a document from the specification of its form. Markup languages like SGML (Standard Generalized Markup Language) and Hytime [146] are characteristic of this effort. They permit a delayed binding for what we might call the “surface structure” of a document (the format in which it is finally presented), but they still require the author to provide the “deep structure” (a hierarchical decomposition of the content as a structured document). See Section 3.7 for more details.

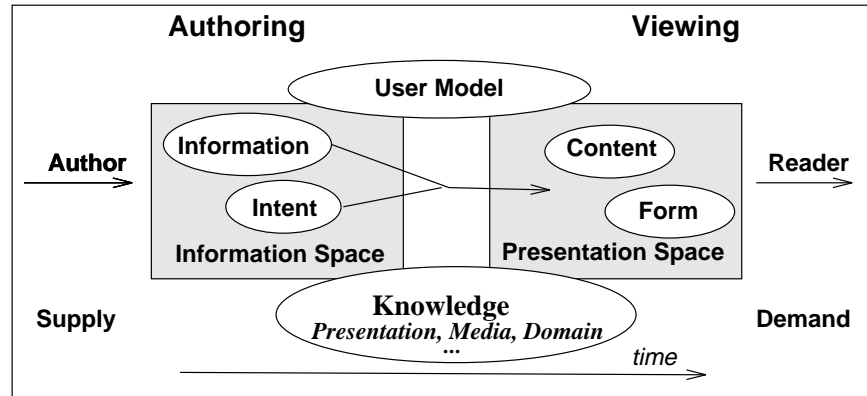


Figure 1.2: **The intent-based approach to authoring**

Content versus Intent: In order to tailor presentations to the needs and desires of individual readers, we need consultable models of these readers. For the “demand-side” of the equation to have a direct effect on the form and content of the document, decisions about the final presentation must be delayed until “run-time,” when the model of the reader can be brought to bear on the final stages of the design process. One difficulty is that user modelling is a new and complex problem.

As part of this thesis, techniques for user modelling have been developed and applied to the authoring problem. Thinking of authoring in terms of the knowledge required to support the activity has resulted in a new approach developed in this thesis called “intent-based authoring,” which may ultimately resolve the principal problems with the traditional approach.

1.3 Intent-based Authoring

A more complete de-coupling of specification and presentation processes is required before the goal of truly personalized presentations is attainable. In addition to the content of the document, the author must also supply an *intent*. The

author's intent is an arbitrarily complex communicative goal analogous to the notion of illocutionary force in the literature of *speech-acts* (see Section 4.3 for more on Speech Act Theory and Section 5.3 for more on authorial intent as used in this dissertation), but can be safely interpreted in the context of this dissertation in its typical dictionary definition, which offers as synonyms: *intention, intent, purpose, design, aim, end, object, objective*.⁵

This authorial intent is usually implicit in the work; a newspaper article is (sometimes) written to inform, an editorial to convince, a dissertation such as this to argue for the acceptance of a new authoring paradigm, and so on. The author's intent is a (possibly abstract, very high-level) communicative goal.

Making explicit this intention at the time the document is specified opens the door to truly user-specific document presentation. Information and presentation spaces can be clearly separated, bridged by various knowledge sources. In particular, a model of the viewer permits user-tailored determination of content at runtime; supply meets demand. Illustrated in Figure 1.2, we call this approach to authoring *intent-based authoring*, and describe here an application of the approach to the authoring of video documents. Video is used as proof-of-concept because it has characteristics which make it a popular recording medium, and because it is in many ways more difficult to deal with than other media (see Section 3.5.1); the intent-based approach to authoring advocated in this thesis is expected to apply equally to other media.

MacKinlay [130], Karp and Feiner [106] and others have argued similarly in the domains of graphical presentation and animation. Feiner explicitly uses the term "intent-based presentation." Previous work in automatic presentation has dealt with some aspects of the issues addressed herein, though it has been restricted for the most part to choosing "the right way" to display items based on their syntactic form [130, 175]. Semantic qualities of the data to be displayed are seldom consid-

⁵Source: Webster's 7th Dictionary, on-line copy.

ered.

Unlike Karp and Feiner [106], who describe a system for intent-based animation, we do not start with a perfect model of the objects to be viewed and then decide on the sequence of video frames to be generated. Instead, we start with a typically large collection of pre-existing video frames (usually sequences of frames) and select and order these to communicate the intended information. Our task is one of (automatic) “assembly,” rather than (automatic) “synthesis,” a different problem entirely. A *presentation* for our purposes is an edit list which specifies the order in which a selection of video clips is to be played.⁶ In different terms: if the presentation is about a cube in space, our model is not of the cube, but of the video tape whose subject matter is that cube; we do not model the cube, with its size and position, but the tape, with its frame numbers and contents.

Recently, other researchers have considered related problems. Hardman *et al.* [94] undertake to free multimedia authors from having to specify all the timing relations for presentation events; some of these are derived by their system at run-time. Goodman [85] also build presentations on-the-fly from canned video clips and other information. The work reported in this dissertation focusses on user modelling, rather than the media and domain concerns that motivate most other work. As shown in Figure 1.2, the user model is a crucial bridging element between the authoring activity that takes place at compile-time in the absence of the eventual viewer, and the viewing activity that takes place at run-time in the absence of the author.

⁶Such a characterization deliberately excludes from consideration details of how clips are to be visually related (*i.e.*, special editing effects like cut, fade or dissolve), attributes of the playback (*e.g.*, screen contrast, color balance, etc.) and other aspects of video authoring that could easily fall within the purview of a framework of this sort.

1.4 *Valhalla* and the Departmental Hyperbrochure

The prototypical application called *Valhalla*, described in Chapter 6, is an intent-based authoring and presentation system. *Valhalla* is an intent-based implementation of the *University of British Columbia Department of Computer Science Hyperbrochure*. Originally conceived as a static one-hour video presentation, the hyperbrochure has been pressed into service as a prospecting tool for students, staff, faculty, granting agencies, industrial partners and other internal and external interests; this usage is asking much of a single, linear presentation. The needs of one group of viewers are quite different from others, not to mention the differences in the particular interests of individuals within these groups. The need for a more versatile way to show different people what the Department of Computer Science has to offer was identified, and *Valhalla* emerged partially in response to this need, and because it represented an opportunity to deploy the results of this research.

The Departmental Hyperbrochure now consists of two thirty-minute video disks that include an introduction to UBC's Computer Science Department by its head, interviews with most of the faculty and staff, as well as walk-throughs of the laboratories.

The remainder of this section is a walk-through of an actual sample session with *Valhalla*. The reader might keep this example of the usage of the system in mind while reading other parts of this thesis. The same example is treated in more technical detail in Section 5.4.1.

1.4.1 Sample Session

Tom, a faculty member at the Department of Computer Science, arrives in one of the department labs with Joan, a visiting student from the University of Toronto. Joan is considering transferring to UBC, and wants to find out more about the department. She doesn't have her own computing account there yet, so Tom logs

in and starts up *Valhalla*. The system derives an initial model of Tom based on widely available information indexed by his login id; *Valhalla* knows that Tom is faculty, and that he is local to the department, and assumes that his gender is male because there is currently a higher percentage of males than females in the department. The system also knows that Tom belongs to the graphics research group and assumes that his principal research interests lie in that area. Based on this initial model, *Valhalla* prepares a presentation consisting of a set of video clips from the Hyperbrochure. Tom leaves to do other work, and Joan takes his place.

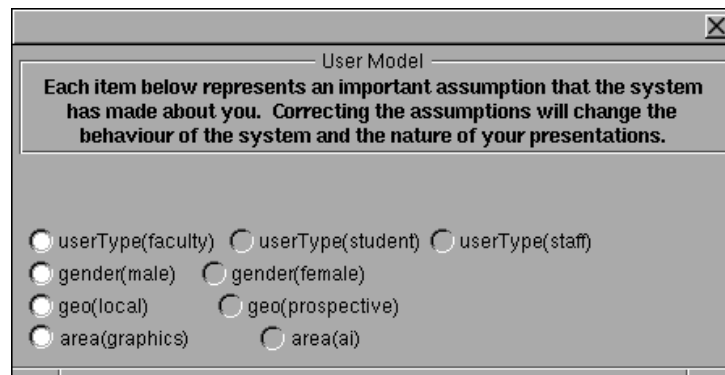


Figure 1.3: **The *Valhalla* User Model Window**

She watches some of the basic introductory material with which the presentation begins, but begins to wonder why the material goes on to talk about graphics, until her gaze falls on the user model window, in which the system has displayed the assumptions that have had the most effect in preparing the current presentation (see Figure 1.3). Joan sees that the system thinks she is male, local, and interested in graphics research. She clicks on the interface to correct the obviously false assumptions and instructs the system to design a new presentation based on this revised model, by manipulating the virtual VCR interface shown in Figure 1.4.

The new presentation includes clips about the AI research group and its labora-

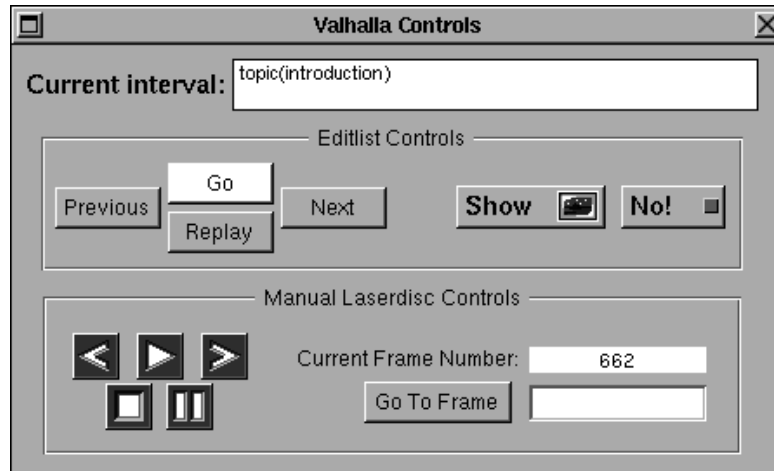


Figure 1.4: **The Valhalla Control Window**

ories, as well as a sequence of scenic views of the Vancouver area and opportunities for entertainment. A brief history of the University concludes the presentation.

1.5 Overview

This dissertation is divided into three parts. Part I is an introduction.

Part II is a literature review, consisting of Chapter 2, a survey of existing approaches to the modelling of agents, and Chapter 3, a survey of the broad spectrum of authoring systems with respect to a number of characterizing dimensions described in Section 3.1. Chapter 4 is a review of relevant theoretical material upon which the contribution is built, including hypothetical reasoning, probability and decision theory.

Part III describes the theoretical and practical contributions of this dissertation: Chapter 5 goes into some detail about the reasoning framework adopted to support the intent-based authoring paradigm, and Chapter 6 describes a prototype imple-

mentation built to demonstrate these ideas.

Part IV concludes with Chapter 7, advancing some generalizations and implications of the intent-based authoring and presentation paradigm, as well as some proposals for further work.

Although the wide space of authoring sampled in Part II provides many useful directions for practitioners interested in intent-based presentation, the field is largely an unwritten book, waiting for integrative contributions from researchers in user modelling, psychological perception, and artificial intelligence.

A number of “scenarios” are scattered throughout this thesis. These are intended to give the reader a sense for the philosophy and goals of the intent-based authoring paradigm, and although some of them are unabashedly science-fiction, all behavior described in the scenarios can be implemented by addressing technical and non-theoretical issues. A central component in all example scenarios is the underlying model of the users involved, necessary to achieve the functionality described.

Full code listings are available over the Internet from the author.⁷

⁷The author can be reached at csinger@cs.ubc.ca, and information is available at <http://www.cs.ubc.ca/spider/csinger/home.html>

Part II

**Literature Survey and Formal
Background**

Scenario: Information Retrieval

The reader is warned that the following scenario is science fiction; the intent is merely to motivate the reader with the long-range goals of the intent-based authoring paradigm, which is to move away from telling computers how to do things, to telling them what to do, and finally to just telling them about ourselves.

Information Retrieval

Dan and Mike are both users of FAST, the powerful information retrieval system of the twenty-twenties. FAST is connected to a bewildering variety of widely dispersed databases on all aspects of human endeavor, and operates –as its acronym suggests– at great speed over high bandwidth networks.

When Dan asks the system for the names and descriptions of deadly viruses, FAST begins its response after accessing various cross-indexed medical and historical databases with the story of the eradication of HIV, the last virus to be tamed by medical science. Dan is a doctor.

When Mike presents a similarly formulated query, the system begins with a description of Michaelangelo, the computer virus that threatened to destroy PC disk-drive information on the artist's birthday in nineteen ninety-two, in the dark, depressing, early days of information retrieval. Mike is a computer programmer.

Chapter 2

User Modelling

“You have a time machine and you use it for ... watching television?”

“Well, I wouldn't use it at all if I could get the hang of the video recorder”...

—Douglas Adams

Dirk Gently's Holistic Detective Agency

The loosely defined area of user modelling has grown over the past decade out of its origins in the field of natural-language dialog systems [196], into a wide range of disciplines concerned with developing cooperative systems for heterogeneous user populations [115]. Some of the more prominent disciplines include Human-Computer Interaction, Intelligent Interfaces, Adaptive Interfaces, Cognitive Engineering, Intelligent Information Retrieval, Intelligent Tutoring, Active and Passive Help Systems, Hypertext and Expert Systems.

Although the specific reasons for modelling agents are manifold, the general appeal of the undertaking is that it promises more effective use of the available communications channels between the agents being modelled. Most of the attention has focussed on *user* modelling, a special case of agent modelling, where the agent being modelled is the (human) user of an interactive system. The emphasis in this setting is to increase the cooperativeness of the system *vis a vis* the human;

this thesis shares the view that user modelling is a likely vehicle to engender more cooperative behavior in human-machine interaction.

Kass and Finin [107] provide a useful framework for presenting the user modelling literature by categorizing it along six dimensions. They choose to analyze models as to their ■ degree of specialization, ■ modifiability, ■ temporal extent, ■ method of use, ■ number of agents, and ■ number of models. The present discussion will be related to this framework.

2.1 Overview

A number of classes of research activity have been subsumed under the label of *agent modelling*. The literature divides fuzzily among natural language understanding and generation, computer aided instruction or intelligent tutoring systems, and cognitive modelling. The thread of information retrieval runs through these somewhat arbitrary divisions [118] [141]. Each of these areas has its own reasons for modelling agents.¹ Recent efforts at developing *user modelling shells* promise to take the field from the *ad-hoc* to the systematic.

In natural language, there has been growing consensus that models of agents are necessary to both understand references in user utterances, as well as to generate statements that will be comprehensible to the user; modelling is understood to be a necessary component of natural language dialogue [35] [36] [107] [91] [178]. These observations expand readily from text-only to multi-modal and multi-media environments: the presentations produced by a system for the consumption of a user should be tailored to the user's expectations, abilities, and goals.

Practitioners of computer aided instruction (CAI) [198] have made use of models of student users (student models) to decide what to teach, and how to teach

¹See [12] for an annotated bibliography of the field, divided into the areas of Computer Aided Instruction, Expert Systems, Knowledge Representation, Logic Programming, Natural Language, the Philosophy of Science, and User Modelling *per se*.

it [50]. The interests of this research community address the question of how a user acquires an accurate model of the system (the *system model*), as opposed to how a system can acquire an accurate model of the user.

Cognitive modelling [101] has as its goal the development of psychologically valid models of human cognition. Although researchers in this area generally build computational systems to test their theories, these systems can sometimes form the bases for useful modelling tools. For example, Craddock [53] develops a model for database retrieval which explains aspects of common-sense reasoning in humans, and which admits of a relatively direct implementation path; his implementation could be used as an episodic knowledge base (EKB) that stores user models. Provan and Rensink [168] refer to psychophysical results to support their model of neural connectivity and Marcus [131] also leans on psychological findings to defend his novel approach to parsing natural language; such efforts should not be ignored when building systems to reason their way to user models from observation of human responses to visual and textual stimuli.

Different strategies and technologies have been employed to address the difficult problems associated with user modelling. *E.g.*, default reasoning [103] [50] [60] [13], truth maintenance [98], and Dempster-Schafer analyses [178]. An early approach to user modelling, which is still the subject of some current research and which is only now beginning to find its way into commercial products, is *Stereotyping* [17] [45] [170] [169].

A stereotype is a collection of data which typifies a class of users. Rich [171] defines stereotypes as a means of making a large number of assumptions about a user based upon only a small number of observations, and she pioneered their use in the Grundy system for recommending books to users. The data are usually represented in some kind of logical calculus as statements of belief or knowledge or goals, and there will be as many stereotypes as there are identifiable classes of users. The approach involves first identifying a fixed set of classes of users *a priori*,

then deciding the membership of an individual in one of these classes, and finally attributing the contents of an applicable stereotype to the individual. The accuracy of user models which rely upon stereotyping depends directly upon the number of pre-determined user categories.² The more stereotypes, the better, although the difficulty of determining which stereotype to apply to an individual grows with the number of stereotypes. The activity of choosing which stereotype to apply is called *triggering*. Variations upon the theme of stereotyping have been developed in a number of directions; individuals can be permitted to inherit characteristics from multiple stereotypes (*e.g.*, [16]), an approach which affords more accurate modelling at the expense of more elaborate means for arbitration between applicable, mutually inconsistent stereotypes. Stereotypes may also be ordered into taxonomic hierarchies which provide savings in the space required for the representations: if the stereotype of a computer novice contains typical beliefs of novice computer users about the operation of a computer system, then the stereotype of an advanced user need only contain typical beliefs of advanced users where these conflict with, or do not appear in the novice stereotype. The advanced stereotype inherits the contents of the novice where the latter does not conflict with the former. Taxonomies can simplify the attribution process; if it has been determined that a user is an expert, for instance, there is no need to verify stereotypes higher in the hierarchy.

An extension to stereotyping is what Ballim [16] has called “ascription by perturbation,” wherein an agent assumes that another agent is similar to itself and therefore attributes its own beliefs to the other. This approach offers considerable power in environments where a large fraction of the agents’ knowledge is common. The usefulness of the approach is illustrated by how well it works in our

²An early, typical example is the beginner-intermediate-expert distinction exploited in some popular off-the-shelf word processing programs primarily to vary the verbosity of their help-messages. The assignment of the user to the appropriate category was determined by the user himself, thereby solving the thorny acquisition problem.

own lives; we never know what our fellow humans are thinking or believing, but we rely on our own introspection and perform attributions with internal justifications like: “If I were him, I would be hungry by now...” A canonical perturbation ascription rule would be: “*assume that another agent’s view is the same as one’s own except where there is explicit evidence to the contrary.*” [16, p.76]. Humans appear to be remarkably successful with this approach.

Chin [45] [44] introduces the notion of *double stereotypes*. In addition to categorizing users, his KNAME system categorizes information into levels of difficulty, so that inferences can be represented as relations between user types and difficulty levels. (“*Experts know all simple and mundane but only most complex knowledge.*” “*Beginners know most simple, a few mundane and no complex knowledge.*”)

A promising improvement to the technique of stereotyping is to dynamically derive membership categories from episodic databases of user activity. *Doppelgänger* [155] is a generalized user modelling system that uses learning techniques to interpret data about users acquired through a variety of sensors. Applications connect via standard protocols to a server that provides access to user data. Each user model is a point in a very high dimensional space whose dimensions are determined by the available sensors; this point moves through the space as information about the user is gathered. The categories in *Doppelgänger* are called *community models*, which are computed as weighted combinations of member models, “and thus change dynamically as the user models are augmented.” Section 7.2 of this thesis considers the use of learning techniques in future work.

A strategy related to stereotyping is the use of *profiling*, which involves giving users control over various aspects of system operation by allowing them to set the values of a prescribed set of parameters. Common examples are the tailorability of the Unix operating system with scripts and alias mechanisms, and the customizability of the Xwindow interaction environment. There is also a wide range

of application programs which offer user-modifiable operation (*e.g.*, [54]).

Another approach being studied in the research community is the use of plan-recognition. Recognizing the plan of a user permits the derivation of his or her goals and intentions. This has been employed to help provide pro-active user feedback when faulty plans are recognized [189].

Problems with plan-recognition have been the management of uncertainty and the prohibitive size of the plan library required for serious applications. Various default reasoning techniques have been applied to the former (*e.g.*, weighted abduction [8]), but the latter difficulty has hardly been addressed (but for an exception, see the PHI system [20]). Plan recognition is usually performed [109] under the strong assumptions that 1) the recognizer agent has complete knowledge of the domain, and that 2) the agent whose plan is being inferred has a correct plan. These assumptions are clearly not universally true: users may find ways of doing things that the designers of the systems had not considered, and users all too often have faulty plans usually based upon faulty or incomplete models of the systems they are using. These assumptions are explicitly tackled by systems which try to model the faulty plans that users might have (*i.e.*, Bauer [20] and Thies [189]), which in turn have to deal with a potentially infinite number of faulty plans. Plan-based approaches have grown out of basic artificial intelligence research, and applications are few, but some work is already being done to determine how useful plan recognition can become to a range of interaction environments [86].

An interesting twist on the user model is the *Programmable User Model* [204]. This tool embodies psychologically motivated constraints in a programming environment which interface designers use to build a model of a user. In principle, such tools will ensure that any task in which a user may engage will be computationally supported in cognitively sensible ways.

The remainder of this section examines in more detail some issues of agent modelling that are relevant to the current work.

2.2 Issues: Acquisition and Exploitation.

The problem of agent modelling divides broadly into questions relating to the *acquisition* and to the *exploitation* of agent models. Agent models must be acquired. Even the use of user stereotypes requires determination of the user's membership class, although the issue can be circumvented by simply asking the user to decide for himself or herself.³

2.2.1 Acquisition

Acquisition is of two varieties: the system model must be acquired by the user, and the user model must be acquired by the system. The emphasis in this thesis is on the latter process.

Kass and Finin consider acquisition along a dimension whose extrema are 'implicit' and 'explicit' forms.⁴ Explicit acquisition can be as simple as asking the user to fill out forms or enter descriptive keywords; this data can then be used, for instance, to determine the user stereotype. Systems employing explicit acquisition have been called 'adaptable' [76] (*cf.* the 'computer as tool' metaphor [108]).

Implicit acquisition is generally more subtle, requiring that the model be inferred from observation of the user. An approach that has met with some success is that of monitoring the communication between user and application with the aim of inferring all or part of the user model. For instance, Csinger and Poole [60] [55] employ a normative theory of inter-agent communication based on a Gricean analysis [89] to derive the beliefs of interlocutors in a natural language setting. Their system is implemented in a logical framework for common-sense reasoning [158].

Zukerman [207] presents a planning mechanism which she uses in conjunction

³As users are typically not very good at deciding such things [169], other methods are desirable.

⁴Recent work makes this distinction in various forms. Laurel, for instance: "Increasingly, systems will need to employ either explicit conversations with people to determine task objectives or implicit user-modeling techniques to infer objectives from behavior..." [123, p107]

with a model of user beliefs and inferential capability to predict the possible (perlocutionary) effects on hearers of utterance components. Modelling these effects in this way permits a traditional anticipation-feedback loop for utterance design. These approaches have the common aim of modelling potential perlocutionary utterance effects by recourse to a Gricean model of dialogue. Zukerman continues this line of investigation [163] with a system called RADAR that generates both responses and queries of two types: disambiguating queries and queries to elicit additional information. Decision-theoretic measures are used to determine the dialogue strategy. Wu [200] uses decision theoretic techniques to decide when explicit interaction with the user is to be preferred over implicit hypothesizing, by maximizing the expected utility of the intervention. The work described in this thesis also moves in this direction; see Section 5.5.

Kass and Finin mention a number of other efforts taken along similar lines. The implicit approach to acquisition promises to extend into the multimedia environment, as soon as the nature of interaction with these new technologies can be captured in a set of normative rules.

Kass and Finin separate the issue of acquisition into the acquisition of goals, plans, and beliefs, suggesting that acquiring beliefs is the hardest of all.

2.2.2 Exploitation

The exploitation of models is highly task-dependent, but some broad distinctions can be drawn. Kass and Finin's framework identifies the 'method of use' dimension. They present what they imply is a continuum between 'descriptive' and 'prescriptive' models.

The difference between descriptive and prescriptive models may be nothing more than the style in which they are employed. Once acquired in some fashion, a model may be consulted for a variety of reasons; if an explanation is sought for the behavior of an agent, then that model may be called a descriptive model. The

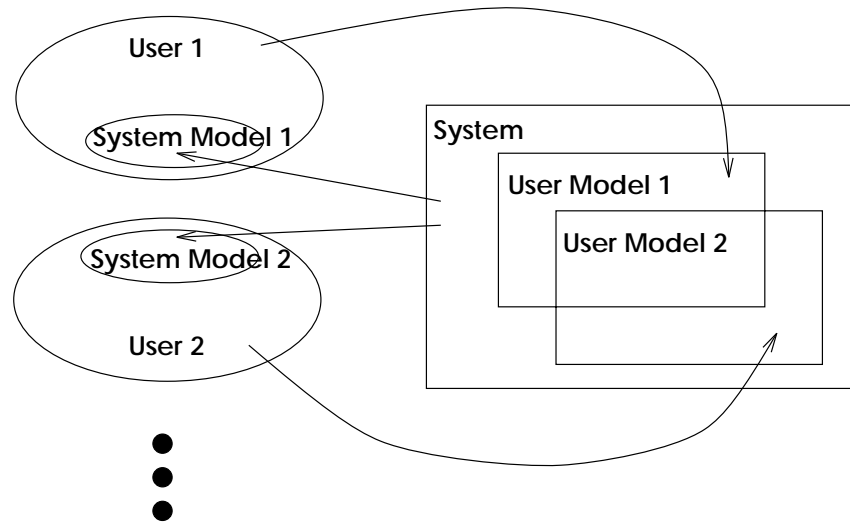


Figure 2.1: **User and system models**

same –or another– model consulted with the intention of tailoring system presentations to an agent’s knowledge, for instance, is being used in a prescriptive sense. A descriptive model of an agent may or may not be accurate; this accuracy can be tested by comparing predictions of the agent’s behavior with actual, observed behavior.

A normative model is one which canonizes normal, expected behavior: it can be employed descriptively to explain the behavior of an agent, as well as prescriptively to anticipate the agent. When predictions on the basis of a normative model conflict with actual, observed, behavior, the observed behavior of the agent can be interpreted as ‘wrong’ in some sense. The correctness of the normative model is not questioned. Some domains admit of such models, others do not.

One agent may refer to its descriptive model of another agent to decide its actions. A typical example of this usage is sometimes referred to as the *anticipation feedback loop* as found in natural language dialog systems which refer to a model

of their human interlocutor to ensure that the utterance under consideration will be acceptable to the user.

2.2.3 Correction: Situations requiring correction.

The acquisition task is never complete. Changes in the system or in the domain of interest may occur from time to time, and will require the user to update her system model. It is the task of the system to present information to the user in a manner that facilitates awareness and comprehension of these changes. This is just the kind of pedagogy CAI researchers have been exploring [198].

Likewise, the user may forget information over time, or the user model may have been incorrectly acquired in the first place; either situation requires remedial action by the system.

A number of scenarios present themselves: ■ Beliefs erroneously attributed to users which they do not have (when these beliefs are, in fact, true, a situation arises called false consensus). ■ Beliefs not attributed to users which they do in fact have (pluralistic ignorance, special case). This is a serious acquisition problem. ■ Erroneous beliefs correctly attributed (user misconceptions identified). ■ Ignorance: simply having no beliefs in respect of the proposition concerned.

Bel_{user}	$Bel_{system}(Bel_{user})$	Bel_{system}	
a	a	a	normative
b	$\neg b$	$\neg b$	false consensus
c	$\neg c$	c	pluralistic ignorance
d	d	$\neg d$	misconception found

Table 2.1: **Errors in user models.**

See Table 2.1 for a summary of these error situations.

These categories of modelling error grow more interesting with the number of agents being modelled. For instance, in the case of a CSCW environment where

multiple participants are engaged in a negotiation task in which mutual consensus is the desired outcome, a facilitator agent might more easily identify cases of false consensus than any individual participant; the facilitator will certainly be able to act more easily upon this information than any of the individual agents. In negotiation tasks where it is the underlying goal of all participants to maximize joint outcome, some or all of the participants may not believe that some or all of the other participants share this goal. A facilitator or mediator agent might be able to act on recognition of this case of pluralistic ignorance to the benefit of the group and its common goal.

Kass and Finin advance as one of the dimensions of their analysis the notion of ‘modifiability,’ intended to distinguish models along a range between those which are static, and those which are dynamic. It is only dynamic models which will admit of correction, and the authors point out that ‘user models that track the goals and plans of the user must be dynamic.’

2.2.4 Scope: What is Represented?

A model of a user divides naturally into two components: the normative or generic component, and the specific. The generic component models the abilities and limitations of normative humans. This includes such quantities as psychophysically derived limits to visual resolution (see Section 3.4), color preferences and even certain typical pathologies as colorblindness. Although this component may need to be modified for ‘abnormal’ users, it can in principle be acquired once and for all from psychophysical studies and putative cognitive models.

The specific component relates to the goals and beliefs (*e.g.*) of a specific, individual user. It is distinguished from the generic model in that it must be acquired and maintained for each individual. While the generic component will be useful to ensure that systems present information in cognitively sensible ways, it is the specific component which will induce adaptable, user-sensitive cooperative oper-

ation, and is the target of the present investigation.

Kass and Finin distinguish between models which are ‘individual’ and those which are ‘generic,’ recognizing that there is a continuum between these extremes. The stereotyping approach outlined above lies somewhere along this continuum, particularly since various hybridizations are possible, such as creating a hierarchy of stereotypes to better accommodate variation in agents without dramatic increases in storage requirements.⁵

2.2.5 Extent and Adaptivity

Kass and Finin also discuss the ‘temporal extent’ of the model, arguing that the useful lifetime of information varies. The maintenance of the model should be subject to conditions of controlled ‘forgetfulness,’ where information about the agent that is judged to have outlived its usefulness according to some criteria is deleted, or forgotten. This notion is not pursued in this thesis; the issue of how elaborate a temporal representation is necessary is orthogonal to the investigations of this dissertation.

The information in the models should be faithful to the composition of the user population, which may change. Predefined user categories such as conventional stereotypes may be inaccurate, misconceived, or out-of-date. Various means of adapting to the user population have been considered, from as early as the Grundy system [171], to the approach described in this dissertation.

⁵Such an approach is immediately suggestive of an object-oriented agent model. The object-oriented methodology would yield the dual benefit of clean and separable information structures for the model which allow inheritance mechanisms, as well as well-defined accessibility via methods. See Chin [45] and Wilensky [199] for suggestive leads in this direction.

2.3 Multiple Agents, Multiple Models

In general, there will be more than one agent involved in a collaborative process, and the system will likely need to model some or all of them. Even in existing interactive systems, there is a need for some sort of multi-agent modelling. Kass and Finin mention medical diagnosis systems, in which both the user and the patient are to be modelled. Even though the user and the patient may be one and the same individual (doctors get the flu too, after all!), it is to the distinct roles in the task domain that the operation of the system is sensitive. This issue will emerge at various points in this thesis. In general, though, a separate model may be required for each agent-role.

Certainly in the case of systems designed explicitly to support the cooperative work of more than one individual—*i.e.*, systems designed to support *collaborative* work—and particularly for the class of such systems called decision-support tools, explicit models of multiple agents will be required. The existence of multiple users is considered in Figure 2.1. Acquisition relationships are suggested by the directed arcs. Not shown are models that users might have of other users. Moreover, it may turn out to be necessary to model agents' models of other agents, including the reasoning capabilities of these agents [13].

Kass and Finin distinguish between multiple agents on the one hand, and multiple models on the other. They suggest that multiple sub-models might need to be maintained for each user, since their levels of expertise may vary between (sub-)domains. They appear to consider only the stereotyping approach in their discussion.

2.4 User Modelling Shells

Parallel to familiar developments in other areas, there is growing interest among the user modelling community in User Modelling Shells. Just as user interface

management systems (UIMS) alleviate part of the systems implementation burden and enable cost-effective generation of non-trivial interfaces, a user modelling shell (UMS) provides services for implementing non-trivial modelling capabilities. Just as expert system shells made advanced knowledge-based techniques accessible to systems developers, UMS's promise to promote transfer of advanced user modelling techniques and technologies into application development environments, which in turn promises to fuel a new round of research and development in the field of user modelling.

Current UMS research focusses on developing "integrated representation, reasoning and revision tools that form an 'empty' user modeling mechanism. When filled with application-dependent user modeling knowledge, these shell systems would fulfill essential functions of a user modeling component in an application system." [115].

GUMS [75], BGP-MS [113] [114], and UM [111] are all efforts at implementing these functions, and all of them rely on the stereotyping approach described earlier.

GUMS [75] permits only single-inheritance in the stereotype hierarchy, and users can belong only to a single stereotype. If new observations about a user invalidate his or her membership in the current stereotype, GUMS moves upward through the stereotype taxonomy to a more general user stereotype. Revision of the user model in GUMS therefore results in a loss of information.

BGP-MS [113] [114] (Belief, Goal and Plan-Maintenance System) represents assumptions about the user in an extension of Prolog, and employs multiple inheritance in a partition hierarchy to extend models of individual users. The system provides various development and run-time services for developers of applications requiring user modelling. The developers of BGP-MS state their intention of adding an automated truth-maintenance system to provide incremental consistency of the user models, and they are investigating the use of modal logics for

increased expressiveness of the user models.

User Modelling Tool (UMT) [30] is a general purpose user modelling shell whose approach to user modelling falls into the class of assumption-based user modelling because it uses an assumption-based truth maintenance system (ATMS) to maintain the consistency of the user models. Stereotypes and production rules are the techniques used in UMT to generate and activate the models *per se*. A LISP implementation for Symbolics environments is available [31].

Although these efforts are perhaps somewhat optimistic in view of the early state of user modelling technology, they point the way towards the first commercial implementations.

2.5 User Modelling: A Definition by Consensus

A frequently asked question at the 1992 User Modelling Workshop was whether User Modelling was somehow the same as Interface Design, or more generally, if there was anything that was *not* User Modelling [4]. This question naively presupposes some consensual definition or understanding of what user modelling actually *is*, and demonstrates how easy it can be to mislead even self-professed practitioners in the field with the term *user-modelling*.

By 1994, the user-modelling community had outlived its detractors and outgrown the workshop format. Although practitioners were now willing to use the term without embarrassment, and as if they all agreed what it meant, there was still no “definition” by consensus at the 1994 User Modelling Conference, which enjoyed contributions from an even wider variety of disciplines.

A number of things in particular *are not* intended with the use of the term in this thesis. For instance, no cognitive adequacy of any sort is intended for the models described here; our approach to modelling agents does not purport to accurately represent the human users of systems to some arbitrary degree.

In its broadest interpretation, user modelling is nothing more than applying the user-centred view to the design and implementation of systems. Whether the system in question is an advanced user interface, a CAD system, a database retrieval system, or some other more or less advanced system, when the designers of these systems adopt the user-centred view described herein, they are engaged in user modelling.

In the rest of this document, the use of the terms *user modelling* or *agent modelling* is somewhat more specific, referring to the acquisition or exploitation of explicit, consultable models of either the human users of systems or the computational agents which constitute the systems.

Scenario: Intelligent Meeting Support

This is another motivating scenario, illustrating that models of individual users can be exploited at run-time to tailor the form and content of presentations.

Intelligent Meeting Support

The geographically dispersed executives of InterSpect Systems Consulting Corp. are participating in a meeting via the services of WOW, the intelligent meeting support system that has gained a great following in the early years of the twenty-first century. Participants benefit from a variety of media services including video, graphics, and text, all transmitted over high-bandwidth communication links.

Maria, the ebullient manager of central operations, is holding forth on the need to maintain quality by hiring only the best candidates for positions now vacant.

Ralph, the chief accountant of InterSpect, is silently preparing a financial report indicating that it might be difficult to pay the best candidates what they are worth, given the combined effect of the current state of the national economy and the company's debt load.

Fred, a member of the board of directors, hates numbers. Accounting bores him, and pie-charts infuriate him, but he is fascinated by a new tool for 3D visualization of multivariate data.

As Maria finally winds down at the urging of the WOW facilitation manager, which has determined that she really has had quite enough bandwidth to make her point, Ralph's report is made available to the other participants. Maria, who has a great grasp of numerical data, and who actually savors their visual impact on a page, sees in a corner of one of her displays the rows and columns of an old-fashioned spread-sheet representing Ralph's fiscal objection to Maria's policy. Fred sees a multi-dimensional scatterplot constructed by WOW to make maximal use of the capabilities of his visual system, and other participants experience other varied events in these and other modalities.

When Ralph draws the attention of other participants to Payroll Item Number 2070, perhaps only by uttering the expression in English, not only is his utterance simultaneously translated into the (natural) languages of the other participants and optionally presented on their audio channels, but these other events also oc-

cur: 1) A set of relevant numbers is highlighted on Maria's display, 2) when he is attending to it, the dimensions of Fred's scatterplot are interchanged to perceptually emphasize the data referred to by Ralph, and 3) similar events are experienced by the other participants.

Chapter 3

Authoring

Every style is but one valid way of looking at the world, one view of the holy mountain, which offers a different image from every place but can be seen as the same everywhere.

—Rudolph Arnheim

Art and Visual Perception: A Psychology of the Creative Eye, final paragraph

Style takes its final shape more from attitudes of mind than from principles of composition, for, as one elderly practitioner once remarked, “writing is an act of faith, not a trick of grammar.” ...style is the writer, and therefore what a man is, rather than what he knows, will at last determine his style.

—*The Elements of Style*, page 84.

This chapter takes a broad view of Authoring research and samples the range of work at a number of relevant focal points. To provide some order for the descriptions to follow, and to situate the work leading to this dissertation, the following dimensions of analysis are used.

3.1 Dimensions of Authoring

The work is grouped by the tasks they are designed to support. The work of Mackinlay, Casner, Roth as well as some of Feiner's is addressed under the heading of *Automatic Graph Generation*. Pickett and Grinstein are representative of work in *Exploratory Data Analysis*, while Ware and Beatty, Cowan, and Cleveland are presented along with *Psychophysical Research* performed by psychologists. Feiner, and McKeown, are best considered in the context of *Multi-media*. Bly and others are discussed under the label of *Computer Supported Cooperative Work*, and the themes of *Hypertext* and *Hypermedia* appear in a number of guises under these headings, as well as in their own section. Structured documents, cyberspace, and finally intent-based documents—the core of this dissertation—are explored in separate sections.

A number of systems are presented within the framework described above and their interrelationships are shown within the space defined by the dimensions listed below.

Deliverable: The deliverable is that which is being authored. There are sometimes [role-dependent] distinctions between artifact and deliverable: the deliverable is not always a physical artifact, and different agents (reflecting different roles) may emerge from the activity with different deliverables: from the author's point of view, the deliverable is the book, while from the publisher's point of view it may be the month-end sales figures.

Task/Data Domain: The domain refers to the field to which the task being expedited is relevant. This categorization may also be role dependent: the author of the programming manual may regard his contribution as being to the field of structured programming, while the editor may feel that she is contributing to the field of document design and production.

Temporal Scope: Is the emphasis on: presentation and object-level real-time support? representation and meta-level maintainability? Short-term versus long-term storage? Version control? How, in other words, does the system conceptualize and deal with Time?

Target Media: The target media are those supported by the system or work under investigation. Varieties of text, graphic, audio, video and any combination thereof are the target media of different systems, each of which may take different views of the relationships between medium and mode.

Mode: Flesh and Blood Interface: What attention is being paid to human perceptual capabilities and limitations [57]? To visual, aural and gestural primitives? Different aspects of the visual, aural and kinesthetic modalities define the space of human-computer interaction today. Are there other modes (senses) to be considered?

Style: Is there an over-arching notion of style in the system? Is it adjustable? In what ways, by which roles, and to what degree? What degree of inter-presentation coordination is allowed? (See Beach [21] and Cargill [37]).

Models: Are there explicit or implicit models of users, systems, agents, roles, ...? How are they employed and combined to achieve the functionality of the system?

Role relationships: Who is/are the Author(s), Reader(s)? Are they co-temporaneous? Are they co-present? Other task and domain specific roles include: editor, facilitator, chairperson, manager... The broad view allows for temporal shifting and stretching of the authoring process. The locus of creation of presentation can be shifted along a continuum from conventional author, to situations where the presentation is decided by a combination of conventional author and conventional reader, to situations where the presentation is designed with no input from any single agent resembling a conventional author (*e.g.*, a computer-

generated event log of system activity).

There are also questions about the number of actual authors and actual readers, and whether activities by these agents are simultaneous and distributed. Annotating is a special case of non-simultaneous authoring by one or more potentially distributed authors.

Heterogeneity: This issue ramifies into a number of questions at varying levels of analysis.

- **Internal (Human) Interface:** Is the interface to the system role-dependent? *E.g.*, Do readers and writers have the same interface, or is there a ‘modal disparity’ between them?
- **External Interface (Interoperability):** Is the system designed to cope with varied sources of information with differing protocols?
- **Organizational Interface:** Can different organizations using the same system make use of each other’s information spaces?
- **Seamlessness:** Can users of new systems integrate their existing tools and work-spaces? In particular, do new CSCW tools support the use of existing tools for individual work? (Can seamless transitions be made between individual and collaborative work? [125].) Some transitional elements are bound to remain central for some time: users may want to continue using pencil and paper, and systems should not force them to change their ways.¹

Information and Presentation Spaces: The *presentation space* is the arena in which the presentation takes place, composed of various media and directed at var-

¹There remain many advantages to hardcopy that will be difficult to displace, so for the foreseeable future, usable, practical electronic systems will need to interface with the paper and back-of-envelope worlds.

ious modes; the *information space* refers to that which is presented. These definitions provide useful categorizations of authoring systems.

- Static, or changing, dynamic information space?
- Static, or changing, presentation space?
- Static, or changing, presentation?
- Shared Information or presentation space. Particularly confounding problems arise when information is shared between synchronous collaborators. One of the important motivations for modelling users (see Section 2) is the desire to tailor presentations to individual needs, understanding and goals. This desire conflicts with synchronous collaboration when one participant wishes to direct the attention of another participant by direct reference to an element on his private display. This reference may be very difficult to resolve when the referent either does not appear at all on the private display of the second participant, or appears in some different form, or at a different location. (The example on page 33 presented a scenario intended to typify this kind of problem in the domain of intelligent meeting support.) The presentation of identical views² alleviates these difficulties by ignoring the problem of tailoring presentation.

The survey begins by situating a few traditional authoring systems within the space defined by these dimensions.

The familiar BOOK is itself a **deliverable**, across a wide variety of **domains**, not the least of which is recreation or entertainment. Its **scope** varies from the short-term throw-away paper-back to the archival of information in weighty, hard-bound tomes; the information in a book is not typically presented in real-time, except at poetry readings and for the delivery of bed-time stories to children. Readers

²Known as WYSIWIS [125], or *what you see is what I see* presentation.

can interrupt their progress at will, and the subject matter can refer to remote points in time. The **target media** is clearly paper of some sort or another, upon which the information is available to humans via their **visual sense**, unless they are in Braille, in which case they refer to the tactile sense; books also provide a pleasant kinesthetic feel. The pages of a book typically adhere to a discernible **style**, which lends familiarity to the on-going reading process. The only kind of **modelling** is of the implicit variety, undertaken at the time of writing of the book, and which reflects the author's desire to appeal in some fashion to the reader. Books typically admit only the roles of author and reader, though other roles are hidden in the production process: the touches of editors, publishers and translators sometimes remain visible in the final copy. There is usually –though not always– a single author and books are generally printed with the hope of attracting more than a single reader! Books fit **seamlessly** into the lives of most humans, and what problems exist are accepted out of long habit: books can be carried in hand-bags and attaché cases and read on crowded buses during rush hour, as well as in bed with a flashlight. Both the **information** (content) and the **presentation spaces** (form) of a book are static.

LETTERS are like books in most respects, differing notably on the dimension of role; there is typically a single reader to a letter, or a small number of secondary readers who may appear on a carbon-copy list in the document itself. A CORRESPONDENCE is a sequence of letters between individuals, and therefore admits of a temporal aspect.

A MOVIE is a 'book' whose target media is film or video, and which is accessible to human viewers via the visual and aural modes or channels (notwithstanding the efforts of Odorama and Sensurround to expand the viewing experience to other modes!) The scope is usually real-time with respect to presentation, although a video tape can be stopped and rewound at the viewer's discretion.

A traditional theatrical PLAY exhibits aspects of the movie, with the added

complication that the audience can affect the performances of the actors in various subtle ways; the **presentation space** is dynamic. The notion of INTERACTIVE THEATRE (*e.g., theatre-sports*) amplifies these audience-feedback effects [123]. It is not as easy to rewind a live performance as it is a video tape.

A VIDEO GAME of the arcade variety uses a combination of graphics, video, sound, and sometimes motion feedback to provide a multimedia simulation of an alternate reality. Some games permit multiple players to share the alternate reality (shared **information space**), either on the same video screen (shared **presentation space**) or on separate terminals. The video game is even further along the interactivity continuum, permitting its users to modify not only how information is presented, but its content as well; the information space is dynamic.

Some home video games maintain simple **models** of their relatively small sets of players in order to recall preferences, and perhaps to restore the state of a suspended game...

These simple, familiar examples illustrate the use of the dimensions of authoring; the remainder of this chapter surveys the esoterica of authoring, using the same dimensions.

3.2 Graph Generation

So I sat down and wrote a program that'll take those numbers and do what you like with them. If you just want a bar graph it'll do them as a bar graph, if you want them as a pie chart or scatter graph it'll do them as a pie chart or scatter graph. If you want dancing girls jumping out of the pie chart in order to distract attention from the figures the pie chart actually represents, then the program will do that as well.

—Douglas Adams
Dirk Gently's Holistic Detective Agency

The automatic presentation of information has occupied its share of the AI as

well as graphics³ literature. Early work, in particular, recognized the strong relationships between knowledge representation and (graphical) presentation which characterizes the current work. For instance, Zdybel *et al.* [205] define an Information Presentation System (IPS) as a system that:

1. Automatically generates displays according to content-oriented specifications
2. Provides a systematic basis for interpretation of user graphic input
3. Functions reasonably well without demanding custom-tooling to a particular application
4. Is easily extensible to satisfy domain and user-specific display requirements.

They further state (1) that their “view of an IPS is that it is itself a knowledge-based system,” (2) that a “high degree of sensitivity to the human end-user can be built in,” and (3) that “an IPS is a place to embody a consistent set of decisions about the human factors of graphic display.” The first point is basic to this thesis, that presentations can in some sense be expressed in and even derived with some form of logical calculus. The second point takes aim at the issue, also central to this thesis, of user modelling, and the third refers to psychophysical issues addressed in this chapter under the heading ‘flesh-and-blood,’ and embodied in the implementation described in this dissertation as *perceptual salience*.

In their description of the View System, Friedell *et al.* [79] discuss how “the graphical presentation of data is tailored to the user’s identity, task, and database query.” Pre-defined presentation plans are chosen by a best-first search mechanism. They recognize the potential of encoding knowledge about graphical presentation in the form of what they call ‘synthesis operators’ [80], and also discuss

³In this dissertation, “graphics” means virtually any technique used to produce visual representations of information; this will naturally include business graphics like bar graphs of corporate data, 2D and 3D drawings generated with and without the aid of computers, and visualization techniques for multivariate data.

“reasoning about how to select and combine ...primitive elements of object descriptions.”

Even the possibility of reasoning with multiple media appeared in the literature of the early 1980’s. Neiman [143] writes: “...the use of the knowledge structures to generate multi-modal output demonstrates the generality of the knowledge representation techniques employed.” His system was used to generate explanatory, data-driven animations for users of a CAD system, coordinated with natural language explanations.

The fields of graphics and AI begin, unfortunately, to diverge at about this time,⁴ and the potentially very fruitful area of automatic presentation defined by the intersection of these fields has suffered for it. Nonetheless, a few researchers have continued in this tradition, and their work is reviewed in the remainder of this section.

The work of Mackinlay [129] [130] is the first to explicitly address expressiveness and effectiveness criteria for visual presentations.⁵ He concerns himself with relational data (this is the domain) and restricts himself to graphical languages commonly associated with business graphics (the deliverable). The presentations are designed for traditional screen or plotter devices (target media). Although there is no explicit modelling of users, the system embodies explicit knowledge of various media. There is considerable effort to render presentations in accord with human perceptual abilities and limitations: the system can thus be seen as modelling certain normative human characteristics, but this model is static and acquired from psychophysical studies in the literature (flesh-and-blood). There is no over-arching notion of style by which one presentation might be coordinated with another, except in special cases when a presentation is broken into, say, two line graphs with

⁴This claim is based upon the non-incidence of graphics and presentation papers in important AI conference proceedings after 1983. This regrettable schism appears to be under repair by the mid-nineties.

⁵For basic background on graph representation see Bertin [24] [25] [26] and Cleveland [48] [46] [47], or [57] for a survey.

Deliverable	Business graphics
Task/Data Domain	Relational data
Scope	Not applicable
Target Media	Traditional screen and plotter devices
Flesh and Blood	Implicit psychophysical limitations
Style	Not really
Models	Explicit: media limitations; Implicit: Human perception
Role Relationships	Single author, multiple reader
Heterogeneity	Not applicable
Spaces	Static information and presentation

Table 3.1: **Mackinlay and related work.**

a common axis. The prototype developed by Mackinlay is called *APT*, and has the following typical role-relationships: Single-Author and Multiple-Readers who are not necessarily mutually co-temporal or mutually co-present with the Author. The presentation is static, prepared for a particular data/medium combination; there is therefore no particular support for heterogeneous environments, although the same data may be presented differently in environments which differ in the available media. See Table 3.1 for a summary.

Mackinlay's work is interesting both in its own right, and because it has been the starting point for a variety of efforts by other researchers.

Roth [175] attempts to characterize semantic dependencies in the data to be presented. He describes the static categories of *data types*, *relational-structure*, *arity*, *relations among relations*, and recognizes the need to represent the dynamic needs of users. Because he distinguishes time as a separate data type, he is able to make special arrangements for the display of temporal information.

Casner [38] undertakes to include task descriptions in the process of automating graphical presentations. His notion of expressiveness is at a much higher level

than Mackinlay's, and derives from a logical formulation of the task in which a user is involved. Casner's system embodies generic models of human perception which are consulted to produce presentations in support of particular tasks, and which will minimize the perceptual and cognitive demands placed on users engaged in these tasks. The effectiveness of the corresponding presentations is measured by reaction-time regression studies of users engaged in five tasks in the airline reservation domain.

Marks [132] has addressed the issue of avoiding unwanted conversational implicatures in a coordinated text-graphics environment. One of the novelties he claims is the inclusion of relations that describe the perceptual organization of symbols. Canonical organizational principles he mentions explicitly are sequential layout, proximity grouping, alignment, symmetry, similarity and ordering. Rather than fabricate some convincing *post hoc* argument for the necessity of paying attention to perceptual issues, he properly and soberly points out that "...it is virtually impossible to design meaningful network diagrams for which no perceptual organization will occur."

Henry and Hudson [97] have taken what might be called a semi-automatic approach to the generation of graphs. Their paper describes a system which supports a user in the exploration of large graphs, primarily by allowing iterative, interactive refinement of layout algorithms. Although the authors do not explicitly mention modelling of the user, this paper is interesting in the present context because the observations about layout may be relevant to navigation through abstract information spaces as well.

It may not always be the case that (hypertext) systems need to display to their users a graph of all or part of the underlying database, but it will always be the case in appropriately large systems that only some subset of the information can be presented.⁶ This puts the emphasis upon user-centered means of deciding what

⁶The idea of an "intelligent zoom" [19] changes this characterization in that while the entire

should be presented, and how it should be presented. Approaches like the one described in this thesis might be used to select a critical subset of the links in a hypermedia document, which might then be explored by a user. This critical subset is continuously subject to re-evaluation, in the course of new input from the user, and from other sources of information as may be available to the system.

Noik [150] pursues another approach to automatic graph layout, taking the notion of the fisheye view to multiple focal points in hierarchically nested structures. This leads to the idea of using some technique to determine the focal points from a user model.

Although most reviewers would likely place Feiner's work in the context of visualization or multi-media (as has been done here with some of his work), some of his efforts can be seen to contribute to the field of automatic generation [71].

3.3 Data Analysis and Visualization

Or you can turn your figures into, for instance, a flock of seagulls, and the formation they fly in and the way in which the wings of each gull beat will be determined by the performance of each division of your company. Great for producing animated corporate logos that actually mean something.

But the silliest feature of all was that if you wanted your company accounts represented as a piece of music, it could do that as well. Well, I thought it was silly. The corporate world went bananas over it.

—Douglas Adams

Dirk Gently's Holistic Detective Agency

Some of the visualization work being undertaken today is of interest here for several reasons. Researchers in this field have recognized the importance of the human in the HCI loop, and cooperative systems are being designed to take full space is rendered, only parts of it are rendered legibly; such approaches have the advantage of providing some context for the viewer.

advantage of human perceptual abilities. To automatically achieve the communicative goals of the author in his absence, systems must be able to present information appropriately, taking advantage of human perceptual abilities and avoiding its limitations.

The best known approach to data visualization is the scatterplot [90] [197]. The success of this technique is due to the ability of the early vision system to group points in space based upon proximity and similarity in color, size and shape. Ware and Beatty have shown that up to five dimensions can be effectively mapped to a full color scatterplot display, and suggest ways in which the visual effect can be maximized. Were it not for the need to detect patterns in data of arbitrarily high dimension, efforts might have stopped here.

An increasingly popular approach to enlarging the dimensionality of displays is what has been variously referred to as *iconography*, and *geometric coding*. This approach employs a generalization of the traditional graphic primitive, the *pixel*, into a parameterized icon whose features are mapped to distinct dimensions of the data stream. A famous example which proved more useful as a characterization of the method than representative of its success, is the *Chernoff Face* icon family [43].

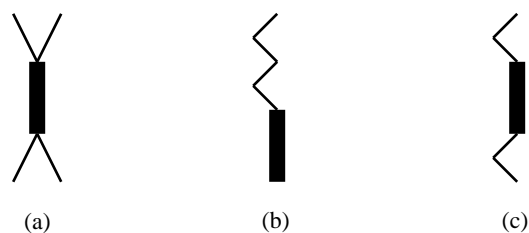


Figure 3.1: **EXVIS five-dimensional stick figures**

The *generalized icon* (gicon) is a generalization of the pixel to higher dimensions. The strategy has been to allow the information in different channels of the

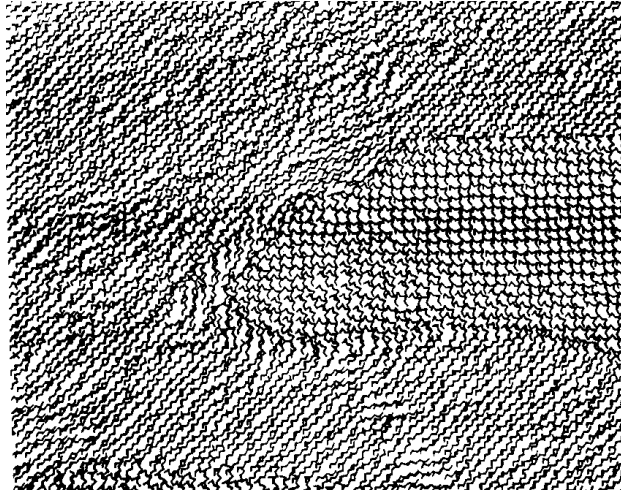


Figure 3.2: **Iconograph**

input data to control corresponding pixels in each gicon. In general, the gicon is an $n \times m$ array of pixels, each mapped to a different input channel. The available display surface is then tiled with these icons. The logic of this and related approaches is that the number of information channels which can be displayed is increased: “Geometric coding allows for further and far reaching extensions [over color] of dimensionality. Observers can utilize shape perceptions to sense the combinations of data at each location and texture perception to sense how those combinations are spatially distributed.” This was the rationale behind the Chernoff Face icon family, as well as the stick figure family described by Pickett and Grinstein [157]. The latter is a stick figure consisting of several connected line segments, where the angle of inclination of each limb is controlled by a different dimension of the numerical data to be visualized. Figure 3.1 presents a representative member of the aforementioned stick figure family, and Figure 3.2 shows how large numbers of them interact to produce global perceptual effects from the underlying data set.⁷

⁷This figure depicts satellite imagery data from the western tip of Lake Ontario; the data was

Deliverable	Graphic
Task/Data Domain	Relational data
Scope	Not applicable
Target Media	Conventional CRT, also audio
Flesh and Blood	Its raison-d' être
Style	Not applicable
Models	Not applicable
Role Relationships	Not applicable
Heterogeneity	Not applicable
Spaces	Static

Table 3.2: **Data Analysis and Visualization.**

The search for effective icons is also lead by studies of pre-attention (see Section 3.4): “Shifts along certain dimensions of color, shape and motion of elements lead to preattentive discrimination, and it is variation in these dimensions that we must seek to bring under data control in our texture displays.”

Some early implementations have been described in the literature. The Exploratory Visualization (Exvis) project [185], for instance:

is a multi-disciplinary effort to develop new paradigms for the exploration of data with very high dimensionality. The fundamental philosophy behind Exvis is that data representation tools should be driven by the perceptual powers of the human. In addition, the interpretation of data of very high dimensionality will be maximized only when we learn how to capitalize simultaneously on multiple domains of human perceptual capabilities.

This project is in the early stages of exploring the possibilities of iconographic data representation using sound attributes, along with the integration of auditory and visual displays into a single unified data exploration facility. (See also Grinstein *et al.* [90] and Pickett [156] and the summary in Table 3.2.)

collected by a U.S. Air Force Geophysics Laboratory weather satellite [90].

3.4 Psychophysical Research

Basic work in psychology has resulted in improving models of human perception. Much of this work (*e.g.*, [144] [191] [190] [193] [192]) has been concerned with elaborating a putative dichotomy between processes which are pre-attentive and those which require attention. Pre-attentive processes are characterized by their speed: they are fast, typically accomplished within 100ms, suggesting that they are performed in parallel by the human perceptual system. Such processes are sometimes referred to as automatic, parallel, or early-vision processes. Although such a dichotomy is conceptually attractive, it has been increasingly unable to account for the data, and new models are appearing which refer to a continuous ranking of perceptual difficulty. Pre-attentive tasks are at the extreme ‘easy’ end of this continuum, while tasks requiring attention are at the other, ‘hard’ end of the scale.

Other researchers (*e.g.*, [137] [66]), while also interested in developing a basic perceptual language, are not so concerned with the underlying psychological model.

Color Considerations Color deserves a separate section in this document for several reasons. Our world is, for most of us, a very colorful place, and the value of color should not be ignored in the computational models we build, of people, and things. Color is, not surprisingly, one of the most effective psychophysical stimulus dimensions. Even in the absence of a complete neurophysiological underpinning, a tremendous amount of informal as well as empirical information is available on the use of color to accomplish various communicative tasks. Not only is the use of color in visualization powerful, but the kind of knowledge we have of color capabilities provokes questions about other human perceptual capacities.

Ware and Beatty show that it is possible for human observers to perceive five data dimensions simultaneously using color [197]. The data they used was

characterized by a hyperellipsoidal probability density distribution, but they conclude with respect to the generality of their results that: “colour is likely to be effective in assisting in the perception of correlations in multidimensional space.” Although in most cases they found that adding color was expressively equivalent to adding three more spatial dimensions, color is not a completely heterogeneous perceptual space, and “resolution is worse in some directions than in others.” In particular, when clusters are separated along dimensions which have been mapped to color, perception suffers. Clusters are perceived as distinct when they are separated by between three and five standard deviations along most of the possible vectors; “much greater cluster separation is necessary before two clusters can be resolved” when they are separated on [only] “a few” specific color vectors.

They observe that users require no training to use their color-based five-dimensional visualization tool, but point out the importance of control over the background color, which tends to emphasize particular colors in the display, and consequently particular correlations in the data (this is a *perceptual implicature*, which systems can attempt to mitigate, or to anticipate and exploit to emphasize data of interest).

Murch also gives an interesting summary of the use of color, from the point of view of a graphics practitioner [140]. He distinguishes between the qualitative and quantitative uses of color, and provides a stimulating list of guidelines for the effective use of color derived from physiological, perceptual and cognitive studies. Murch also provides a list of the sixteen best and worst color combinations. This is the *kind* of knowledge that will need to be consulted by mature automatic presentation systems.

Benbasat undertakes a series of empirical investigations on the impact of color on presentation [23], and the impact of presentation on a variety of manage-

rial decision-making tasks [22] [126]. These investigations are representative of a thread of the literature entirely separate from what has been thus far cited in this paper. The results of Benbasat's work are consistent with those of Mackinlay.

There is a wide range of sometimes contradictory information available on the use of color to represent information in visual displays. Making use of this information is difficult, however, and unless there is a pressing need to convey a large number of categorical variables, a designer is best off with gradations of a single hue to represent changes in the value of a quantitative variable.

Graphical Semantics Montalvo [136] [137] and Grosz [132] are both interested in the *meanings* of graphical displays. Grosz has also done much work in computational linguistics, in a purely textual environment; some ideas have exported well to other media. A more pragmatic approach is detailed by Kurlander and Feiner [120]. These and other similar guidelines are suggestive of a beginning for a database of default axioms for reasoning about presentations of information.

Results of these studies are of direct consequence for designers of human-computer interfaces.

3.5 Multimedia

*Words are too solid
they don't move fast enough
to catch the blur in the brain
that flies by and is gone...*

—Suzanne Vega, *Language*

One day while walking along the road, the monk Gisho met his master, who was blindfolded. Gisho asked, "What have you seen?" and his master replied, "The wind whistling in my ears."

—Zeutuban III

By definition and by convention, a system which makes use of more than one medium⁸ is a multimedia system.⁹

In its most liberal interpretation, the rubric "multimedia" would cover any system which employs the sound of a bell to direct user attention. Typically, though, the term refers to systems with more or less esoteric applications (by today's –or yesterday's– standards) of graphics or video.

Multimedia systems promise great improvements to interfaces, and afford scope for new *types* of interfaces. Fox [78] writes of the possibilities in computer aided instruction:

If computer systems can develop accurate models of user knowledge, including goals and plans as well as facts, multi-media information might then dramatically improve the bandwidth and effectiveness of instruction.

Similar statements can be made for other areas of investigation, and in particular for the general goal of user-tailored run-time presentation. The rest of this section reviews research in multimedia presentation.

Feiner Much of Feiner's work has been directed at identifying and resolving multimedia presentation issues [72] [69] [70], and he has given consideration to the use of models (albeit static ones) of tasks, objects and their interactions, and

⁸Some confusion exists in the literature(s) as to whether it is *medium* or *mode* which is being multiplied, and the terms *mutimodal* and *multimedia* are often conflated. This thesis is not the place to embark upon yet another religious tirade designed to settle the issue; the terms will be used loosely where such usage is harmless.

⁹*mul-ti-me-dia* 'me^d-e^-e (1962) :using, involving, or encompassing several media <a multimedia approach to learning> (Webster's 7th Dictionary, on-line copy).

of models of user knowledge. These models have been brought to bear on the presentation task to determine which objects are to be included in a rendering of a scene, the level of detail with which they should be rendered, and what if any special visual effects are to be employed. For instance, if the task in which a viewer is engaged involves the manipulation of a complex piece of machinery, its functional parts may all be rendered as three dimensional solids; if the task is to disassemble the machine, an exploded view or the use of transparency may be called for to show the inter-relationships of the working parts. Feiner's work has been primarily research-oriented and has not resulted in any commercial products.

Other work by Feiner *et al.* also addresses automated generation of graphical presentations. His work with Peter Karp [105] addresses the automated generation of animated presentations with the use of models of knowledge about filmmaking, and points out the advantages; not only does design-time authoring support for animated presentations decrease their cost, he points out that “automation could ultimately make it possible to generate presentations on the fly that are customized for a particular viewer and situation, adaptively presenting information whose content cannot be fully anticipated.” Feiner does not pursue this goal, but we do in the current work. Explicit models of users do not play a part in the prototype associated with Feiner and Karp's work, but the notion of intent-based presentation is beginning here to take hold. With Seligman [183], Feiner pays particular attention to the communicative intent behind the presentation; this is where the term ‘intent-based’ first appears. The *Intent-Based Illustration System* (IBIS) designs illustrations with a generate-and-test approach using a rule-based system of *methods* and *evaluators*. The former are rules that specify how to accomplish visual effects, while the latter are rules that measure how well a visual effect is accomplished in an illustration. The evaluators can be thought of as psychophysically motivated constraints on the presentations.¹⁰

¹⁰See “perceptual salience” in Section 5.4.1.

Still more work explores the coordination of different media (text and graphics in particular) in single presentations [73].

WIP The WIP project at DFKI under the direction of Wolfgang Wahlster is similar to Feiner's IBIS system in that it too uses underlying static models of objects and tasks to determine the layout of a presentation. The WIP researchers say of their approach that it "should be understood as a starting point which is of practical use for the automatic synthesis of various kinds of pictures...in a multimodal system." [173] These statements are made in the light of the realization that there is not yet a mature theory of graphical communication upon which to build such systems.

WIP's most notable incremental contribution to automatic multimedia generation is the implemented co-dependence of its content-design and format-presentation engines: the execution of these units is temporally interleaved, permitting feedback from later in the presentation pipeline to affect and even retract earlier design decisions. For instance, if the system has decided to show a picture of an espresso machine as part of the instructions for its use, and has begun laying out textual labels of all its functional parts only to find out that one of the labels is too long to fit within the boundaries of the part, it may decide to lay out the other labels in a different way or to draw the entire espresso machine from a different perspective that would support the original labelling method.

The WIP architecture also lends itself to parallelization [195].

ALFresco Stock [187] [186] describes an on-going effort to integrate natural language and hypermedia interfaces. The domain is the exploration of 14th century Italian frescoes, and gives users the means to retrieve multimedia information via natural language queries. Stock argues that this approach takes advantage of multimedia to increase the bandwidth of the communication channel between hu-

man and system, and reduces the lost-in-hyperspace problem.

FRESS, EDS, Intermedia and InterNote Work at Brown University has resulted in several generations of systems, each building upon the strengths of its precursors. Early experience with the Hypertext Editing System in the late 1960's and then FRESS [202],¹¹ and EDS [202] led to the more modern Intermedia [202] and InterNote [39] systems.

FRESS (File Retrieval and Editing System) is a multi-user hypertext system developed in the late 1960's, and EDS (Electronic Document System) is a hypermedia system developed in 1982 for VAX and Ramtek 9400 color display environments. One of the major differences between FRESS and EDS is the addition of maps to help users avoid getting lost. Both systems offered support for bi-directional linking and keyworded links and nodes.

Intermedia is an electronic document system which is not a separate application, but a framework for a collection of tools that allow authors to make links between standard types of documents created with heterogeneous applications. "The material an application creates is *the document*." Intermedia was designed with multi-user interactive educational applications in mind, and thus emphasizes interactive display and annotation facilities. Professors, for instance, "would be authorized to add or change 'canonical' hypertext structure, whereas the students would be authorized only to add links and annotations." [147] Links in Intermedia are uni-directional connections between two arbitrary and application-specific objects [92]. See Table 3.3 for a summary.

Finally, InterNote is an extension of Intermedia designed to better support small collaborative groups involved particularly with document review and revision. A much-cited aspect of the extension involves the ability to transfer data across links using a technique the authors call *warm linking*.¹²

¹¹Commercially reimplemented in the early 1970's by Phillips Corporation [51, p447].

¹²Another advocate of non-standard linking practices is Schnase *et alia*s [180], who urge that

Deliverable	Generalized electronic document
Task/Data Domain	General desktop support
Scope	Version control and maintenance
Target Media	Text, graphics, animation
Flesh and Blood	No explicit perceptual effort
Style	Not applicable
Models	Implicit
Role Relationships	Blurred
Heterogeneity	Between applications in the desktop environment
Spaces	Dynamic information space

Table 3.3: **Intermedia.**

NoteCards [93] is implemented within the Xerox LISP environment, and was designed to support authors, researchers, designers, and “other intellectual laborers” in their daily toil.

An electronic generalization of the familiar paper index card, the *notecard* can contain text, drawings or bitmap images, and –more recently added– video sequences. Cards also have titles, and there can be a variety of card types. Typed, bidirectional *Links* connect notecards into networks. Types are labels chosen by the user to represent the relationship between source and destination card. Link sources are anchored to icons, but destinations are entire cards. *Browsers* are special notecards displaying a structural diagram of a network of notecards; they are created by the system and are thereafter manipulable as any other notecard. They

arbitrary computational methods be integrated into any link structure. This enlarges even further upon even unconventional definitions of authoring; observe that the now popular http protocol for TCP/IP based communication of HTML-coded information permits just this kind of linking via the CGI interface. See <http://hoohoo.ncsa.uiuc.edu/cgi/interface.html>

also permit manipulation of the structure of the information space via direct manipulation of the objects which appear on the browser. *Fileboxes* are special cards used to organize or categorize collections of notecards.

NoteCards also provides a limited search capability to locate all cards matching some user-supplied specification, currently restricted to text-string search on titles. The system is in use at numerous locations within Xerox, as well as at universities and government agencies for document authoring, legal argumentation, development of instructional materials, design of copier parts, and competitive market analysis.

Halasz [93] makes a number of useful distinctions in order to arrive at three dimensions for analysis. In particular, he explains how systems designed to support *browsing* differ from those designed to support *authoring*. As one extreme of a continuum, systems purely for browsing do not permit modification of the information space. Systems which are primarily intended for authoring are likely to be used by a small number of authors to prepare information for a large number of more-or-less casual readers; these systems will have well-developed creation and modification tools, and will support continuous modification of information structure as part of ongoing task activities.

3.5.1 Video Authoring

The spotlight follows her for a moment, maybe picking up some stock footage. Videotape is cheap. You never know when something will be useful, so you might as well videotape, it.

—Neal Stephenson
Snow Crash, page 33.

No matter how apolitical the producer of the work of art may seem, every work has political relevance of some sort.

The following extract from Davenport *et al.* [62] might well serve as a manifesto for this thesis, with only a few changes and additions to reflect the underlying communicative goal of the author of the presentation, a role unaccounted for by Davenport:

In the case of on-line video servers as well as home-movie editing assistants, the machine must respond to the user by selecting the “best” shots, sounds, and text chunks, then orchestrating or sequencing them to emphasize a particular story. The story content should reflect the user’s background and intent.

The incentive to provide presentations which have been particularized to the viewer’s needs and interests is even stronger with video than with traditional media because time is a precious human commodity, and time is what it takes to annotate, and to view video. Traditional authoring paradigms do not support such run-time determination of form and content.

Video is finding increasing use as a transcription medium in many fields because it arguably provides the richest record (the “thickest description” [82]) of the events of interest. Video recording offers high bandwidth, greatly exceeding human note-taking skills and speed; researchers can later review and annotate video at leisure. And, increasingly, video is cheap. These attributes virtually ensure a growing abundance of video material for future on-line presentation systems.

On the other hand, the limitations of the traditional approach to authoring are most obvious when applied in non-traditional media, such as in the video medium. The raw material must first be acquired, which involves filming and possibly digitizing. From this raw source, video authors must assemble cuts into a cohesive presentation. The raw footage can be very voluminous, and the relevant parts of it very sparsely distributed. Ten hours of video taken during a field study for a new graphical user interface (GUI), for instance, may include many instances of

coffee drinking and doughnut eating by the users that may not be relevant to any conceivable presentation. Nevertheless, it takes someone *at least* ten hours¹³ to scrutinize the footage for something useful. The process of identifying these useful events and sequences has been called *annotation*, and a number of systems have been designed to expedite it. (See, for instance, Buxton and Moran [34], Goldman-Segall [83], Harrison and Baecker [95], MacKay and Tatar [127], MacKay and Davenport [128] and Suchman and Trigg [188].)

When the author has finally identified a set of cuts he or she deems to be relevant to an eventual presentation, the traditional notion of authoring requires assembling these into their final presentation order. Although quite adequate for creating rock music videos, this approach suffers from the aforementioned limitation, that such a presentation can not be tailored to the needs of individual viewers.

It is here that video data diverges significantly from text, graphics and even animation. Video data is inherently uninterpreted information in the sense that there currently are no general computational mechanisms for content-searching video data with the syntactic precision of generalized textual search.¹⁴ Even graphics, because there is usually an underlying model or database that can be queried, and animation, which also has a model or database with temporal information added, can be searched for information using existing computer tools. But frames and sequences of frames in video data cannot easily be queried for semantic content except in fairly specialized domains.

At present, the only practical way of accessing a video database is for a human to first *annotate* it so that the annotation can be used to guide the author and the

¹³Empirical investigations to date suggest that human annotation of video material takes an order of magnitude more human time than the duration of the video being annotated; see, for instance, Harrison [95].

¹⁴See, however, Cherfaoui and Bertin [42], who use digital image processing techniques to extract some types of information from video. Recent work by Goldberg and Madrane on automatic extraction of spatio-temporal indices from video at Eurecom Institut, Sophia Antipolis, France, is also noteworthy. Refer to Joly and Cherfaoui [102] for a survey of related approaches.

viewer. Creating this annotation, at least with the current tools, is an inherently linear operation (in terms of the time required to do it) and is a major bottleneck in the authoring of video documents.

It is useful to distinguish between the transcription processes of *logging* at the lexical level, which lends itself to some degree of automation, and *annotation*, a semantic/pragmatic task which will require human intervention for the foreseeable future. A log of a meeting can be acquired automatically, for instance, by the Group Support System (GSS) software used by the participants. This log can be subsequently used to index a video record of the meeting to find instances of user actions at as low as the keystroke level and as high as the level(s) of abstraction embedded into the GSS (*e.g.*, “brain-storming session,” “open discussion,” etc.) [164]. Annotation, on the other hand, is at a higher level of abstraction, defined by the eventual use to which the record of the meeting is to be put.

See Csinger [59] for how the intent-based authoring paradigm influences the annotation process; the remainder of this dissertation focusses on the post-annotation processes of *presentation*.

In the video medium, selecting the intervals of the record to be displayed, as well as the order in which they are to be displayed, are both serious problems.

Previous work in automatic presentation has dealt with some aspects of both of these questions, and has been restricted for the most part to choosing ‘the right way’ to display items based on their syntactic form [130] [175] (see Section 3.2). Semantic qualities of the data to be displayed are seldom considered; Karp [106] is an exception, where he describes a system called ESPLANADE (Expert System for PLANning Animation, Design, and Editing), a knowledge-based animation presentation planner that uses as input a separately supplied script and a set of communicative goals. ESPLANADE creates a presentation plan at the individual frame level, specifying a hierarchy of sequences, scenes and shots.

Presentation and transcription are inextricably intertwined. A presentation sys-

tem can not present what has not been transcribed; the executive can not retrieve all instances of the mention of a competitor company's name unless the minutes of the meeting contain these references, nor can he retrieve everything said by his subordinate, Doug, unless the minutes are appropriately structured. If the meeting involves a GSS, the facilitation function of the GSS can be expected to provide some of the knowledge required for both the presentation and transcription functions.

Davenport *et al.* [62] describe their approach to interactive digital movie making, a domain similar to ours in that they must log and annotate video footage for later retrieval by computer, in the absence of a human editor. Their domain differs in that it permits control over the acquisition of original raw footage. They are also not as interested in modelling the user *per se* as they are in giving the user meaningful interaction affordances to select variants of the movie. As movie-makers, Davenport *et al.* go to some effort to maintain the stylistic consistency of their presentation, an important element with which we have not yet concerned ourselves.

Davenport *et al.* [63] describe “cinematic primitives for multimedia,” a set of dimensions along which video shots are annotated for later reference. Their challenge, they claim, “is to develop robust frameworks for representing story elements to the machine such that they can be retrieved in multiple contexts.” Our goal, though not focussed on interactive film and storytelling, is similar: presentations always have a ‘story to tell,’ even if they are designed automatically. Creators of interactive or multivariant video are interested in preserving “underlying narrative structures” [68, p12], a quality not far removed from what is called *intent* in this thesis.

In the absence of a solution to the general problem, the amount of costly human effort currently involved in annotating and browsing multimedia information will only be multiplied with the growing interest in the new technologies. Chapter 6 includes a description of how our approach to video authoring has been applied;

this medium was chosen because authoring in the video medium is even harder than in conventional media; there is nothing in our approach that prevents it from working across media boundaries.

3.6 Computer-Supported Cooperative Work

Computer Supported Cooperative Work (CSCW) developed over the eighties into a separate field of research in its own right. Work in this area examines the potential use of computational support for individuals engaged in collaborative group work. The field draws on research activity in diverse disciplines including computer science, artificial intelligence, psychology, sociology, organizational theory and anthropology [87, p5].

Not as relevant to the current work, perhaps, as some of the other areas surveyed here, it is included because automatic authoring and presentation systems can be multi-agented, supporting the collaborative activities of multiple authors and readers. The intent-based authoring paradigm as related in this dissertation involves agents in the roles of author and reader(s), communicating in general asynchronously via the author's intent and the user model of the reader. Although this thread will not be followed in this dissertation, future work will need to address it.

A distinction should be drawn between systems which are *cooperative*, and those which support *collaborative* work. Although the literature varies in its usage of these terms, they will be used consistently throughout this document in the following way: Cooperative systems are those which exhibit behavior which is sensitive to the needs of the individual user-agent. Collaborative systems are those which have been designed to support the joint activity of a number of user-agents.

Much of the CSCW research literature has been broken up along the dimensions of support for tasks in which the multiple collaborators operate synchronously or asynchronously, in the same place or remotely, as reflected in the familiar time-

	Same Place	Distributed
Synchronous	blackboard...	telephone...
Asynchronous	bulletin board...	email...

Table 3.4: **CSCW time and space diagram.**

and-space partitioning of Table 3.4. A good deal of this literature has been concerned with face-to-face collaboration. With the continual advance of computational hardware technology, and the increasing bandwidth available for information transfer, there will be less need for collaborators to travel for meetings, and a concomitant decrease in the emphasis on face-to-face meeting support technology. This discussion will therefore focus on geographically distributed systems. Halasz [93] writes:

Hypermedia is a natural medium for supporting collaborative work. Creating annotations, maintaining multiple organizations of a single set of materials, and transferring messages between asynchronous users are the kinds of activities that form the basis of any collaborative effort. These are also activities for which hypermedia systems are ideally suited.

Work in the field of CSCW is so closely associated with hypertext and hypermedia that it is tempting to conflate these issues. Researchers intent upon creating a sense of co-presence, or being higher on “the social awareness scale” [125], or what Bly has called “connectedness” [27] and Laurel “engagement” [123], have jumped at multimedia technologies as part and parcel of the solution, without showing first that such measures are necessary. In fact, users of systems which incorporate video to facilitate face-to-face interaction pay far less attention to the video information than expected [27].¹⁵

¹⁵For further emphases that technology alone is not enough to solve the CSCW problems, see Engelbart’s urgings [65].

Sarin *et al.* [177] advance the notion of a [real-time] conference as an ‘abstract object’ and discuss some conference design issues. They identify the following dimensions: shared versus individual views, access control, concurrency control, getting data in and out (from other applications, or from paper, etc.), constraints on real-time conference design. They describe the ‘virtual terminal approach’ as a way of giving single-user applications the means of serving multi-user conferences: a ‘virtual terminal controller’ is responsible for multiplexing user I/O.¹⁶ Sarin’s work does not get sidetracked with multimedia issues and good advice is to be found throughout the article for practitioners involved with the design and implementation of CSCW systems.

Lee [124] presents a system and an approach designed to support group decision making, focussing upon representation of the task-derived components of decision-making processes. In particular, the *alternatives* are represented explicitly, as are the *goals* to be satisfied, and the *arguments* “evaluating the alternatives with respect to these goals.”

ZOG and KMS: Shared, distributed hypermedia systems designed for collaborative work. KMS [3] [203] was developed over the 1970’s at Carnegie Mellon University, and then commercialized by Knowledge Systems, Inc. A version found application on board the *USS Carl Vinson* for a variety of tasks. KMS was designed “to help organizations manage their knowledge,” and features organization wide support for collaboration in a broad range of areas, including electronic publishing, on-line documentation, project management, software engineering, computer aided instruction, electronic mail and issue analysis. Screen-sized WYSIWYG workspaces called ‘frames’ contain text, graphics and image items, and can be linked to other frames or used to invoke programs. Links are unidirectional, and destinations are entire frames, which are viewed one at a time.

¹⁶This approach is subsumed by the ‘symbiotic interface’ proposed by Booth and Gentleman [29].

There is no mode boundary between navigating and editing, which is to say that there is no system-imposed distinction between reader and writer. These role categories are implicitly conventionalized in the KMS user community, and much of KMS's functionality depends upon convention. For instance, rather than provide a separate system-level representation for annotations, these are ordinary links distinguished only by a character prefix provided by the annotator and conventionally understood by the user community as identifying an annotation. Certain functionality within the system depends upon convention as well; electronic mail service, for instance, depends upon access to shared frames which function as user mailboxes. The designers of KMS have in this way relied upon convention to augment their rather minimal user-interface. See Table 3.5 for a summary.

Deliverable	Document organization
Task/Data Domain	General info. manipulation
Scope	Versioning and maintenance
Target Media	Conventional graphic display
Flesh and Blood	Not explicit
Style	Local convention
Models	Not explicit
Role Relationships	Implicit, conventionalized
Heterogeneity	Seamless
Spaces	Dynamic shared information

Table 3.5: **KMS.**

Time-multiplexing of screen real-estate via quick response time is used in favor of multiple-window 'space multiplexing.'¹⁷ Adhering again to a minimalist principle, KMS does not provide a separate mechanism in support of user naviga-

¹⁷Discussions with Kelly Booth have led me to believe that although increases are still possible in both time and space multiplexing as described above, growth will be more bounded in the space than in the time domains.

tion; there are no ‘overview maps’ or other devices to help orient the user in unfamiliar parts of the information space. Instead, the designers claim to have been vindicated in their belief that fast response –which enables low cost exploration and backtracking– is enough to avoid getting irreparably lost.

KMS is also notable for its approach to access control. The system supports shared access to a single logical but physically distributed database. Since the basic unit of information is a frame and is thus limited to what can fit in a screenful, any large database will be composed of a large number of frames; this number is generally much larger than the number of users of the database. The designers of KMS use this observation to point out that access conflicts will be rare, and that conventions can once again be relied upon to evolve which will serve to circumvent those collisions which might occur.

Access privileges are established on a frame-by-frame basis by the creators of frames. Once again, it is convention, rather than embedded functionality, which governs access to information under KMS.

A guiding force in the design of KMS was the desire to permit paper renditions of KMS documents. This is achieved by hybridizing markup notation into the WYSIWYG frame displays. This, along with the design intention to support individual as well as collaborative work, contributes to the seamlessness of the system.

Shared Workspaces The main thrust of systems under this rubric is the creation of collaborative working environments that promote the illusion of ‘mutual situatedness.’ Although collaborators may be in physically separated environments, they work in a space which is shared, in the sense that they can refer to objects in that space, relying upon mutual awareness of the terms of the referral as well as of the referent (*cf.* deixis). Many strategies have been adopted to this end.

Sara Bly and other Xerox researchers have explored a variety of shared work-

spaces. An interesting experiment was conducted involving the interconnection of the Portland, Oregon and the Palo Alto laboratories with a network of video, audio and computing technologies called *Media Space* [153]. Similar experiments were conducted, connecting the Palo Alto labs with Xerox labs in the United Kingdom.

Other work at Xerox has focussed on shared drawing surfaces [28]. Further experiments [135] explored the three-way sharing of the drawing surface.

TeamWorkStation [99] [100] is a desktop, real-time system advancing a shared workspace in the form of a sharable computer screen for concurrent pointing, writing and drawing, as well as live video and audio links for face-to-face conversation. The creators of TeamWorkStation emphasize its ‘seamless’ operation: since the shared screen is video-generated, the individual participants can continue to use the tools with which they are already familiar, including the still ubiquitous pencil and paper. Although the system is currently implemented on Macintosh computers, this same factor will permit interoperability between heterogeneous computers. The technology is used here to facilitate remote interaction, and every attempt is made to avoid interfering with existing work habits.

Distributed Knowledge Worker (DKW) [96] was developed at the IBM Canada Lab in recognition of the importance of meetings to the operation of businesses today, and with the awareness that face-to-face settings are not always possible for these meetings. Addressing the issues surrounding remote, real-time meetings, DKW is a minimal support system, providing the means with which to harness available bandwidth rather than imposing a structure on the meeting itself: the ‘meeting facilitator’ component of the system manages rather than controls the meeting. DKW supports multiple meetings, in which a shared information space is also a shared presentation space (*i.e.*, WYSIWIS) in the form of whiteboard, multi-user text editor, video, file transfer and text chat functions, not all of which are completely implemented. The set of on-going meetings is a conference. A ‘minutes log’ is automatically augmented with information about meeting starting and

ending times, times of members joining and leaving a meeting, information saved on the whiteboard, and information on files that were transferred.

3.7 Structured Documents: Content from Form

Some document preparation systems (*e.g.*, Scribe [165] and LaTeX [121]) have made the important distinction between content and form, between *what* is presented and *how* it is presented. Such systems make possible the separation of the specification of the contents of a presentation from the specification of how it should look, and though little advantage has been taken thus far of the potential represented by this separation, the ramifications are beginning to be well understood.

This methodological separation of content from form prefigures the further separation of intent advocated as part of this thesis. In the same way that the structured document paradigm makes possible new ways for browsers to interact with the presentation space, the intent-based paradigm enables new ways for readers to interact with the information space.

Efforts continue to build automatic interface and application generators [149] [148], where the form of the interface need not be decided in advance by the application designer. Relational data models likewise separate low-level disk storage issues from higher, user-level interpretations of meaning [49] [41]. And, of course, the evolution of high-level programming languages has led to increasing abstraction; programmers need know less and less of the underlying machine constructs and can focus instead upon concepts defined at the task level. The leading edge of this trend is to be found in the object-oriented programming paradigm, where system behavior emerges from the interaction of self-contained objects created by a programmer to be isomorphic to concepts in the task domain.

Even so, the structured document paradigm [7] strays only little from the conventional, traditional notion of authoring (see Section 1.1) in that it is usually the

author himself who decides the content of the eventual presentation. A brief review of the literature pertaining to structured documents appears in this section.

Reid [166] offers what he calls “observations about systems employing structured documents” [p108]:

- Structured documents contain more information than just the text or graphics itself. That information is usually called “the structure.”
- Most structured documents must be processed or compiled or formatted into some concrete form before they can be printed or displayed.
- The compilation process discards information: the same structured document can be processed into several different concrete documents.

Newcomb [146, p67] offers a definition:

Structured documents are so named because the hierarchical and sequential structure of the various kinds of information they contain is made explicit by identifying tags. Each tag associates a “generic identifier”— the name of the kind of thing being tagged (*e.g.*, “subsection”) — with the data surrounded by a start tag and end tag of the same generic identifier.

The familiar (in academic circles) LaTeX [121] document formatter (built as a macro package for the TeX type-setting system [112]) is in this spirit, as was Scribe [165], its acknowledged predecessor. The “structure” and other information added to create a structured document is sometimes referred to as its “markup” and the dialects used to convey this information have been called “markup languages.” Standard Generalized Markup Language (SGML) is perhaps the best known, with its variant Hypertext Markup Language (HTML), currently in wide use for the authoring of documents on the World Wide Web (WWW) (see Section 3.9).¹⁸ Such

¹⁸SGML (ISO 8879-1986) has been adopted by many of the world’s largest publishers and by many governments, including the US and the EC.

documents generally leave rendering decisions to a run-time browser program such as the popular Netscape and Mosaic products, which are graphical browsers that try to provide a full color high-resolution layout for HTML files, or Lynx, a text-only browser.

Reid [166, p107] nicely sets out the problems in store for intent-based authors when he points out (the emphasis is mine):

There is usually a separation in time or space between the creator of the document and the user of the document. The difficulty in communicating and storing structured documents comes from the need to make sure that the reader will interpret the structure in the way that the writer *intended*.

...

...to make sure that there is enough information included with the document that equivalent bindings can be made by the reader, to produce a printed or displayed unstructured document that properly resembles what the writer *intended*.

But then he goes on to conclude that these problems are unsolvable:

The problem of communicating structured documents is fundamentally unsolvable; it is almost unsolvable by definition. This is because the only way to be certain that the structure will not be misinterpreted, that the binding decisions will be made properly, is to make all of the binding decisions and remove the structure from the document. This reduces the problem to one that is quite solvable, namely the communication of an unstructured document, but also that is not nearly as useful.

Reid appears not to have considered the application of automated intelligence to supply the missing information. The intent-based approach advocated in this thesis is actually twofold: the author supplies an intent as an extension to the structure referred to in the structured document paradigm, and a user model is consulted at run time by an intelligent browser that supplies what Reid thinks of as the missing rendering information.

What Reid takes as a weakness of the structured document paradigm may well be one of its greatest strengths. The late-binding referred to by Reid is what makes possible the user-tailored presentation of the author's intention. That the receiving station, the presentation system, may be ill-equipped to cope with the task at hand is something computational systems will have to deal with; human interlocutors have been misunderstanding each other since before they even developed language, or speech, yet enough effective communication has taken place to permit the development of human civilization.

3.8 Hypertext and Hypermedia

Hypermedia provides one solution to the problem with traditional authoring identified in this thesis, of being unable to tailor presentations to individual users at run-time. In hypermedia systems, the viewer completes the job of the author by selecting and ordering the information to be viewed through the process of navigating at run-time the links established by the author. But this only pushes the problem from one person (the author) to another person (the viewer) and it dramatically increases the demands on the author who must provide the explicit navigation cues. Reducing the amount of human effort required from the author and viewer is still a significant problem with current approaches to video authoring. These effects can be mitigated by the intent-based authoring approach advocated in this dissertation.

A hypertext document or hypertext is a collection of distinct nodes of information connected via a network of links. When nodes contain information of different types, such as graphical, auditory or video sequences, the term *hypermedia* is often applied to the network. Innumerable variations exist on the kinds of links employed, and the way they are activated. The links both impose a structure on the document, and permit run-time determination of the order in which a reader

accesses the information.

Most irksome of major problems with the hypertext approach that will not diminish with mere technological advance is the ‘lost in space’ problem: in such a huge information space, it is easy to get lost, or sidetracked. A promising solution to this problem is the application of user modelling to hypertext systems.

“Hypertext” is a term coined by Nelson in 1965 [145] and covers a wide range of concepts, many of which had their inception in the vision of the *memex* by Vannevar Bush [33, p190].

Bush’s vision went further, and were it not for technological limitations of his time, he would have tried to build hypertext systems very much along the lines we see today.¹⁹

The idea of modelling the user of such systems appears in Bush’s writing, as well as in Alan Kay’s vision of the *Dynabook* [110]. Negroponte’s [142] musings are also easy to interpret in the context of user-modelling:

Imagine a machine that can follow your design methodology and at the same time discern and assimilate your conversational idiosyncrasies. This same machine, after observing your behaviour, could build a predictive model of your conversational performance. Such a machine could then reinforce the dialogue by using the predictive model to respond to you in a manner that is in rhythm with your personal behaviour and conversational idiosyncrasies.

These and other hyper-X concepts appear in many of the systems reviewed in this chapter; some of these were described under multimedia, others under CSCW topic headings. This choice was based in each case upon the original design intention. Thus, FRESS, EDS, Intermedia and InterNote, –which were not explicitly designed to support collaboration– appear under multimedia, while ZOG and

¹⁹See “Memex Revisited,” in *Science is not Enough*, 1967, as well as the original Memex article, “As We may Think,” *The Atlantic Monthly*, vol. 176, July 1945, pages 101-108, reprinted in [87, p17–34] and available at publication time on the World Wide Web at URL <http://www.csi.uottawa.ca/~duchier/misc/vbush/as-we-may-think.html>.

KMS are described under CSCW.²⁰

The rest of this section reviews several hypertext systems conspicuous in the history and the literature of the field.

Xanadu [147, p33] is the embodiment of Nelson's vision of a future in which a single hypertext system provides access to all of the world's literature.²¹ The Xanadu model permits access to any part of any document from any document. Nothing is ever deleted in Xanadu, so that links to specific versions of a document are guaranteed to persist.²² Copyright protection will be supported in commercial versions of the system [51].

NLS was developed by Engelbart in the nineteen-sixties, as the first (computational) hypertext-like system, and was highly successful within the research environment at SRI. It provided the impetus toward interactive computing that drove much of the research for the next two decades [147, p32].

The **Symbolics Document Examiner** is considered the first 'real-world' hypertext system [147, p38]. It serves as the on-line interface to the extensive on-line documentation for the Symbolics system. It is of note to this survey because it is representative of systems which enforce a modal boundary between authors and readers, and it does so in the most direct fashion: Documents are authored with a separate application called Concordia. This strategy is at the extreme opposite end of the spectrum from systems like Intermedia, which do their best to blur these boundaries.

The most notable difference between the features provided to users of Document Examiner, the reading tool and Concordia, the writing tool, is the link mechanism. While readers are limited to uni-directional links, authors in the Symbolics

²⁰Interesting observations on how to make hypermedia systems more user friendly are to be found in [77], [133], and [206].

²¹Parts of the vision are implemented and are available commercially from the Xanadu Operating Company.

²²This feature is relevant to the notion of 'thickness' discussed elsewhere in this document, as well as to issues of literary criticism and deconstructionism.

environment are provided with links that are bi-directional. The designers of the system felt that authors needed to know the possible paths readers might take to arrive at a node in order to provide a useful “rhetoric of arrival” [147, p167]. A user of Document Examiner can not make changes to the underlying hypertext document, though he can save sets of pointers to nodes called *bookmarks*.

The popular **HyperCard** system shipped free by the Apple Corporation with all Macintosh computers permits both authoring and browsing of hypertexts, but distinguishes these uses via a user-level mechanism. The system operates in various modes: browse-only, authoring, and programming [9].

A likely reason for the fertility of the area of hypertext is the syncretic, interdisciplinary backdrop against which the work has taken place: computer scientists, management information specialists, sociologists, anthropologists, artists, educators, mathematicians and other groups have all contributed to the development of concepts and implementations in hypertext.

3.9 Putting it all Together: Cyberspace?

There is much talk today about cyberspace, the semi-mythical electronic environment in which we and our computational surrogates will one day meet to work, play, and think. Networks spring up every day now, and everyone is anxious to be connected. Why all the sudden interest?

Conklin [51, p454] cautions against the easy answer that recently empowering technology has made these visions of the near future more realistic. He suggests that there has been a gradually growing awareness of the potential benefits of hypertext, and that where the computer industry showed no interest twenty years ago in demonstration systems running on state of the art, dedicated hardware, experts today fawn over the potential of a basic home computer connected to the network. Social changes are at the heart of the information revolution, and it will be further

such changes at the individual and collective level that will fuel continued “electronification” of information and practice.

The rapid acceptance of the global network makes it the ideal carrier for the intent-based authoring and presentation paradigm. The World Wide Web combines unified access to the different kinds of information on the Internet, with electronic publishing in Hypertext Markup Language (HTML), a format for which browser client programs exist on all major computer platforms, from laptop to mainframe. It is a new and continually evolving publishing medium that allows reading and writing and interlinking of documents irrespective of topic or geography, but with the overwhelming size of the information space come difficult search and retrieval problems.

Nelson’s vision of a literary hypertext network spanning the globe is rapidly becoming reality, but what form will it take? How will familiar notions carry over from the paper and pencil documentation preparation tradition? The answers to these questions spell out the future for the intent-based authoring paradigm, where the role of the author is completely redefined. Who owns the copyright on a document rendered by a browser based upon a run-time model of the reader, where the only contribution of the “author” is an intent?²³

David Sewell of Rochester University’s English Department writes [184]:

The traditional view of an “author” as a single autonomous agent, the sole intentional creator of a work, is a product of the age of the codex book, when writing was both material and unalterable. But the electronic medium . . . “denies the fixity of the text, and . . . questions the authority of the author . . .”

Curiously, though, electronic communication has tended to hang on tenaciously to the single, identifiable author: on-line journals have conventional tables of contents and author attributions, nearly all e-mail and news-posting systems identify message senders, and on networks like Usenet the elaborate “.sig” or signature appended to one’s postings has become a way of transcending the uniformity of the medium . . .

²³ Authors in the intent-based authoring paradigm can contribute content, but they are *required* only to contribute intent.

Despite the network's potential to allow anonymous collaboration, it is rare for even experimental network art and participatory projects to be anonymous...

Those of us actually taking part in the on-going 'electronification' of the global information space surely recognize the veracity of Sewell's observations. How many of us would be willing under the existing social structures to devote our time and energy to producing anonymous documents that can be borrowed, modified and claimed by others? Lamenting the death of the author [18] was obviously premature.

The message for systems developers of the near future is that support is still required for version control, access control, copyright control. Control is the operative word; if the author is dead, his ghost still wants the rights to his work.

Scenario: Information Gathering

This scenario considers how the implementation described in Chapter 6 functions in the context of the UBC Department of Computer Science Hyperbrochure, an actual application under development in conjunction with this thesis.

Information Gathering

John, a prospective graduate student, starts up Valhalla after signing on as guest. The user model window pops up with the system's a priori hypotheses about John. Since usage of the guest account carries little information beyond the reasonable assumption that the user is not a current member of the department, some default hypotheses are based upon the knowledge that the terminal John is using is located in a faculty office, and that the departmental on-line calendar lists a faculty recruiting seminar that day. These coincidences conspire to produce the false assumption that John is a prospective faculty member.

If John notices by looking at the user model window that Valhalla thinks he is a prospective faculty member, he may correct this false assumption at this point by clicking on the button that represents that he is prospective student. John can interact with the user model window immediately, or he may wait until after pressing the show button and perhaps wondering why the presentation is not meeting his needs as a prospective student. In either case, after correcting the system's misconception, he is presented with a brief introduction to the department by its head, and then with a number of clips designed to motivate and increase his interest in the department. Valhalla makes numerous assumptions here about the interests of students and instantiates these goals with footage about sports facilities on campus, regular social events in the department, and a brief overview of research activities. John's hypothesized age, which is influenced by whether he is a student or faculty member, has an influence upon whether the system assumes he is single or married, which in turn influences content selection. John lets the presentation play to conclusion and logs out.

Mary, a prospective faculty member, signs on at the same terminal, also as guest, and consults Valhalla. This time, the a priori assumptions are more relevant. (Mary is, in fact, the visiting faculty scheduled for that day.) Mary sees the

introduction, and then an overview of each of the laboratories in the department. She replays the clip of the Laboratory for Computational Intelligence (LCI) several times; this usage information is passed on by Valhalla's interface to the reasoner, which infers that Mary is more interested in AI research than other activities in the department (although there could be other explanations). Mary asks for another presentation (either before or after the current one runs to completion) and is then presented with more detailed footage about the LCI, as well as with interviews with key AI researchers in the department. This second presentation is shortened to accommodate Mary's optimal viewing time, as represented in the system's model of her.

Both John's and Mary's presentations include clips about the Vancouver area, because it is considered by many to be very attractive. This kind of information can even be acquired automatically, by noticing, for instance, that out-of-town users tend to linger over scenic shots in the video presentations much more than do locals (who can just look out the window).²⁴ Had they been assumed by Valhalla to be current, rather than prospective members of the department, John and Mary would not have been presented with this extra information. On the other hand, a user accessing Valhalla over the Internet might receive a presentation with even more emphasis on the local geography, on the assumption that they had never before been to Vancouver.

²⁴*I.e.*, the *a priori* probabilities of assumables can be upgraded according to well-known learning algorithms [201].

Chapter 4

Formal Background

This chapter provides background material upon which the contribution of this dissertation is based.

The subject of symbolic logic and default reasoning is introduced first, and then a particular formalism for hypothetical reasoning is pursued in Section 4.1.1, which is the basis from which the formalism of this thesis as well as the prototype implementation are later built.

Decision theory has been advanced as a normative tool for design under uncertainty. It is briefly reviewed in Section 4.2. Decision theoretic approaches involve averaging over some or all models of the world to produce a “compromise” design that maximizes expected utility.

Speech Act Theory is introduced in Section 4.3.

4.1 Symbolic Logic and Default Reasoning

Symbolic logic was intended originally as a language for unambiguously describing mathematical entities, but with recent technological developments has come a growing interest in the use of logical systems for *reasoning* as well as for representation. Given a set of axioms or formulae which are true, a logic is a set of

syntactic rules for deriving new statements from existing statements. Statements that conform to the syntactic rules of the language are called *well formed formulae* or *wffs*.

First-order logic is both a language for expressing knowledge and a means by which further statements can be derived. Assuming that the available knowledge is *complete*, *consistent* and *monotonic*, the derived statements can be regarded as true.

Completeness with respect to a particular domain is the property that all facts needed to solve the problem at hand are present in the system or derivable from those that are. Consistency is the property that all the axioms are true and cannot lead to contradictions. The property of monotonicity holds when the addition of new facts is guaranteed not to lead to contradictions; the size of the knowledge base in terms of the number of statements in it can only grow.

Non-monotonic reasoning systems are those which are designed to solve problems and manipulate representations in which one or more of these properties do not necessarily hold. The inferences made in such systems are said to be *defeasible*, because new information (observations about the world or environment, for example) may invalidate earlier conclusions, which may in turn have to be retracted. Research in the field of non-monotonic reasoning sometimes goes under the rubrics “default reasoning” or “hypothetical reasoning,” and is broadly characterized by the common goal of achieving reasoning behavior in closer correspondence with intuition.

Numerous formal approaches have been developed in support of making inferences in the absence of complete and reliable information (see, for instance, Brewka [32], Konolige [117], Geffner [81], Reiter [167], Etherington [67] and Kautz [109]). The contribution in this thesis is built upon the Theorist formalism developed by Poole [159].

4.1.1 Default–Programming with Theorist

To make the following presentation more precise, the simple hypothetical reasoning framework of Theorist [162] is used.

“Vanilla” Theorist is defined in terms of F , a set of closed formulae, called the “facts”, and H , a set of (possibly open) formulae called “possible hypotheses,” or *assumables*. The following definitions are relevant:

Definition 1 (Scenario) *A scenario is $F \cup D$ where D is a set of ground instances of elements of H such that $F \cup D$ is consistent.*

Definition 2 (Explanation) *If g is a closed formula, an explanation of g is a scenario that implies g . Such a g is referred to here as an explanandum (the plural being, of course, explananda).*

Definition 3 (Extension) *an extension is the set of logical consequences of a maximal (with respect to set inclusion) scenario.*

There’s more than one way to use a hypothetical reasoning formalism. It can be used at least for prediction and for abduction,¹ often in a single domain or problem.

Theorist is particularly of interest because the same formal definition allows for both default and abductive reasoning [159]. It is also implemented; the examples provided in this thesis have been tested on a running version of the program.

These different uses of Theorist can be characterized along two dimensions:

- Status of Explananda, and
- Status of Assumptions

These two dimensions are the rows and columns, respectively, of Table 4.1, which is referred to henceforth as the *Domain-Formulation Grid*, reflecting its intended use as an aid in the formulation of problems and domains for Theorist and other formalisms for hypothetical reasoning.

¹The Encyclopaedia of Philosophy defines abduction as: “C.S. Peirce’s name for the type of reasoning that yields from a given set of facts an explanatory hypothesis for them” [64, p5-57]. The term “abduction” is used throughout this thesis in the formal sense of Csinger and Poole [61], which is consistent with Peirce’s treatment; refer to Section 5.4 for details.

Status of Explananda

The first dimension concerns whether the explanandum is known or not. This distinction corresponds to a choice between the following:

Abduction: The system regards the explanandum (the observation of the world or the design objective) as given, and needs to find an explanation for it. The idea is to find assumptions that imply the goal. We consider all explanations of the goal as possible descriptions of the world.

Prediction: The system does not know if the explanandum is true, and the idea is to determine what can be predicted from the facts (the general knowledge and the observation or design objective).

The issue is whether the explanandum is known to be true or whether it is something that has to be determined. For instance, if a reasoning agent knows (or has as defaults) that $a \rightarrow b$ and that a , the agent can predict b from its knowledge. Another agent who also knows that $a \rightarrow b$, as well as b , might be able to assume in the absence of contradictory evidence that a . The first agent is using prediction, while the second agent is using abduction.

One interesting difference between abduction and prediction is in the relevance of counter-arguments. For instance, when predicting g , it is important to know if $\neg g$ can also be explained. In abduction, however, an explanation of $\neg g$ is irrelevant [159].

Status of Assumptions

Along the other dimension we can distinguish between the two types of tasks:

Design tasks [74] are those in which the system can choose any hypotheses it wants.

For example, a system can choose the components of the design in order to

		Explanandum	
		Known <i>Abduction</i>	Unknown <i>Prediction</i>
Who	Design User		
	Recognition Nature		

Table 4.1: **Domain–Formulation**

fulfill its design objective, or choose utterances to make in order to achieve a discourse goal.

A consistency check is used to rule out impossible designs. All other sets of components that fulfill the goal are possible, and the system can choose the “best design” to suit its goal.

Design can be done abductively to try to hypothesize components in order to imply a design goal [74]. Alternatively, design can be done predictively to derive a design from goals and any hypotheses we care to choose.

Recognition tasks are those in which the underlying reality is unknown, and all we can do is to guess at it based on the observations we make about it. This definition includes diagnosis, scene recognition and plan recognition.

Recognition can also be performed abductively or predictively [160]. In an abductive framework, each explanation is a possible description of the world, while the disjunction of all explanations is the description of the world. In the predictive framework, an appealing strategy is to predict something only if it is explained from the observations even when an adversary chooses the hypotheses [159], which corresponds to membership in all extensions.

This distinction turns on whether the system is free to choose any hypothesis that it wants or whether it must try to “guess” some hypothesis that “nature” or an adversary has already chosen.

For example, agents who know only that there is to be a meeting on the hour, sometime between 09:00 and 12:30, but not at 10:00, are able to make only disjunctive statements about when it will take place; they *recognize* that the meeting is at 09:00 or 11:00 or 12:00. In contrast, the agent organizing (*designing*) the meeting is free to pick any time that is consistent with its own knowledge. The planning agent is free to choose that the meeting will take place at 11:00.

Note that these frameworks are different ways to use the same formal system for different purposes. In order to use the system, we have to choose one way to implement our domain.

In general, there are not enough constraints in a domain to uniquely determine the approach that the reasoning system should take in formalizing its characteristics [160]. The ‘causality’ in the domain does not uniquely constrain its default-reasoning axiomatization.

These choices are succinctly represented by the number of ways of situating the problem into the domain-formulation grid of Table 4.1.²

4.1.2 Summary and Conclusions

Formalisms for hypothetical reasoning can be used abductively or predictively. Theorist is one such formalism.

Finding enough constraints in a domain to uniquely define its default axiomatization is not usually possible. Default implementations can be classified along (at least) two dimensions: the assumption and *explananda* status dimensions, which we have represented as the rows and columns of the domain–formulation grid shown

²The grid merely summarizes some of the different possible uses of the hypothetical reasoning formalism; different problems/domains fall into different boxes, corresponding to different uses of the reasoning system.

in Table 4.1. The domain formulation task can be superficially regarded as one of finding how the domain fits into the grid’s representation framework.

4.2 Decision Making Under Uncertainty

Making decisions in the absence of a complete description of the world is a complicated task. Many arguments have been advanced to support the popular claim that decision theory yields optimal results (see, for instance, Savage [179]).

4.2.1 Bayesian Decision Theory

Various models of decision making under uncertainty have been proposed with the goal of attaining an optimal decision, all of which embody the notion of maximizing expected utility over a probability distribution of states of the world. Decision theory offers a kind of normative standard for decision making under uncertainty, and has been applied to design tasks under uncertainty. Some of this literature (see Cheeseman [40] for a discussion) argues that the best design is the one that results from averaging over all possible models; it will be argued in Section 5.7.1 that classical decision theory is not the right approach for the intent-based authoring paradigm being advanced in this thesis.

One model [179] consists of a set S of possible states of the world, a set O of possible observations, and a set Ω_d of decision alternatives. A conditional probability distribution $P(o|s)$ describes how likely it is to observe o when the state of the world is s , and a prior probability distribution $P(s)$ describes how likely the world is to be in state s . The utility function $\mu(d, s)$ represents the reward to the decision maker for selecting decision $d \in \Omega_d$ when the world is in state $s \in S$. The general problem is to decide on a mapping from O to Ω_d which dictates the action to take for each observation; such a mapping is usually referred to as a *policy*.

The *expected utility* E_δ induced by the policy $\delta : O \rightarrow \Omega_d$ is defined by

$$E_{\delta} = \sum_{s \in S, o \in O} \mu(\delta(o), s) P(o|s) P(s)$$

The *principle of maximizing expected utility* states that a rational decision maker chooses the policy δ' that satisfies

$$E_{\delta'} = \max_{\delta} E_{\delta}$$

where the maximization is over all possible policies. The quantity $\max_{\delta} E_{\delta}$ is called the *optimal expected value* of the decision problem.

4.2.2 Example

A simple example follows. Let $S = \{rain, \neg rain\}$, $O = \{wet, \neg wet\}$. The conditional probability distribution $P(O|S)$ and the prior probability of S are empirically measurable, but assume here that they are as follows:

	wet	dry
rain	0.8	0.2
no rain	0.1	0.9

Because we are in Vancouver, $P(rain) = 0.9$.

Utilities might be as follows:

	rain	no rain
take umbrella	0	-10
don't take umbrella	-100	0

The problem is to decide whether or not to bring the umbrella given the observations. There are four possible policies:

$$\begin{aligned}
\delta_1(wet) &= take, & \delta_1(dry) &= \neg take \\
\delta_2(wet) &= \neg take, & \delta_2(dry) &= take \\
\delta_3(wet) &= take, & \delta_3(dry) &= take \\
\delta_4(wet) &= \neg take, & \delta_4(dry) &= \neg take
\end{aligned} \tag{4.1}$$

$$\begin{aligned}
E_{\delta_1} &= \mu(\delta_1(wet), rain) * P(wet|rain) * P(rain) + \\
&\quad \mu(\delta_1(dry), rain) * P(dry|rain) * P(rain) + \\
&\quad \mu(\delta_1(wet), \neg rain) * P(wet|\neg rain) * P(\neg rain) + \\
&\quad \mu(\delta_1(dry), \neg rain) * P(dry|\neg rain) * P(\neg rain) \\
&= \Leftrightarrow 18.1
\end{aligned} \tag{4.2}$$

$$\begin{aligned}
E_{\delta_2} &= \mu(\delta_2(wet), rain) * P(wet|rain) * P(rain) + \\
&\quad \mu(\delta_2(dry), rain) * P(dry|rain) * P(rain) + \\
&\quad \mu(\delta_2(wet), \neg rain) * P(wet|\neg rain) * P(\neg rain) + \\
&\quad \mu(\delta_2(dry), \neg rain) * P(dry|\neg rain) * P(\neg rain) \\
&= \Leftrightarrow 72.9
\end{aligned} \tag{4.3}$$

Similarly, $E_{\delta_3} = \Leftrightarrow 1.0$ and $E_{\delta_4} = \Leftrightarrow 90$. We would choose policy δ_1 over policy δ_2 because $E_{\delta_1} > E_{\delta_2}$, which corresponds to our intuitions.

Interestingly, the decision maker prefers, under the given utilities and probability distributions, the policy δ_3 of always taking the umbrella regardless of the observation; no observation can improve his outcome. This is because these utilities reflect a strong aversion to getting wet, and only a small nuisance factor for

being unnecessarily encumbered with an umbrella on a sunny day. A different utility function would, of course, yield different decisions.

The lesson here is that it is often difficult to operationalize our intuitions with meaningful utility values.

4.2.3 Decision Analysis

The preceding formalization has considerable representational power, but at considerable computational cost for real-world problems. It can be used to select an optimal sequence of actions (a policy) from many possible sequences (policies). It takes the position that one cannot anticipate future observations, and must therefore decide what to do by averaging over possible future observations. Such power is not always required.

If the observables or some subset thereof were available to the decision maker before a policy needed to be formed, then using the available information could reduce computational requirements.

We can write an expected utility expression that refers not to a policy, but to an action $\delta(o)$ in the presence of observation o :

$$E(\delta(o)) = \sum_s \mu(\delta(o)|s)P(s)$$

Elements of decision theory are used in this dissertation for the sensitivity analysis described in Section 5.5.

4.3 Speech Acts

Speech act theory [182] [14] distinguishes different categories of communicative acts. Searle [181] divided speech acts into five general classes, from which different hierarchies have been developed (see, for instance, Bach and Harnish [15]). Searle identified:

1. Representatives: acts that make a statement about the world, and can be judged to have a truth value (*e.g.*, inform, boast, deny...)
2. Directives: acts that involve influencing another agent's intentions or behavior (*e.g.*, request, beg, suggest, command...)
3. Commissives: acts that commit the speaker to some intention or behavior (*e.g.*, promise...)
4. Expressives: acts that express the speaker's attitude toward something (*e.g.*, apologize...)
5. Declaratives: acts that explicitly involve language as part of their execution (*e.g.*, quit, fire, marry...)

The authorial intentions referred to in this thesis are generally of the first type, but it is not hard to see how a system like the one described in this thesis could be used to encode communicative goals and perform communicative acts from other categories in Searle's hierarchy.

Interesting work has been done to extend speech act theory to other modalities (*e.g.*, Novitz [152] gives convincing arguments for the ability of pictures to accomplish the equivalent of illocutionary acts) and multimodal presentation systems have made use of this work [5] [10] [11].

The notion of communicative intent espoused in this thesis is similar to any one of many alternative formulations of the *speech act*. In particular, the *illocutionary* act corresponds quite well with authorial intent: the understanding that the author is trying to convey. An utterance is said to be successful or *felicitous* when it results in the appropriate, intended *perlocutionary* act, or effect. When the hearer of an utterance is convinced where the speaker's intention was to convince, the utterance is felicitous.

Humans make many assumptions about their interlocutors while communicating with each other, and this assumption-making strategy has mapped well onto some formalisms for default reasoning [56]. When these assumptions are incorrect, they can lead to *perlocutionary failure* and an infelicitous utterance.

Much of the speech act literature refers to attempts at defining a meaningful taxonomy of these acts into requests, statements, indirect, direct, and so on. More recent work in automatic presentation [5] consider the generation of documents as a sequence of speech acts to achieve a more complex communicative goal. Some of these recent studies have also widened the scope of the inquiry to include non-linguistic “utterances” like graphics. Operationalization of Rhetorical Structure Theory by Moore and Paris [138] is another effort to build taxonomies that might prove useful in automatic generation systems [139], and André [6] has extended and adapted RST theory to suit the planning of multimodal presentations.

The approach described in this dissertation is insensitive to the details of a particular theory of speech acts, but is informed by these on-going efforts to structure communicative acts from underlying speech act components; see Section 5.3.

Part III
Contribution

Scenario: Back to School

Steve and Peter are both considering going to grad school, and both of them are using a future, enhanced version of Valhalla, the application which is described in this thesis, to help them make their decision. As they are distinct individuals—and they are very different individuals—the system supports them in different ways. The intent of this scenario is not to imply that users can be characterized to any degree by their .newsrsrc files, or more generally, by their news reading habits, but that even in such data is potentially useful—and potentially abusable—information.

Steve goes back to school

“Now that my coop program is finished, let’s see what grad school has to offer!” Steve relishes the idea of pursuing some of the questions his recent experiences have raised, and wants a new intellectual challenge. He glances at his watch. “Just enough time before that lecture on database theory. Great!” He logs into a machine on the University of British Columbia’s computer network, starts up his favorite web browser, and asks to receive a presentation from Valhalla, the automated on-line departmental hyperbrochure.

Because Steve is signed on to a departmental machine with an account that identifies him as a visitor of Kellogg Booth, a professor in the department, Valhalla reasons that Steve is a visiting researcher in a field related to computer graphics or human-computer interaction, since these are the principal interests of his hosting professor.

Steve watches the departmental introduction by its head, and pays attention to several clips on human-computer interaction before skipping over the next few clips on computer graphics research within the department. Valhalla re-designs the presentation to emphasize human-computer interaction research, but Steve grows restive and clicks on a button that reveals elements of the user model upon which Valhalla is predicating its designs. “Ah,” breathes Steve, “It thinks I’m a visitor.”

Steve is driven to look at the display of the user model because he can’t understand why Valhalla is showing him so much material on graphics and HCI, when in fact his interests lie in applied physics and numerical analysis. He sees from

the model display that Valhalla thinks he is a visiting researcher, and simply clicks on a radio button to inform the system that he is an undergraduate student. The system recalculates the best model and designs a new presentation that includes an overview of sporting and social events on campus.

Peter goes back to school

“I’m tired of my job”, thought Peter; “I hate my boss, my life is going nowhere. Hmmm. I may as well go back to grad school.”

Since it’s the mid-nineties, the most effective way to find up-to-date information on anything is to surf the net, and Peter fires up his favorite browser. Peter’s browser incorporates the latest version of a user-modelling add-on developed at the University of British Columbia, It knows quite a bit about Peter and is able to tailor interaction to suit him. A variety of university names appear on the monitor, but of the thousands of educational institutions offering post-graduate studies, only those appear which offer courses in computer science, and which are also close to either skiing or surfing.

“Let’s see now... Grad school... Montreal: too cold in winter. MIT, nah. Stanford, too many earthquakes. UBC. Yeah, let’s take a look at UBC...” He wishes to himself that he had a coffee right now, but can’t quite muster the energy to go to the kitchen and make one. He has just enough time to stroke the stubble of his week’s growth, before the home page for UBC’s Department of Computer Science fills the display. Peter clicks on a hot-link promising a departmental overview.

This future version of Valhalla tries to negotiate an exchange of user-modelling information with Peter’s net browser program, which refuses because Peter has instructed all of his software agents never to divulge anything. Valhalla then asks for a copy of Peter’s .newsrsrc file.

“No harm in that, I guess,” mumbles Peter, and releases his .newsrsrc with a click. Valhalla correlates the new information with a vast database of .newsrsrc files and derived user models in order to arrive at some reasonable hypotheses about Peter and his interests. Valhalla counts the number of active newsgroups and notices that Peter is up to date on at least fifty groups including alt.fan.monty-python, alt.rec.humor, alt.tv.simpsons, alt.jokes, and talk.politics. Valhalla concludes that Peter has time to spare (Peter’s version of Valhalla has no sense of humor and does

not know how to avail itself of this opportunity for cynicism), and revises its default assumptions about presentation time limits. Valhalla infers from the domain suffix of Peter's network address that he is not currently local to the Vancouver area, and also finds that interest in leisure activities correlates well with the active news-groups in his .newsrsrc file.

There is no perceivable delay before Peter is shown a breath-taking video sequence of professional skiers zooming through fantastic vistas, maps of the ski regions in the Vancouver area, and an introduction to some of its interesting hiking trails. Because Peter lingers over sequences about the bars on campus, Valhalla tries to test some alternative hypotheses that might explain his action at the interface. An overview of the department's beer brewing club prompts Peter to fast-forward over the material, but he rewinds and plays twice in its entirety a walk-through of night-life possibilities. Valhalla infers that Peter wants to party.

"All right! Let's party!" says Peter out loud, actually relishing the prospect of graduate school. The presentation ends with a minute or two about the research facilities of the department, during which Peter begins to doze off.

"Yeah, UBC. That's the one for me. Gosh. I hope my marks are good enough..."

Chapter 5

User Models for Intent-based Authoring

5.1 Overview

Following the brief motivation in Section 5.2, where the argument for Intent-based Authoring is recapped as the solution to problems with traditional authoring techniques, Section 5.3 introduces theoretical elements of the solution and their application.

The approach, using probabilistic recognition and cost-based design, is advanced in Section 5.4; this approach is an extension of the formalism described in Section 4.1.1.

Users need to understand the behavior of the system on their own terms in accordance with what is called in this thesis the scrutability¹ desideratum, which the system supports with a sensitivity metric. Scrutability and the sensitivity metric are both described in Section 5.5.

Section 5.7 considers some alternatives to the approach advanced herein. The

¹scru-ta-ble (adj) [LL scrutabilis searchable, fr. L scrutari to search, investigate, examine – more at SCRUTINY] (1600) :capable of being deciphered: COMPREHENSIBLE (Webster's 7th Dictionary, on-line copy).

use of decision theoretic techniques is considered in Section 5.7.1 for the domain of multimedia presentation design, and it is concluded that these “compromise” designs are inappropriate for interactive presentation environments. Section 5.6 is a detailed example of the operation of Valhalla, and Section 5.8 summarizes this Chapter.

5.2 Motivation

Except for some of the work by Feiner [183], and the WIP system at DFKI [195] (see Section 3.5), the presence of the user/reader has little effect on the form and usually no effect on the content of the presentation produced by the system. This early commitment to form and content is a severe limitation; there is no reason in general to assume that a single document can serve the purposes of multiple readers, and certainly any single document will be sub-optimal for some reader [104].

Authors have grown accustomed to having complete, dictatorial control over the form and content of their document product, and equally accustomed to the complete loss of control ensuing from the production process. Until now, we have not questioned this traditional model of authoring, in which authors as knowledge workers labor over their intellectual child until it is ready, finally to launch it into the world via a printing press or a CD ROM burner; then, like hopeful parents, they wait and watch to see if their book or multimedia product has the desired effect.

This traditional model is so entrenched because, until now, there has been no alternative. Technological advances thus far have merely amplified the strengths and the weaknesses of existing approaches to authoring, and it is only recently that processors have become powerful enough for us to consider applying intelligence, rather than mere horsepower, via computation.

The Intent-based authoring paradigm advanced in this thesis permits authors to send their work out into the world not as a rigid block of content, sealed forever

inside the cover of a book or the shrink-wrap of a CD ROM release, but as a body of knowledge framed by the author's point of view. Paul Saffo of the Institute of the Future calls this intellectual commodity *context*, and recognizes that its value surpasses that of mere content.² Intent-based authoring makes explicit the point of view of the author, so that client-side computational mechanisms can be brought to bear at run-time to design a presentation of the material that conforms to the author's point of view, or *intent*, while meeting the needs of the individual viewer.

5.3 Components of the Theory

Before content can be separated from intent, and the author's compile-time specification task completely decoupled from the user's run-time viewing task, some critical knowledge bases must be developed. Some of these knowledge bases will be extremely labor intensive to create, and the first intent-based presentation has taken years to prepare. The second one will be able to build on the knowledge of the first, the third on the accumulated set of knowledge, and so on. At some point, preparing the n -th intent based presentation will require less work from an author than it would have prepared a traditional document. Similar to the considerations which motivate code re-use in software engineering, this economy of effort will be an important factor in the success of the paradigm.

The two key knowledge-based ingredients in the intent-based authoring theory are 1) representations of the author's compile-time communicative goals, or intent, and 2) the user's run-time information-seeking needs and goals. These are the components which mediate the new, abstract, extended interaction between author and reader. The first is to be supplied (or selected from an existing set) by the author, the second is to be determined automatically at run time by the system and used in turn to determine the content of the presentation. As will be shown

²From his keynote address at the W.R.I.T.E. conference in Vancouver, Canada, 16 June 1995.

throughout this chapter, the author intent and the user model both serve as constraints on the application of presentation schemata, which are rules that encode notions of stylistic coherence, and reliable organizational principles for effective communication. A simple example of a presentation schema is that a presentation should have a beginning, a middle and an end, and that the beginning should introduce, the ending summarize, the material to be found in the middle.

In addition to representations of authorial intent, user model, and the presentation schemata, other knowledge bases are also consulted by the system at run-time to prepare the presentation. These knowledge bases may be contributed by the author, the viewer, an annotator, or a knowledge engineer; in the near future, intelligent agents may scour the Internet for additional knowledge that could be used in unanticipated ways. Each of these knowledge-based components is considered now in some more detail.

Authorial Intent In declaring explicitly his intention, an author can license a presentation system to prepare a presentation that will meet unanticipated run-time contingencies. Even in the absence of the author, a presentation can be made automatically, in accordance with the author's intention, taken by the system to be a specification of the presentation.

An intent is analogous to a speech act (see Section 4.3 for some background). It is a complex communicative goal, arbitrary up to the limits of expressiveness of the language in which it is articulated. The language used here is described in Section 4.1.1 and Section 5.4.

A simple example of an intent is "Show the user a presentation about the Department of Computer Science, but don't bore the user with irrelevant material and try to accommodate the presentation to the amount of time the user has available. Try to impress the user with the department." This intent is broad, and quite general. It might be expressed in the underlying representation language by writing a rule

called **show** and telling the system to apply that rule at run-time; assuming that the knowledge base already contains one or more ways to satisfy the rule **show**, different presentation schemata would compete for dominance until the best presentation is derived for the most likely user model. These processes are described at length in this chapter.

User Models Models [115] [196] of the readers or viewers of presentations are also needed to overcome traditional limitations and move into intent-based authoring. A reader model can be considered to be a representation of the reader's attributes relating to his or her information-seeking needs and objectives in consulting the system. A reader model consists of a set of hypotheses about that reader, acquired at run-time by making observations of the user's interaction at the interface.³ The application of rules from the knowledge bases to derive the eventual presentation, must be consistent in a logical sense with the elements of the model.

The representational range of the models may vary between domains; different domains will require the representation of different kinds of information. A knowledge engineer determines a priori this representation range, by specifying a number of dimensions of analysis along which user attributes can be assigned. The values that these attributes can take are predefined, and are called *assumables*, or potential hypotheses, and they collectively comprise an ontology for possible user models. As developed in later sections of this chapter, a user model in our theory is a set of assumptions drawn from the set of assumables. The assumables establish the representational range of the possible models, and so should be crafted with their eventual usage in mind. For instance, in the system that has been implemented, the domain of discourse is the Department of Computer Science, and potential users fall into the pre-determined classes of Faculty, Student, or Staff, along the dimen-

³Models may also be stored and retrieved where appropriate; these issues are beyond the scope of this thesis, but see Section 7.2 and Section 7.2.2 for some discussion about future work, and privacy issues, respectively.

sion of User Type; they are also evaluated on whether they are visiting or local, and whether they are male or female. These and other sets of assumables affect the design of the eventual presentation as described in this chapter.

A user modelling system may attribute to a user assumptions at different levels of abstraction; while watching a video of a dinner being prepared by a chef, for instance, if the viewer is also *ex hypothesis* a chef, the system might assume that the user believes that chicken marinara is being prepared. If the viewer is a typical North American fast food junkie, the system may attribute the belief that the dish involves chicken and some sort of sauce. How these user models are acquired by the system is discussed in this section.

Acquisition: Models in our approach are acquired both explicitly and implicitly through an abductive reasoning framework described later in detail. The system makes observations of the user's interaction and tries to explain these observations by making hypothetical attributions of user status on one or more relevant dimensions of evaluation. Explicit acquisition takes place when observing the interaction of the user with a description of the user model. Implicit acquisition takes place when observing the interaction of the user with the presentation. The difference lies only in that during explicit acquisition, the user is made aware of the model and is actually called upon to refer to and to manipulate it, while during implicit acquisition, the user need not be aware of the model at all, nor even of the fact that modelling is taking place. Explicit acquisition has the advantage of reliability, but it is intrusive, and can distract the user from the task; implicit acquisition has the advantage of unobtrusiveness, but suffers from potential inaccuracy. A combination of these techniques is used in the system, with a view to having the best of both. See Chapter 2 for a discussion of the difference between implicit and explicit acquisition of user models.⁴

⁴Once acquired, individual models of users can be filed in a database and retrieved later, as necessary, and updated on a continual basis. Although beyond the scope of the work related in this

Other Knowledge A rule base of facts and assumables (as described in Section 4.1.1) is supplied by a knowledge engineer.

Part of the database is world knowledge that the system uses in domain dependent (“Faculty members don’t take courses, typically,” etc.) and independent (“There are 24 hours in a day,” “Men are mortal,” etc.) ways, and there can be knowledge about the characteristics of the media involved (“Don’t play the audio track when video is played at less than half-normal speed,” “Use still-frame techniques when showing a video clip of less than one second duration,” etc.) Categorical, contingent, and hypothetical statements are expressible in the language.

Presentation plans are the bulk of the knowledge encoded. These are variously elaborated rules for delivering information; a convince plan might present examples as evidence in support of a conclusion; stylistic or cultural factors might govern whether the evidence precedes the conclusion (prefix) or comes after it (postfix). A schema to describe something might be implemented as a rule which says: “Describe a Thing by Describing its Parts.” An intent-based author might invoke a schema that has been defined to not offend the viewer; such a schema might consult a database of cultural sensibilities, and tailor the presentation to suit the hypothesized cultural vagaries of the viewer. (In some cultures, for instance, the viewer might be offended by the tone of voice, the style of address, the dress or even the gender of a speaker; the system can choose the appropriate design element at runtime to suit these constraints. Such just-in-time choices were not possible under the traditional model of authoring.)

Presentation schema are instantiated finally with actual content: text, video clips, audio, etc., selected by the application of rules during a proof process described later. In order to select these content elements, the system must be able to reason about the content. For this reason, another important part of the database is devoted to meta-descriptions of the content from which a presentation is to be

thesis, see Section 7.2 and Section 7.2.2 for some discussion.

made. In the case of the system implemented for this thesis, the content is a collection of video clips, a selection of which is assembled by the system at run-time into a presentation tailored to meet the needs of the individual user as represented in the hypothesized user model. To be able to reason about the contents of the video, the system requires a description of its contents, which we call *annotations*. The annotations link keyword descriptions of the contents of specific clips with the associated time codes on the media (video tape, laser disk, digital video streams, etc.) For example, an annotation might carry the information: An interview with Maria Klawe containing overview material is to be found between time codes 1:12:14 and 1:15:01 on the stream called “Department”.

Figure 5.1 is a tree diagram of a presentation that has been designed to convince someone to join the faculty in the Department of Computer Science at UBC. It consists of the application of both the Convince and the Describe schemas just mentioned. The presentation has a *description*, followed by a *conclusion*; the description of the department is instantiated with an *intro* by the department head on video, followed by descriptions of parts of the department (laboratories), which in turn are instantiated by video clips of interviews with representatives from the two labs that were judged by the system to be of most interest to the viewer, and scenes from the labs. The conclusion consists of descriptions judged by the system to be of general appeal; rules have been encoded to assert that it is generally believed that the Vancouver area is very scenic, and that this is a quality that can be used to convince people, so scenic shots of the area are presented. This example is taken from the working version of the system. Later examples in this thesis will show that these scenic overviews are provided only to viewers assumed by the system to be not from the Vancouver area.

Finally, individual pieces of the video record must be chosen to fill the slots in the now elaborated plan schema; the leaf nodes in the tree representing the plan schema are to be expanded. In Figure 5.1, the logical form *Describe(lci)* is instan-

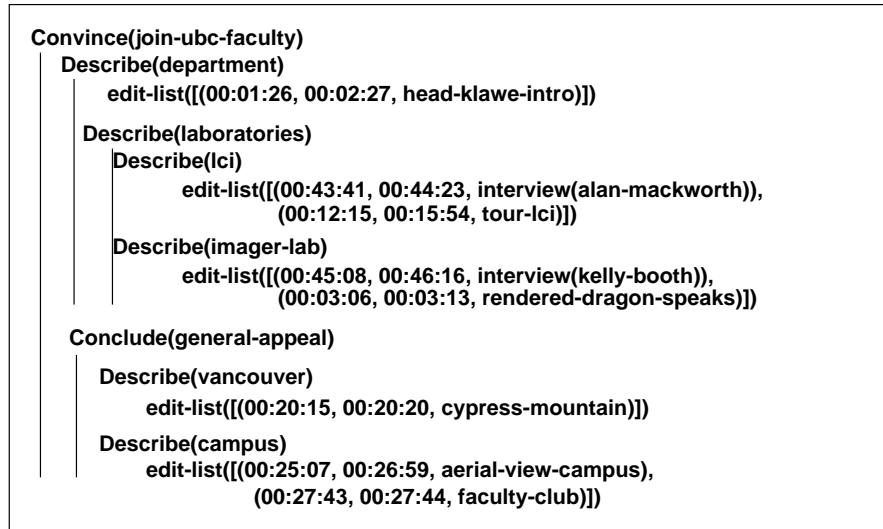


Figure 5.1: A partially elaborated presentation

tiated with the video clips labelled *interview(alan-mackworth)* and *tour-lci*, which are to be found on the video tape between absolute time codes 00:43:41–00:44:23 and 00:12:15–00:15:54, respectively. This example is taken from the working version of the system.

Some part of the database might one day consist of a collection of intentions from which intent-based authors can select, if they don't want to specify their own intentions (or don't know how, because they aren't programmers...). Intent-based authors could compose the intentions from the database into higher level intentions. For instance, if there already exists a representation of an intention to convince as well as a representation of an intention to amuse, an intent-based author might conjoin these to amuse and convince the reader. Only the hint of this future facility is currently available in the system, which still requires that intent-based authors have a strong ability to program their intentions in the underlying repre-

sensation language.⁵

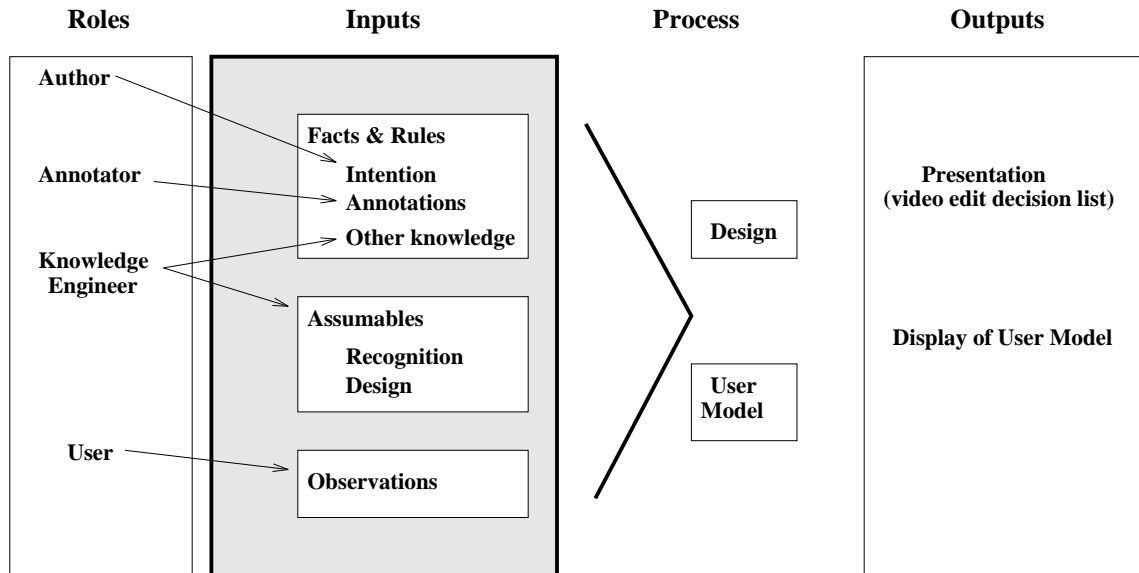


Figure 5.2: **Roles, Inputs and Outputs**

5.3.1 Summary: Inputs, Outputs and Roles

The outputs of the system are (1) an edit decision list of video clips which is played under user control on a video display device, and (2) a presentation to the user of parts of the user model derived by the system. Refer to Figure 5.2: Author(s) supply or select intentional descriptions of their communicative goals. Knowledge engineers provide general and specific knowledge, as well as the assumables for model recognition and presentation design. The system calculates the most likely user model from observations of the user's activity and uses that to design

⁵A visual editor could also be added, that allows authors to compose intentions through the manipulation of a graphical interface.

the “best” presentation. Both the presentation and components of the user model are displayed to the user.

Several agents are involved in the Intent-based video authoring model: They are:

- 1) Annotator: the person or artifact who indexes relevant events and intervals in the raw footage
- 2) Author: the person or artifact who specifies the intent (and optionally elements of the content and form) of the eventual presentation
- 3) Viewer: the eventual consumer of the presentation
- 4) Knowledge Engineer: the person or artifact who prepares the knowledge bases, particularly the domain-dependent knowledge base

Note that more than one of these roles can be played by a single agent.

5.4 An Abductive Framework for Recognition and Design

The language of Theorist introduced in Section 4.1.1 is now extended to include probabilities and costs, and a notion of explanation is presented that reflects a new combination of design and recognition within a single formal framework.

The set \mathcal{H} of assumables is partitioned into the set \mathcal{R} of those available for recognition and the set \mathcal{D} available for design. Each assumable r in \mathcal{R} has associated with it a prior probability $0 \leq P(r) \leq 1$. \mathcal{R} is partitioned into disjoint and covering sets J_i which correspond to independent random variables (as in Poole [161]).⁶ Every assumable d in \mathcal{D} is assigned a positive cost $\mu(d)$.

Representative recognition assumables found in the database include the ones shown in Figure 5.3, where P_n is the prior probability associated with actions of the corresponding classes, $\sum P_i = 1.0$, and the syntax in use is:

⁶The description here of the recognition process conforms to the discussion of probabilistic Horn abduction provided by Poole [161].

```

% For recognition of the user model:
%   faculty/student/staff user
disjoint([userType(faculty):0.35,
         userType(student):0.6,
         userType(staff):0.05]).

%   local/prospective user
disjoint([geo(local):0.65, geo(prospective):0.35]).

```

Figure 5.3: **Representative recognition assumables**

$$\text{disjoint}([assumption_1 : P_1, assumption_2 : P_2, \dots])$$

Some examples of facts and rules that appear in the database are shown in Figure 5.4.

5.4.1 Recognizing User Models

Observations are made of the actions of the user at an interface. The interface agent communicates this information to the reasoner, which tries to explain⁷ the observations, making recognition assumptions along the way. User actions can be of two major types: the user can interact with a control device to manipulate the presentation elements directly (a virtual VCR control panel, for instance, by which the presentation can be replayed, paused, fast-forwarded, etc.), or the user can interact with a representation of the system’s model of the user. When the system makes assumptions to explain its observations of the user’s behavior at the control panel, we call the recognition process “implicit” acquisition; when the system makes as-

⁷*Explain* is used here in its technical sense, as described in Section 4.1.1.

```

% Categorical world knowledge:
gender(george_phillips, male). % George is a male
gender(maria_klawe, female). % Maria is a female

partof(cpu, computer). % a cpu is part of a computer

% Annotations:
% Maria speaks from 1:26 to 2:26 on the video record:
interval(00:01:26, 00:02:26, speaker(maria_klawe) ,[]).

% George is interviewed between 30:57 and 31:19
interval(00:30:57, 00:31:19, interview(george_phillips) ,[]).

% There is a video-only (no-relevant audio track) aerial view of
%   UBC between 25:07 and 26:59
interval(00:25:07, 00:26:59, video_only(ubc_aerial_view) ,[]).

% schemata
% A video clip (or clips) can be a description of a Thing
describe(Thing, Description) <=
    editList([],Description,description(Thing),0,_L).

% a BigThing can be described by describing its parts
describe(BigThing, Description) <=
    bagof(Thing, subsumption(Thing, BigThing), Things),
    desc(Things, [], Description, 0, _Length).

desc([], Description, Description, Length, Length) <= true.
desc([H|T], InD, Description, InL, Length) <=
    editList(InD, OutD, description(H), InL, OutLength),
    desc(T, OutD, Description, OutLength, Length).

```

Figure 5.4: **Example facts and rules**

sumptions to explain its observations of the user's behavior at the user model window, we call the recognition process "explicit."⁸ Typically, the user model window does not contain the entire user model (as there may be very many assumptions in the user model), but only a salient subset of it, determined by sensitivity analysis (see Section 5.5). We call this subset the *salient model*.

Example: Perceptual Salience

Here is an example that involves explicit acquisition of parts of a user model.

When the salient model is displayed to the user, the user's action at the interface may lead to observations which in turn lead the system to calculate a new and different model. For instance, if the user informs the system that the user is a faculty member rather than a student, as had been assumed, the system may reliably retract its initial assumption and assume the corrected value offered by the user (if the user is now telling the truth). Perhaps more interestingly, even the user's *inaction* may result in changes.

For instance, if the system displays to the user the assumption that the user is a faculty member, and the user *does not* critique the assumption, it makes sense for the system to re-evaluate the likelihood of the model that includes the assumption in question, ostensibly to arrive at a higher value for it, under the additional assumptions that the display of the user model has been seen and understood by the user. These additional assumptions are reasonable if: the window in which the assumption is displayed is not obscured, the text in the window is clearly rendered and is large enough to be easily read, the user is not distracted by other events on the desktop, the user is not distracted by other events in the environment (babies crying, cars colliding, etc.), and so on.

The case where a user critiques assumption *A* but does not critique assumption *B* is of particular interest. The likelihood of the model can be increased on the basis

⁸The current version of the prototype implements only the explicit approach.

of the user's direct action as discussed in the previous paragraph, but should also be increased because of the user's *inaction* with respect to the display of assumption *B*, on the grounds that the user *was* actually attending to the appropriate window (because the user *did* critique assumption *A*).

Although the current prototype merely orders the displayed assumptions according to a calculated sensitivity value, various other display strategies can be used to draw the user's attention to one or more of the displayed assumptions, thereby licensing the additional assumption that the user has attended to the window in which the user model is being displayed. A particular assumption can be highlighted (with color, font, or animation, for instance) to increase the likelihood that the user is attending to the relevant portion of the desktop, and to the relevant part of the window. Such perceptually salient display techniques can increase the reliability of the system's assumptions about the user, but require that the system model such things as the media capabilities of the user's display and interaction devices. The system would also benefit from knowing whether the window in question is partially or totally obscured by other desktop objects, which would require some degree of integration between the system and the operating system or window manager. A deep analysis of such desiderata for future operating systems remains to be conducted, but are mentioned here because they are compatible with the interaction paradigm that is described in this thesis, and can be represented in the reasoning framework. Empirical investigations are needed to decide which are the most effective display techniques.

Another way to look at the issue of perceptual salience is to note that the system is engaged in a dialog with the user, and that the presentation of the user model window is a communicative act by the system. The perlocutionary effect [182] of these acts is knowledge, on the part of the user, of the system's user model. The intuition is that the likelihood of achieving this communicative goal is enhanced by using appropriate presentation techniques to highlight the most important parts of

the message, and that the use of such techniques licenses an increased commitment by the system to the beliefs which follow from successful communication.

The user's action at the interface is just another observation available to the reasoning system, and the state of the display is known as a fact, since it is under control of the system. The system explains these observations by (hypothetically) attributing to the user membership in one of the eight interaction classes.

The system calculates the model incrementally, yielding a final model with a cumulative probability influenced by the following rules and assumables; there is one rule for each of the eight interaction classes (see Figure 5.5), of the form:

$$\begin{aligned} action(Var, Action) \Leftarrow & display(Var, Display) \wedge \\ & value(Var, Value) \wedge \\ & class(Var) \end{aligned}$$

where *Var* is the name of the random variable under consideration, *Action* is the actual action taken by the user, and *Display* is the description of the display. *class(Var)* is a conjunct that can be satisfied only via assumption, forcing the cumulative probability of the current model to be multiplied by the prior probability of membership in the interaction class. Other conjuncts are included only to ensure that the eight rules are exclusive.

Users can *explicitly validate* the contents of the display, with respect to a particular random variable; a faculty member can do this, for instance, with respect to the "user-type" random variable, by clicking on the already highlighted "faculty" radio button (see Figure 5.6).

Users can *tacitly validate* the contents of the display by performing some other action, such as requesting that the video presentation be made without further delay.

Users can *implicitly validate* the display with respect to one random variable,

```

% Rules for perceptual salience calculations

% action(R, V) is true when user takes action V
%   with respect to random variable R.
% radio(R, V) is true when the displayed value of random
%   variable R (via a graphical radio button, e.g.,) is V.
% val(R, V) is true when the value of random variable R is V.

% rules when displayed value is correct
action(R, V) <=      % rule for explicit validation
    radio(R, V), val(R, V), ev(R).
action(R, A) <=      % rule for explicit lie
    radio(R, V), val(R, V), el(R),
    A \== V.          % this is a lie
action(R, none) <=   % rule for tacit validation
    radio(R, V), val(R, V), tv(R).
action(R, other) <=  % rule for implicit validation
    radio(R, V), val(R, V), iv(R).

% rules when displayed value is not correct
action(R, V) <=      % rule for explicit correction
    radio(R, D), val(R, V), ec(R),
    D \== V.
action(R, A) <=      % rule for weird lie
    radio(R, D), val(R, V), wl(R),
    D \== V, V \== A.
action(R, none) <=   % rule for tacit lie
    radio(R, D), val(R, V), tl(R),
    D \== V.
action(R, other) <=  % rule for implicit lie
    radio(R, D), val(R, V), il(R),
    D \== V.

```

Figure 5.5: **Rules for perceptual salience**

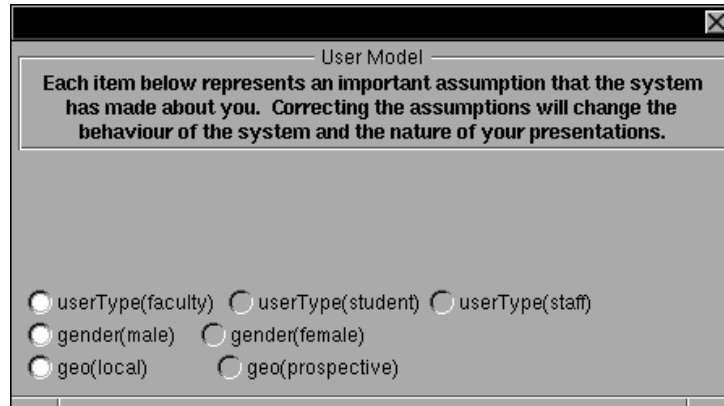


Figure 5.6: **The Valhalla User Model Window**

by performing an action at the interface with respect to some other random variable; a faculty member might implicitly validate the assumption of the system that the user is a faculty member (represented to the user by the highlighted “faculty” radio button), for instance, by clicking on a button that expresses gender or age, or geography.

Users can be *explicitly lying*, by clicking, for instance, on the “student” button when they are in fact faculty members.

A *tacit lie* is when the user takes no action to correct a false assumption by the system. (The user could obviously be missing the cues in the display by accident; the names attached to the eight interaction classes are purely syntactic, and do not necessarily indicate an intent on the part of the user to deceive the system. The names merely make it easier to remember the different categories.)

An *implicit lie* is when the user takes some action with respect to some random variable, but ignores an incorrect assumption by the system.

Explicit correction is the normative action taken by a user when he or she discovers an incorrect assumption by the system; thus the faculty member may click

```

% assumables for perceptual salience calculations
disjoint([ev(R):0.15,          % explicit validation
          tv(R):0.4,          % tacit validation
          iv(R):0.4,          % implicit validation
          el(R):0.05]).      % explicit lie

disjoint([tl(R):0.05,         % tacit lie
          il(R):0.1,         % implicit lie
          ec(R):0.8,         % explicit correction
          wl(R):0.05]).      % weird lie

```

Figure 5.7: **Assumables for perceptual salience calculations**

on the “faculty” radio button when the system has incorrectly assumed that the user is a student. The normativity of this action is represented by the high prior probability associated with it.

Finally, the *weird lie* category captures the action with respect to a particular random variable, where the system has made an incorrect assumption, the user takes action to change this value, but changes it not to the correct value but to another incorrect value. The faculty member, for instance, upon perceiving that the system has made the incorrect assumption that the user is a student, might then click on the “staff” radio button to indicate to the system that he or she is a staff member rather than a faculty member or a student.

The code fragment in Figure 5.7 shows the assumables which represent the eight possible classes to which user interaction can belong. The syntax in use again here is $disjoint([class_1(Var) : P_1, class_2(Var) : P_2, \dots])$, where P_n is the prior probability associated with actions of the corresponding classes.

Some of the more cumbersome detail and irrelevant syntactic sugar has been omitted from the preceding, but the point should be clear. When the user changes a value displayed by the system, the assumption by the system that the user’s action

is an explicit correction, for instance, is preferred over the assumption that the user is explicitly lying to the system; the calculation of the final user model is influenced (as usual) by these incremental assumptions. Perceptual salience is a potentially strong source of knowledge upon which to predicate further reasoning, and will be the subject of much future work. *Any* system that makes presentations to users can benefit from modelling these interactions at some level of abstraction, even at a very basic level.

The probabilities, of course, should be determined through some empirical approach; the values used here are plausible, and serve to order models in which the user has manipulated the user model window in some way. Actual values in a future system would be determined by observing the behavior of users during testing sessions designed for that purpose.

As an example of perceptual salience in the system, let the facts F consist of the rules shown in courier font in this chapter, and let the set of potential hypotheses H (the assumables) consist of the disjoint sets shown throughout this thesis. The initial, “empty” set of observations stems from the intent that underlies the presentation; in this case, the “show” predicate (see Figure 5.8) encapsulates the intent, and requires that certain assumptions be made. This is equivalent to saying that the initial set of observations Obs_I is $\{\exists_{UT}userType(UT) \wedge \exists_G gender(G) \wedge \exists_W geo(W)\}$.

Assume now that 1) the system has arrived at the model consisting of the assumptions that the user is a student, male, and local to the department, (*i.e.*, that $UT = student, G = male, W = local$) 2) the system has represented this model to the user via the user model window, and 3) the user has explicitly corrected the model by clicking on the Faculty button. The new observations Obs_N to be explained are: $\{action(userType, faculty), action(geo, other), action(geo, other)\}$.

Referring to the rules, the observations Obs now to be explained are: $Obs = Obs_I \cup Obs_N$.

The probability of the model before the user takes action is $P_1 = P_{student} * P_{male} * P_{local}$. After the user takes action, the following possibilities exist with respect to the user type variable: The user is a student, and is explicitly lying about being a faculty member; the user is a faculty member, and has explicitly corrected the system's error; the user is a staff member and is lying weirdly about being a faculty member.

With respect to the locality variable, the following conditions exist: The user is local, and has implicitly validated the system's assumption; the user is remote, and has implicitly lied by ignoring the system's false assumption.

With respect to the gender variable, the following conditions exist: The user is male, and has implicitly validated the system's assumption; the user is female, and has implicitly lied by ignoring the system's false assumption.

In the absence of other information or observations, the most likely model is the one in which the user is a faculty member, and has explicitly corrected the system's misattribution (and implicitly validated the system's locality and gender assumptions). The probability of this model is $P_{faculty} * P_{local} * P_{male} * 0.8 * 0.4 * 0.4$. Here are all the possible explanations (user models) of the observations *Obs*:⁹

```
Recognition assumptions with Probability 0.024752 are:
  iv(gender) gender(male)
  iv(geo) geo(local)
  ec(userType) userType(faculty)
```

```
Recognition assumptions with Probability 0.003332 are:
  iv(gender) gender(male)
  il(geo) geo(prospective)
  ec(userType) userType(faculty)
```

```
Recognition assumptions with Probability 0.001092 are:
  il(gender) gender(female)
```

⁹The width of the probability band was set to 0.000005 (see Section 5.4.2).

```
iv(geo) geo(local)
ec(userType) userType(faculty)
```

Recognition assumptions with Probability 0.000572 are:

```
iv(gender) gender(male)
iv(geo) geo(local)
el(userType) userType(student)
```

Recognition assumptions with Probability 0.000147 are:

```
il(gender) gender(female)
il(geo) geo(prospective)
ec(userType) userType(faculty)
```

Recognition assumptions with Probability 0.000117 are:

```
il(gender) gender(female)
iv(geo) geo(local)
el(userType) userType(student)
```

Recognition assumptions with Probability 0.000077 are:

```
iv(gender) gender(male)
il(geo) geo(prospective)
el(userType) userType(student)
```

Recognition assumptions with Probability 0.000022 are:

```
iv(gender) gender(male)
iv(geo) geo(local)
wl(userType) userType(staff)
```

Recognition assumptions with Probability 0.000016 are:

```
il(gender) gender(female)
il(geo) geo(prospective)
el(userType) userType(student)
```

Recognition assumptions with Probability 0.000016 are:

```
il(gender) gender(female)
iv(geo) geo(local)
wl(userType)
userType(staff)
```

Recognition assumptions with Probability 0.000003 are:

```
iv(gender) gender(male)
il(geo) geo(prospective)
wl(userType) userType(staff)
```

Recognition assumptions with Probability 0.000002 are:

```
il(gender) gender(female)
il(geo) geo(prospective)
wl(userType) userType(staff)
```

If other evidence suggests that the user is, for instance, a student (*e.g.*, the user’s id is known already to the system to belong to a student), the system may end up calculating a new, “best” model which includes a weird lie or tacit lie, etc.

Note that the system reports the probability of a model M as a non-normalized prior $P(M)$; this is adequate for the system’s purposes because the value is used only to order models. In order to obtain a normalized posterior $P(M|Obs)$, the system would have to calculate all models which explain the observations, to arrive at the sum $P(obs)$ as seen in Equation 5.1.

$$P(M|Obs) = \frac{P(M \wedge Obs)}{P(Obs)} = \frac{P(M)}{P(Obs)} = \frac{P(M)}{\sum_i P(M_i)} \quad (5.1)$$

This would place an impossible burden on the reasoner, which is currently asked to calculate only the best model, and not all models. Nonetheless, for purposes of illustration, the normalized posteriors for the current example are provided in Table 5.1; the value of $P(Obs)$ is 0.030148. The best explanation is seen from this table to account for over 80% of the probability mass.

User Model: formal definition

Formally, a model is defined as follows:

Definition 4 — Model: A model of the user is an explanation R consisting only of recognition assumptions which explain observations

$P(M)$	$P(M Obs)$	$\Leftrightarrow \log(P(M Obs))$
0.024752	0.821016	0.197212
0.003332	0.110521	2.202546
0.001092	0.036221	3.318108
0.000572	0.018973	3.964735

Table 5.1: **Priors and normalized posteriors.**

Obs about the user: $R \subset \mathcal{R}$, $\mathcal{F} \cup R \not\models \emptyset$, $\mathcal{F} \cup R \models Obs$. The probability of a user model is the product of the probabilities of its elements: $P(R) = \prod_{r \in R} P(r)$, where we have assumed independence of recognition partitions [161]. The ‘best’ model is the one with the highest probability.

5.4.2 Designing Presentations

A single abductive reasoning engine is employed for both recognition of the user model, and for design of the presentation. Design and recognition are interleaved, in the sense that the rule being applied by the reasoner could call at any point for the assumption of either a design or a recognition assumable; a partial model and a partial design are accumulated until either the proof is complete, or it fails.

Various design decisions are made by the system in the course of reasoning. Just as models are defined by their constituent recognition assumptions and formally explain observations about the user, “designs” are defined by their constituent design assumptions, drawn from the set of design assumables, and formally explain the authorial intent.

An example is the use of design assumptions to induce a preference by the system for multiple topics rather than a single presentation topic. This preference can be induced by forcing the system to make an assumption whose cost depends upon

the relationship between elements in the presentation. Specifically, the cost of the assumption is greater when the two elements are selected in support of a single topic, than when they support multiple topics; the system currently values diversity over emphasis, because of the domain in which it is being used: *Valhalla* tries to provide interesting overviews of the department rather than in-depth detail. In conjunction with the `show` intent described earlier and shown in its entirety in Figure 5.8, the following design assumables have the desired effect.

```
disjoint([different_topic_cost:0,
         same_topic_cost:100]).
```

Another example is how the system arbitrates between showing or not showing scenic shots of Vancouver to users who are local or remote. Here is how a preference for showing scenic views to prospective (remote) department members can be encoded. The following rule licenses the showing of a scenic clip for the “right” reasons (*i.e.*, the user is thought by the system to be *prospective*):

```
scenic_clip(prospective, Pin, Pout, LenIn, LenOut) <=
  editList(Pin, Pout, mountains, LenIn, LenOut),
  Pin \== Pout.    % make sure mountain clips are non-nil
```

Pin is the presentation thus far designed, *Pout* contains the new mountain clip. The *editList* predicate makes relevance assumptions while trying to instantiate a clip (or sequence of clips) about the requested subject, in this case *mountains*. *LenIn* and *LenOut* are, respectively, the length of the presentation before and after the addition of the scenic clip.

The following rule licenses the omission of the scenic clip for the “right” reasons (*i.e.*, the user is thought by the system not to be *prospective*):

```
scenic_clip(Geo, P, P, L, L) <= % no mountains here
    Geo \== prospective.
```

These two rules capture what are in some sense the “right” actions for the system to take; it is also possible for the system to take other design actions, but the “right” ones should be preferred. This is accomplished by forcing the system to make (relatively costly) design assumptions in order to take these alternative branches of the search tree.

The following rule models the cost of showing a scenic clip for the “wrong” reasons. It forces the assumption of *bigcostassumption*, at a cost of 200; this is the “nuisance factor” or cost that the system attributes to local users who are shown scenic views they could get by just looking out their windows:

```
scenic_clip(Geo, Pin, Pout, LenIn, LenOut) <=
    % extra cost of mountain clip to !prospective
    boredByView(200),
    Geo \== prospective,
    editList(Pin, Pout, mountains, LenIn, LenOut),
    Pin \== Pout. % make sure mountain clips are non-nil
```

Similarly, the next code fragment models the cost of omitting the clip for users who are not local, and who might have benefitted from seeing some nice scenery.

```
scenic_clip(prospective, P, P, L, L) <=
    % extra cost of no mountains for prospective types
    boredbyNoView(150).
```

The mechanics for inducing the appropriate design costs are included in the last code fragment, only for completeness:

```
disjoint([boredbyView(X):X, boredbyNoView(X):X]).
```

It would not be difficult to generate such rules automatically from a table of preferences, such as the one shown in Table 5.2.

	Scenic	\neg Scenic
geo(local)	200	0
geo(prospective)	0	150

Table 5.2: **Myopic Tradeoff Table**

We call Table 5.2 the *Myopic Tradeoff Table*. It captures the formulation of preference alluded to in this section; the term *myopic* is used to emphasize that the system does not have a complete table of utilities, but uses this approach as a “greedy” discriminator between competing design elements. Such myopic strategies often work well in practice [176, p490]. The table can be constructed for disjoint sets of any cardinality, and for any number of design components; here we present only the myopic tradeoff table for the disjoint set

$$\{\langle geo(local), p \rangle, \langle geo(prospective), 1 \Leftrightarrow p \rangle\}$$

and the presentation element *Scenic*. The cost of showing a scenic clip to a local user (who can look out the window at the mountains any time) is 200; the cost of failing to show such a view to a remote user is 150; there is no cost associated with showing the scenic view to a remote user, or with omitting the scenic view for local users.

These costs influence the final proof in the usual sense that lower cost designs are preferred to higher. The overall weight of an assumption, as always, is proportional to its magnitude; if a dimension is very important in the current domain, the cost magnitudes of its assumptions should be chosen to be greater than the costs of assumptions on other dimensions. This suggests that the costs of design assumptions should not be chosen in isolation, but according to some ranking of relative importance. Such a ranking is not likely to be known *a priori* but will more likely emerge from iterative refinement of the knowledge base, as has been the case with the implementation under review here.

Search Strategy

The prototype referred to in this thesis employs a Prolog meta-interpreter which implements an iterative deepening search strategy wherein first the probability bound on models and then the cost bound on designs is adjusted to yield solutions in desired probability and cost ranges, or *bands*. The width of these bands can be adjusted to yield desired precision, trading execution time for precision.

In classical iterative deepening search [172], the tree is searched depth-first one level deeper on each iteration. The approach combines the space utilization advantages of depth-first search with the characteristic of breadth-first search that, if there is a solution, it will be found; in addition, because shorter paths are searched before longer ones, if there are multiple solutions they will be found in ascending order.

The nodes in the search tree of the application are connected by arcs which can be labelled with the probability factor or cost increment incurred in taking that branch. The cumulative cost of a partial proof is measured in terms of the probability of the partial user model M_p and the cost of the partial design D_p . The pair $\langle P(M_p), C(D_p) \rangle$ characterizes the partial proof currently being evaluated by the system: $P(M_p)$ is given by the product of all probability terms on the path through the search tree to the current node (*i.e.*, the product of the probabilities of all recognition assumptions made thus far), $C(D_p)$ is given by the sum of all cost terms on the path through the search tree to the current node (*i.e.*, the sum of the costs of all design assumptions made thus far).

On the first iteration, the system searches for low cost designs and models with high probability; this is accomplished by setting the probability band to be between 1.0 and $(1.0 \Leftrightarrow p)$, where p is the width of the probability band, and setting the cost band to be between 0 and c , where c is the width of the cost band. Any partial proofs being considered whose cumulative probabilities drop below the probability band are discarded (they fail in the Prolog sense). Similarly, any partial proofs

being considered whose cumulative costs exceed the cost band are discarded. Any successful proofs (*i.e.*, those where $1.0 \geq P(M) \geq p$ and $0 \leq C(D) \leq c$) are reported in the order in which they are found. When all possible proofs within both bands have been considered, the cost band is advanced and the search mechanism re-engaged so that only proofs whose cost meets the condition $c \leq C(D) \leq 2c$ are reported. This process is continued until all proofs have been considered, whereupon the probability band is advanced and the cost band is re-initialized so that only proofs whose probability and cost meet the condition $(1 \Leftrightarrow p \geq P(M) \geq 1 \Leftrightarrow 2p) \wedge (0 \leq C(D) \leq c)$ are reported. The control structure is essentially a nested loop, where the probability band is decremented and the cost band is incremented as the outer and the inner loops, respectively, as in the following algorithm:

```
repeat
  pmax = 1 ; pmin = pmax - deltaP ;
  repeat
    cmin = 0 ; cmax = deltaC ;
    repeat
      iterative-deepening-search(pmin, pmax, cmin, cmax,
                                pfail, cfail, fail,
                                Model, Design,
                                intent(Presentation)) ;
      report(Model, Design, Presentation) ;
    until cfail
    cmin = cmax ; cmax = cmax + deltaC ;
  until pfail
  pmax = pmin ; pmin = pmin - deltaP ;
  cmin = 0 ; cmax = deltaC ;
until fail
```

This control structure determines the order in which the solutions are found, and can be tuned to effect a tradeoff between precision and time; the narrower the band(s), the more reliable the ordering of the solutions. For instance, if there are

five proofs with design costs $\{6, 7, 8, 8, 9\}$, and the width of the cost band c is 5, all of these solutions will be found on the second iteration of the cost loop, but not necessarily in best-first order. On the other hand, if $c = 1$, the solutions are guaranteed by the algorithm to be found in increasing order of cost, but at considerable added computation (many more iterations, or passes through the search space, are required).

The iterative deepening meta-interpreter approach ensures that the first solution to be found is the lowest cost design for the most probable model. Note that setting the initial probability bound to zero results in getting the explanations in an order that depends only upon the design costs. Setting the initial cost bound to infinity results in an order that depends only upon the recognition probabilities.

The underlying representation is compatible with an existing implementation of Poole's Probabilistic Horn Abduction Framework, which maintains an ordered queue of partial proofs [161]; in that implementation, partial proofs are not discarded, but suspended and queued when some other partial proof becomes preferred. Some small changes are required to the existing implementation of Poole's queuing mechanism before it can accommodate the separation of recognition and design assumptions advanced in this thesis; future prototypes may include these modifications.

Intent: formal definition

Intent-based authors are free to deploy the full power of the underlying representation language to specify their intents. Formally, an intent is defined as follows:

Definition 5 — Intent: An **intent** $I(Pr)$ is a predicate over presentations which is true when the presentation Pr satisfies the author's intent.

For instance, the intent to convince the user might be captured in a `convince` predicate that encodes an argument structure consisting of an introduction, a body,

and a summary (each of these with its own rules).

As an example of the specification of an intent, see the more detailed description of the `show` intent provided in Figure 5.8.

Design: formal definition

Formally, a design is defined as follows:

Definition 6 — Design: A **design** (given a model R) is an explanation D of $I(Pr)$ consisting of only design assumptions: $D \subset \mathcal{D}, \mathcal{F} \cup R \cup D \not\models \emptyset, \mathcal{F} \cup R \cup D \models \exists_{Pr} I(Pr)$. The **cost** of a design is the sum of the costs of the design assumptions of which it is composed: $\mu(D) = \sum_{d \in D} \mu(d)$. The ‘best’ design for model R is the least cost design that is consistent with R .

A design “produces” a **presentation** of information (*e.g.*, graphs, video, text). Note that different designs may produce the same presentation (there may be different reasons for presenting the same information). The best presentation in the context of model R is the presentation Pr supported by the least cost design that is consistent with R .

Definition 7 — *produces*(D, M, Pr): We define for notational convenience the relation *produces*(D, M, Pr), which is true when design D and model M lead to presentation Pr as described here, *i.e.*,

$$\textit{produces}(D, M, Pr) \stackrel{\text{def}}{\iff} \mathcal{F} \cup M \cup D \not\models \emptyset, \mathcal{F} \cup M \cup D \models \exists_{Pr} I(Pr)$$

The partitioning of \mathcal{H} partitions each explanation of $Obs \wedge \exists_{Pr} I(Pr)$ into a model and a design component which we denote as $\langle R, D \rangle$. We define a preference relation \succ_p over explanations such that:

$$\langle R_1, D_1 \rangle \succ_p \langle R_2, D_2 \rangle$$


```

show(P) <=
    % UserType is instantiated in the following line:
    interested(UserType, Topic1),          % a first topic
    interested(UserType, Topic2),          % a second topic
    % both Topic1 and Topic2 interest users of type UserType
    % Topic1 and Topic2 are not the same:
    different_topic(Topic1, Topic2),
    % make a bunch of assumptions about the user and then find
    % model-specific topics to present...
    gender(Gender),
    geo(Geo),
    availableTime(UserType, Time),
    % accumulate an edit-list
    editList([], P1, intro, 0, L1),        % an introductory subsequence
    editList(P1, P2, Topic1, L1, L2),     % sequences for first and
    editList(P2, P3, Topic2, L2, L3),     % second topics
    % show an interview with someone of the same gender,
    % in the same area or orientation
    interview_clip(X, P3, P4, L3, L4),    % interview
    gender(X, XGender),
    same_gender(XGender, Gender),
    % also, if the user is not local, show interesting Vancouver
    % mountains otherwise, skip it
    scenic_clip(Geo, P4, P, L4, L),
    costLength(L, Time).

```

Figure 5.8: **Show: an example intent**

iff

$$(P(R_1) > P(R_2) \text{ or } (P(R_1) = P(R_2) \text{ and } \mu(D_1) < \mu(D_2)))$$

This results in a lexicographic ordering of explanations. So, the best explanation consists of the most plausible model of the user and the lowest cost design.

Note that the design D which explains presentation Pr in $I(Pr)$ for some model R is not necessarily generated for some other model R' , *i.e.*, there is no reason why $produces(D, R, Pr)$ should imply $produces(D, R', Pr)$. The logic is a means of “weeding out” incoherent designs, and hence presentations.

This separation of model recognition and presentation design assumables also results in interesting ramifications for the way presentations are chosen; in particular, using \succ_p means that we do not give up good models for which we can find only bad designs. For instance, consider the case where we have disjoint assumables $\langle student, P_s \rangle$ and $\langle faculty, P_f \rangle$, where $P_s \gg P_f$, but the lowest cost design in the context of a model that assumes the user is a *student* is greater than the one in the context of a model that assumes the user is a *faculty* member (*i.e.*, $\mu(D^{best} | \dots student \dots) \gg \mu(D^{best} | \dots faculty \dots)$). We do not give up the assumption that the user is a student; the reasons for deciding in favor of *student* are not affected by the system’s inability to find a good (low-cost) presentation. This behavior is a direct consequence of the methodological separation of design from recognition assumables. Were there only a single space of assumables, the system would not be able to make these distinctions, and would simply select the best model/design combination according to the ordering metric.

5.5 Interactivity and Scrutable Models

One criticism of systems which attempt to model their users is their inscrutability to these users (see, for instance, Cook and Kay [52] and Orwant [154]). If the system acts in an infelicitous manner which does not meet the needs of the user, and

where this action results from an error in the user model, a desirable property of the system is *scrutability*.

Users should also be given the means by which to express their dissatisfaction with a presentation (or with elements of a presentation), always with an eye to providing users with better presentations.

This interaction paradigm supports both the scrutability and the dissention conditions described above. In calculating a model that represents the user, the system makes observations upon which further reasoning is conditioned. The user is provided with direct access (via a graphical user interface) to salient elements of the user model, as well as with the ability to criticize these elements. It is this interaction which is used to evolve the user model from a basic model initially derived from such information as the location of the user's terminal, login id, Internet domain, and so on.

Sensitivity Analysis

A problem with displaying the model to the user in support of scrutability is that the model may be huge. Certainly, there may be too many recognition assumptions to be effectively displayed on a screen, and far too many for quick assimilation by users. This cognitive comprehension task for the user is characterized by focused, serially-directed visual attention. Research in psychology has shown that the time taken for such tasks is proportional to the number of objects in the visual field [193].

A solution is to display a salient subset C of R , to which we refer as the *salient partial model*, consisting of those assumptions which have the greatest effect on the design (and therefore upon the presentation as well). The cardinality of C can be varied to maintain an appropriate pace of interaction; if the user is taking too long to evaluate the alternatives, the number of alternatives in the next iteration can be reduced, and vice versa. In addition, since the alternatives can be ranked,

[graphical] display techniques can be used to render this ordering to the viewer. In this section we describe the sensitivity analysis currently used by the system to select the critical recognition assumptions.

In the following, let $R = \{r_1, r_2, \dots, r_n\}$ represent the currently most plausible model, D the best design that is consistent with R , and Pr the presentation generated. We require a total ordering of the assumptions $r_i \in R$ by which to rank these assumptions for display to the user. Some useful notation follows.

Definition 8 — $C_p(Pr, \alpha)$: Let $C_p(Pr, \alpha)$ be the (lowest) cost of the (best) design that produces presentation Pr in the context of some model M that contains assumption α , *i.e.*,

$$C_p(Pr, \alpha) = \min_{D, M: \alpha \in M \wedge \text{produces}(D, M, Pr)} C(D)$$

To sort the $r_i \in R$ from most to least influential, we now need an expression parameterized by the assumptions $r_i \in R$, which we can use as a measure of the influence of r_i on the cost of D ; we want to know how much of a mistake we would be making if r_i does not correctly model the user, *e.g.*, if we have assumed that the user is a faculty member whereas in fact the user is a graduate student. An expression that serves this purpose is defined as follows:

Definition 9 — **Sensitivity**: If $\text{produces}(D, R, Pr)$, then the **sensitivity** of Pr to an assumption $r \in R$ is the expected cost of the presentation Pr (to users which are incorrectly modelled by r). Let J be the disjoint set that contains r :

$$S(r, Pr) = \sum_{r_i \in J} C_p(Pr, r_i)P(r_i)$$

Definition 9 does not consider all possible alternatives to R ; if more than one $r_i \in R$ do not correctly model the user, this formalism will not be able to diagnose

the multiple fault. This “myopic evaluation” [176, p490] is employed instead of joint sensitivity analysis which would consider multiple simultaneous errors in the model because: 1) single faults are more likely than multiple faults,¹⁰ 2) accounting for multiple faults is exponential in the size of the model whereas the myopic approach is linear, and 3) the effects of multiple faults would be very difficult for users to track, resulting in a huge cognitive burden that compromises the scrutability requirement.

The assumptions in the model R are sorted on the sensitivity of the presentation to each of the assumptions (*i.e.*, to each of the r_i for all i). The salient partial model $C = \{c_1, c_2, \dots, c_k\}$, referred to above, consists of the first k assumptions in this sorted list. These are the assumptions to which the design, given the current user model, is the most sensitive.

In the interaction paradigm related here, the user is shown the assumptions in C , in the context of the disjoint sets to which they belong; *i.e.*, where $c_i = \langle Name_i, P_i \rangle$, the system displays $Name_i$ along with all the names of the assumables in the disjoint set to which c_i belongs. For example, if $Name_i = faculty$ and some $J = \{\langle faculty, 0.6 \rangle, \langle student, 0.3 \rangle, \langle staff, 0.1 \rangle\}$, the user sees a set of names **faculty**, *staff*, *student* with the actual assumption highlighted.

The number of hypotheses displayed, k , is set to some small integer like 5 or 7. In sample implementations so far, simple text is attached to radio buttons, so that users see not only the assumption that was made, but the other members of the disjoint sets to which the assumption belongs; a simple click on a radio button instructs the system to recalculate the user model with the new assumption. See Section 5.6 for a description of the GUI by which C is communicated to the user.

¹⁰*I.e.*, if $p \ll 1$, then $p^n \ll p$.

5.6 Example

The following example is taken directly from the prototype implementation described in the next chapter; specifically, the reasoning and modelling strategies described in this thesis have been embedded in an application called *Valhalla*, an automatic video presentation tool that uses the reasoning strategies we have described to select, order and play video segments from the Departmental Hyperbrochure described in Section 1.4.

This application uses the authorial intent called “show,” the predicate presented in Figure 5.8.

In this example, the system was started from scratch after the user had logged in with the personal user id, `csinger`. The user immediately requested a presentation from the system, which responded as follows (comments have been added for clarity):

The initial model, with likelihood 0.029494, is composed of the following recognition assumptions:

<code>married(yes)</code>		the user is married
<code>orientation(theory,y)</code>		orientation: theoretical
<code>geo(local)</code>		is local to region
<code>area(ai,y)</code>		research area is ai
<code>c_gender(male,student)</code>		is student, and consequently male
<code>userType(student)</code>		is student

These are based upon prior probabilities and other contextual information such as the user’s login id. At UBC, we consult a local personnel database to make a best guess about the user’s membership in a variety of categories; this database informs us, for instance, that user `csinger` is a graduate student, and that user `poole` is a faculty member. Using this source to justify the assumption that a user with a certain login id belongs either to the class *faculty* or *student* is reasonable, but not infallible. User `csinger` might have logged in a visiting faculty member, for

instance, so that he or she could use *Valhalla* to learn more about the department while Csinger attended to an unavoidable interruption in the meeting they were having.

The best design based upon this model “costs” 51 and is composed of the following design assumptions:

<code>overLength(25)</code>	amount by which exceeds optimal presentation time
<code>rel(topic(lci),topic(research),20)</code>	cost of relevance assumption: lci is relevant to research
<code>rel(video(sports),topic(sports),5)</code>	cost of relevance assumption: a video clip about sports is relevant to the topic of sports
<code>rel(topic(introduction),intro,1)</code>	cost of relevance assumption: a clip called intro is relevant to the topic “introduction”
<code>different_topic_cost</code>	the topics included are not identical

Associated with each design assumption is a cost. The cost of the assumption `different_topic_cost`, for instance, is less than the cost of the assumption `same_topic_cost`, from the same disjoint set; this arrangement prefers presentations with multiple topics over those with single topics. Some costs are context dependent: the cost of the `overLength` assumption is proportional to the amount by which the length of the current presentation exceeds the optimal length for a user of this type.

The corresponding presentation P_1 is:

```
(0:1:26,0:2:27,topic(introduction)),
(0:24:44,0:25:3,video(sports)),
(0:12:15,0:15:54,topic(lci)),
(0:31:54,0:32:44,interview(jim_kennedy))
```

Specifically, P_1 consists of four clips from the video archive with the indicated absolute time-codes and topic identifiers.

The user at this point indicates dissatisfaction with the current presentation, perhaps after viewing part of it via the Virtual VCR interface, by clicking on the button provided for that purpose. The system interprets this action as a request for another presentation using the same user model.

The next best design based upon this model “costs” 56 and is composed of the following design assumptions:

```
overLength(35.0)
costRelevant(topic(lci),topic(research),20)
costRelevant(topic(introduction),intro,1)
different_topic_cost
```

The corresponding presentation is:

```
(0:1:26,0:2:27,topic(introduction)),
(0:12:15,0:15:54,topic(lci)),
(0:31:54,0:32:44,interview(jim_kennedy))
```

The costs of the two designs differ because there is a preference built into the system for presentations of an optimal duration, defined by the user’s membership in certain categories.

The user can now navigate through the presentations with a virtual VCR interface.

Which assumptions are displayed in the user model window? This is determined by the sensitivity analysis algorithm, the results of which are shown in Table 5.3; this table indicates sensitivity calculation results for the first model and design pair described in this example. The second column shows the calculated

r_i	$S(r_i, P_1)$
geo(local)	104
c_gender(male,student)	96
userType(student)	61
area(ai,y)	51
orientation(theory,y)	51
married(yes)	51

Table 5.3: **Results of Sensitivity Analysis before user action**

sensitivity for the assumption in the first column. The presentation is most sensitive to the assumption that the user is local, and completely insensitive to his research area or orientation, or to his marital status.¹¹

Because the assumptions are only hypothetical, provision must be made for their revision, both in terms of the underlying reasoning engine, and in terms of a mechanism whereby such revised data can be acquired. In particular, in support of the scrutability desideratum, users can interact with the system to validate or correct the salient assumptions of the model.

Valhalla gives the user the means by which to validate or correct assumptions that the system has made by displaying critical assumptions in what we call the *user model window*, an instance of which is shown in Figure 6.5.

Returning to our example, the user informs the system that he is not, in fact, local, but a prospective student by clicking on the appropriate radio button.

The system then re-calculates, with the result that the new model, with probability 0.045375, is:

`married(yes)`

¹¹*I.e.*, $S(r, P_1) = C(D) = 51$, when r is an area, orientation, or marital status variable. In other words, the cost of the original best design is the same as the cost of designs induced by the sensitivity metric with respect to these assumptions in the context of presentation P_1 .

```

orientation(theory,y)
area(ai,y)
c_gender(male,student)
userType(student)
geo(prospective) # user is not from the region

```

The new design, with cost 137, is:

```

costRelevant(topic(ubc_scenic),mountains,1)
costRelevant(topic(introduction),intro,1)
different_topic_cost

```

The corresponding presentation P_2 is:

```

(0:1:26,0:2:27,topic(introduction)),
(0:18:17,0:28:27,topic(ubc_scenic))

```

r_i	$S(r_i, P_2)$
geo(prospective)	267
c_gender(male,student)	242
userType(student)	237
area(ai,y)	237
orientation(theory,y)	227
married(yes)	227

Table 5.4: Results of Sensitivity Analysis after user action

The contents of the user model window depend again upon the results of the sensitivity analysis, shown in Table 5.4, where the second column shows the calculated sensitivity for the assumption in the first column.

In presentation P_1 , a preference for multiple topics is in evidence. This preference stems from forcing the system to make an assumption whose cost depends upon the relationship between elements in the presentation. The system currently values diversity over emphasis, as described in Section 5.4.2.

In presentation P_2 , the sensitivity to the assumption of locality is demonstrated. When the user was assumed to be local (in P_1), there was no clip in support of scenic views of UBC, but when the user is assumed (in this case because the user actually informed the system) to be prospective and therefore not local, a scenic clip is included. Note that this clip is included at the expense of the earlier research and sports clips; the system currently places a greater value on showing UBC's scenic merits to prospective department members, than on any other design element.

5.7 Alternative Approaches

5.7.1 Decision Theory for Multimedia Presentation Design

Here we explore how decision theory could be used in the service of the stated goals of intent-based authoring, and argue that expected value (see Section 4.2 for background) is probably not the right approach. In this section we focus on the interaction paradigm in support of what has been referred to in this thesis as *scrutability*: a scrutable system is one that makes clear to users the relationship between the model of the user maintained by the system, and the behavior of the system. The means by which scrutability is achieved in the system is an approach to interaction that permits users to critique the user model in pursuit of better system behavior. An important element of this approach is the approach to sensitivity analysis, described in Section 5.5.

Cheeseman [40] discusses the use of decision theory for design, and has suggested that the best design is the one that results from averaging over all possible

models and maximizing expected utility or minimizing expected cost; in Equation 5.2 the expected value of the cost function C over designs is to be minimized, and the best design is $D : E(C(D))$ is minimal.

$$E(C(D)) = \mu(D|M_1)P(M_1) + \mu(D|M_2)P(M_2) + \cdots + \mu(D|M_n)P(M_n) \quad (5.2)$$

$\mu(D|M)$ is the cost of design D in the context of model M , and $P(M)$ is the posterior probability of model M (refer to Section 5.4 for formal definitions of the terms “model” and “design,” which are used informally here). There are reasons in the current application why this may be the wrong thing to do.

For example, consider the following scenario. Assume, for simplicity, that users of the Departmental Hyperbrochure (see Section 1.4) can be only *faculty* members with prior probability 0.6 or *students* with prior probability 0.4, respectively, and that the following costs are known: $\mu(\textit{sports}|\textit{faculty}) = 10$, $\mu(\textit{sports}|\textit{student}) = 5$, $\mu(\textit{research}|\textit{faculty}) = 5$, $\mu(\textit{research}|\textit{student}) = 15$. Assume also that a current user of the system is a faculty member, that the system has assumed that the user is a faculty member, and that this assumption is clearly indicated to the user. Then, $E(C(\textit{sports})) = 8$, and $E(C(\textit{research})) = 9$. There is no indication to the user why the former is chosen for presentation over the latter, and the user may wonder why he is watching an irrelevant basketball game. The faculty member using the system may not be aware of the high degree of aversion on the part of students to research—a value not shown to the user (because there may be a very large number of such values). The behavior of the system is inscrutable.

It may be better to make a mistake and admit it, giving the user the means to correct the contextual error, rather than average over all possible mistakes. There are exceptions, and it is not difficult to conjure counterexamples under slightly different assumptions from the ones that guide our own effort; consider a scenario where it is possible to present material that is known to be highly offensive to a

particular category of user, but which is known to be greatly appreciated by another; pornography is one such delicate issue today, and policies are under review at academic departments around the globe. Another example: one would not want to present a viewer with sensitive corporate data on the assumption that he or she is a joint-venture partner, only to find out later that the viewer is from a competitor's firm.

The system calculates the most likely user model, and then uses it to design the best presentation. Refer to Section 4.1.1 for background material about the formalism.

5.7.2 Costs and Utilities

This section is a discussion of how the values associated with the design assumables should be interpreted.

These quantities can be interpreted in a number of implementation-dependent ways: it could be an estimate, for instance, of how hard it is for the system (from a computational point of view) to realize the design element, or of how much cognitive or perceptual effort (the system thinks) will be required from a human to comprehend some manifestation of the design element. The values are used to constrain choices in the design space (for instance, to prefer some alternatives over others as in the example provided in Section 5.6); this interpretation sees the values as a measure, attributed by the system on behalf of the user, of how relevant a design element is to the user modeled by the current user model.

The investigation in this thesis has been formulated in terms of a system which minimizes positive design cost; this approach encourages designs which are "minimal" with respect to the number of design assumptions made. All costs associated with design assumables are positive, and the function used to accumulate total cost is simple summation, which means that the total cost function is monotonic in the sense that it can only increase as additional assumptions are made over the course

of the proof. It is also “unbiased” in the sense that it is possible only to express relative preference (see Expressiveness, below).

Another possibility would be to implement the dual system which maximizes positive utility; this approach would produce designs which are “maximal” in the number of design assumptions. These distinctions are important because they affect the expressiveness of the underlying knowledge-representation language, as well as the computational complexity of the implementation that is built with it.

Expressiveness: It is not the same thing to say that ‘presentations that do not depict sports events are preferred over those which do’ and to say that ‘do not show sports events.’ The first can be expressed with a system that maximizes positive utility by setting the utility of sports events to be lower than any other utilities, or with a system that minimizes positive cost by setting the cost of sports events to be higher than the costs associated with presenting other kinds of events. The second requires an ability to express negative bias (‘user hates sports’) which cannot be represented in a system that uses a monotonic utility function; making additional assumptions can only increase cumulative cost (or decrease utility) and decrease cumulative probability.

Complexity: The aforementioned expressive power is given up in favor of monotonicity¹² because non-monotonic utility functions would require the system to generate the entire search tree before reporting the best solution: branches would have to be followed to their leaves because arcs could be labelled with negative values, and all branches would have to be followed before it could be known which of them lead to the best design. (Partial proofs would do the system no good at all.) As the search space may be very large, this would be an unacceptable strategy in

¹²The term “monotonicity” is used here not in the logical sense of non-monotonic reasoning, but in the classical mathematical sense of a function whose derivative does not change sign.

our presentation domain, which currently benefits from a best-first search strategy (see Section 5.4.2).

These issues can not be simply dismissed by arguing that the system could shift the scale of costs or utilities associated with the design assumables so that the lowest cost or the lowest utility is set to zero. This compromise strategy taken to permit the expression of negative bias while retaining favorable complexity parameters, introduces complications of its own. Consider, for instance, a candidate design with a “research” component (with utility of 100) and *ten* “sports” components (each with utility of ≈ 10); such a design would have a net utility of zero, and any other design with a positive utility would be preferred to it. For instance, a candidate design with a single “entertainment” component with total utility 5 would be preferred. Assume also that there are no design assumables in the database with associated utilities that are “worse” (less than) ≈ 10 .

If the assumables in the database were preprocessed by the system as discussed above, the assumption associated with the “research” component would now have a utility of 110, and the assumption associated with the “sports” component would have a utility of zero. The candidate design would now have a net utility of 110, and would now be preferred to the “entertainment” candidate, which is now worth 15. Such changes in design preference would be difficult for knowledge engineers and intent-based authors to foresee, and so this scaling remedy does not appear to be a good solution. Again, the ability to explicitly express bias is given up in favor of mathematical monotonicity, which supports the best-first search strategy.

5.7.3 Other similar approaches

Wu [200] uses passive recognition techniques to build a model of a dialog partner, but decision theoretic measures are used to decide when the system should intervene to make direct inquiries of the user. The system is said in such an event to have adopted an *active acquisition goal*; these AAGs are isomorphic to the salient

set of assumptions described above, in the sense that they are the ones judged by the system using decision theoretic measures to be the (only) ones with which it might be worth distracting the user. The approach taken in this thesis differs significantly, however, in that the user can continue to ignore the user model window, and in that even this inaction can be used as a basis for drawing further conclusions.

5.8 Conclusions

Guided by a desire to build cooperative systems, a way was sought to acquire, represent, and exploit simple models of users. The models are acquired with an abductive reasoning strategy, and consist of assumptions about the user, drawn from a pool of possible hypotheses called recognition assumables. Associated with each of these assumables is a probability, and the probability of a model is the product of the probabilities of the assumptions composing the model. The model explains observations of the actions of the user at the interface to the system.

Design assumptions must be consistent with model assumptions, and are drawn from a pool of possible hypotheses called design assumables. Associated with each of these assumables is a cost, and the cost of a design is the sum of the costs of the assumptions composing the design. The design and the model together explain a presentation that satisfies the communicative goal, or intention, of the author.

The requirement of *scrutability* was identified, which is the constraint that the behavior of the system should be seen *by its users* to be a consequence of the user model. It was found that this goal is not met by expected value approaches to design.

The minimal AI approach to user modelling described here services the scrutability condition by providing the means to perform a kind of sensitivity analysis on components of the user model, so that only the most critical elements are displayed to the user, who can then criticize them. The contribution of this thesis is both an

interaction paradigm that permits the user to “debug” the user model, and a probabilistic Horn abduction approach to reasoning that implements the paradigm.

Chapter 6

Implementation

But men do not begin to act upon theories. It is always some real danger, some practical necessity, that produces action; and it is only after action has destroyed old relationships and produced a new and perplexing state of affairs that theory comes to its own. Then it is that theory is put to the test.

—H.G. Wells
Outline of History, page 693

A prototype of the system described in the rest of this thesis has been developed. It embodies the intent-based authoring ideas advanced in this dissertation: *Valhalla* is a scrutable system that selects and orders video clips from a repository of material describing the Department of Computer Science at the University of British Columbia, and the examples in this chapter are drawn from the Departmental Hyperbrochure, described in Section 1.4. *Valhalla* decouples the specification from the presentation tasks of authoring, abandoning the traditional model in favor of the intent-based paradigm.

The author brings an intent, and information he thinks will be relevant to the eventual presentation. After annotation, a representation of this intent, and a set of indices into the raw video reside in a “document.” This is all done at compile-time, in the absence of the viewer. Later, at run-time, the reasoner uses the docu-

ment, along with the user model and other knowledge, to produce an edit-list. The viewer, even in the absence of the author, sees only the most relevant portions of the video when the user model is accurate, and can take remedial action when it is not.

6.1 Architecture

A distributed architecture was chosen for the prototype implementation. Populating Valhalla's framework are a number of autonomous agents that fall into the following classes: client applications, network-based multimedia servers, and other service providers such as reasoning engines and annotated video databases. Figure 6.1 illustrates agents within the Valhalla application framework.

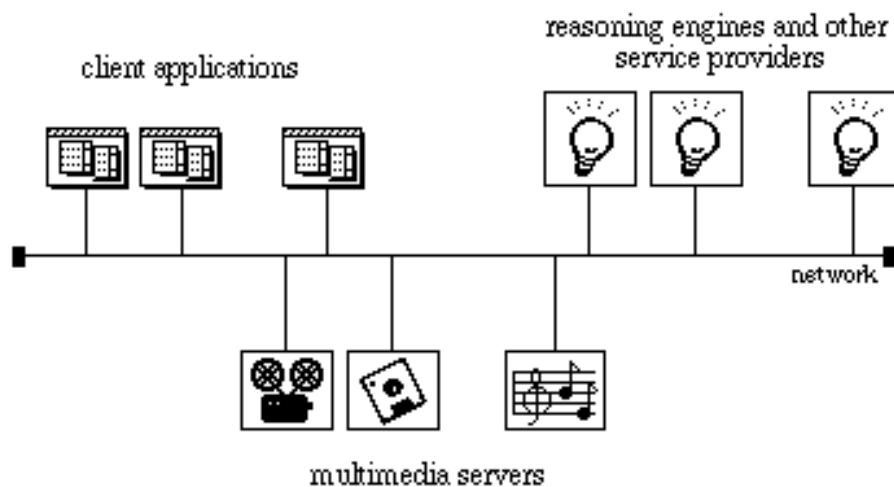


Figure 6.1: Valhalla's Distributed Architecture

Each class of agent has associated with it a single communications protocol,

which allows client applications to transparently communicate with all instantiations of that class. It is this “plug and play” compatibility between members of a class that makes the Valhalla framework flexible and powerful.

One member of each class of application has been currently implemented in the prototype framework. A client application, also named Valhalla, provides a user with intent-based authoring services. This client uses the services of a reasoning engine to generate descriptions of video presentations, and displays the video sequences within the presentations using the services of a distributed multimedia server.

The Multimedia Server

From the perspective of client applications, the server is a single entity providing virtual VCR-like control over multiple media sources. The server has the ability to simultaneously operate in two modes: in local mode, media sources are conventional analog devices, whose output may be routed to a number of available display devices using an RS-232 controlled switch. In remote mode, digital media is transmitted over the network to the client application from one or more remote sites, and is controlled using media playback applications present on the client’s host.

The server is implemented as a series of increasingly abstract application programming interfaces (APIs). Each interface can be directly accessed by a client application, but typical operation of the server would only involve access from the highest (and most abstract) layer. A higher layer interface uses the services of a lower layer in order to provide its own services. Figure 6.2 illustrates the relationship between these interfaces and the various components of the server.

The lowest layer, called the device layer, is accessible only through TCP socket communications and is composed of a series of device drivers. There is one device driver for each physical device, and one driver responsible for dispensing a class

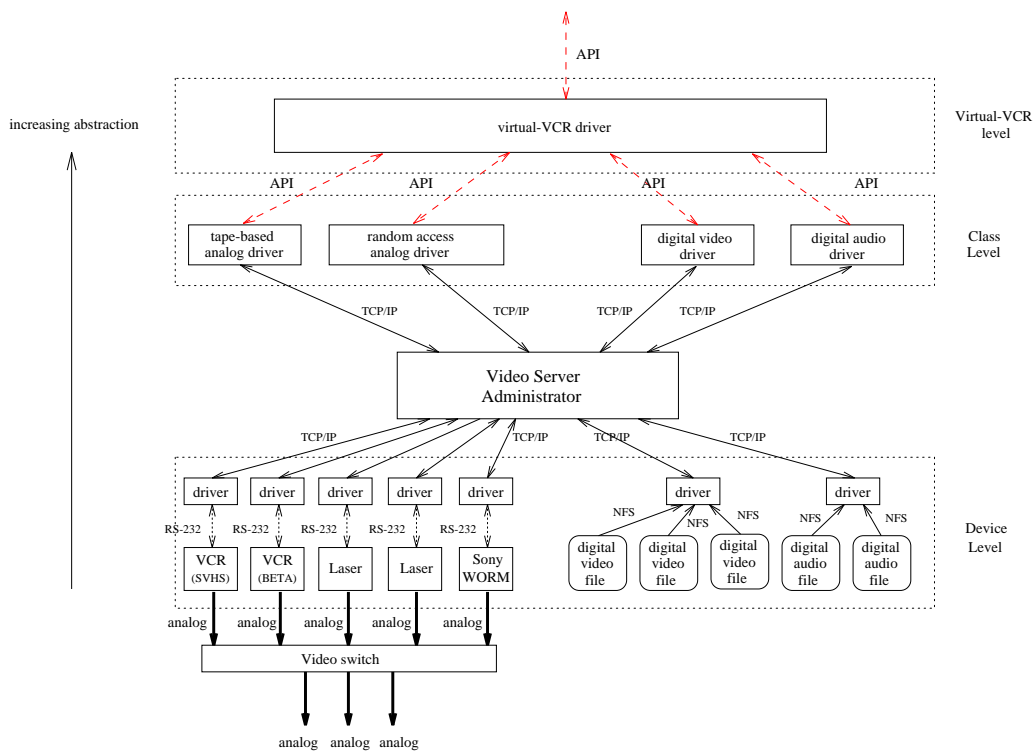


Figure 6.2: The hierarchy of interfaces to the video server

of digital data.

The next most abstract layer (the class level) is accessible to client applications via an API to a library that directly communicates with the server on behalf of the client. This intermediate layer provides completely separate interfaces for each class of device. Devices of the same class share characteristics particular to that class. Providing a separate interface for each class allows client applications to take advantage of these particular characteristics. Supported classes currently include digital video, digital audio, random-access analog video (optical video disc), and tape-based analog video. Future extensions to the server will support other device classes, such as text, MIDI or digital audio.

The top layer, called the virtual-VCR level, is again accessible via an API and provides an interface that client applications can use to control any device. Characteristics of different device classes have been abstracted away in order to provide a single VCR-like interface.

Using an API to communicate with the more abstract layers of the server is advantageous for a number of reasons. The inclusion of a library into the client executable facilitates the distributed architecture of the server; each library can be considered to be an agent of the server executing on the client's host machine. The presence of the server on the client's machine allows the server to directly control applications on the client's host. Such applications would be used for the playback and manipulation of digital media. Secondly, the API itself has been designed to provide a uniform method of interacting with multiple media sources and formats, allowing the differences between classes of media to be partially abstracted away.

The video delivery component of the system is designed to handle tape, video disk and digital video through a video server mechanism currently implemented on a Sun architecture [88].

The Valhalla Client Applications

The first Valhalla client prototype has been completed, and another is currently under development.

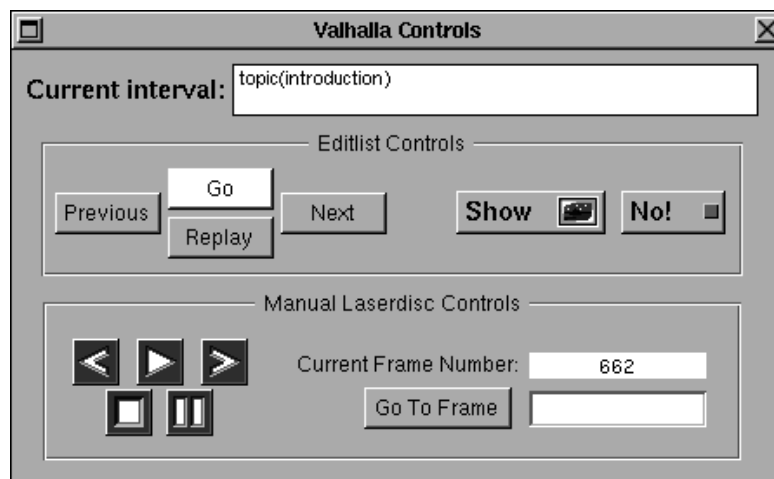


Figure 6.3: **The Valhalla Control Window**

Figure 6.3 depicts the *Valhalla* control window, the main human interface to the Valhalla client, which is implemented in Objective C and currently resides on a NeXT cube equipped with a NeXT Dimension board. The control window contains—in addition to the familiar virtual VCR control panel at the lower left—controls to advance to the next clip in the current edit-list, to return to the previous clip in the current edit-list, to replay the current clip, and to proceed with the presentation (“Go”).

Depression of the “Show” button is interpreted as a request to calculate the next best presentation for the current model, and is passed on to the reasoner. The “Show” button is “wired” to a predicate in the reasoner’s knowledge base that encodes a specific authorial intent; for the Departmental Hyperbrochure project de-

scribed in Section 1.4, the underlying intent of all presentations is always to inform the viewer, to make maximally relevant presentations of the research and other departmental activities to individual viewers. Other applications using the same client interface program may have the “Show” button wired to other predicates encoding other intentions, and it would not be difficult to provide multiple buttons on the interface so that the user could select different (pre-defined) intentions at run-time. It is also possible in the current version of Valhalla to author new intentions at run-time through a special query window that is not shown or discussed further in this document; this avenue is not pursued because the system does not demand that the human in the role of viewer be able to program in Prolog, and to understand first order logic. (It does currently require the authors and knowledge engineers involved to have these skills.)

“No!” is merely a direct way for the user to express dissatisfaction with the current presentation, freeing the reasoner to recalculate both model and design as required. Any activity at the control window is echoed to the reasoner, which can use plan recognition techniques to infer the motives of the user from these observations of user behavior. In the figure, the label of the current interval as provided in the annotation database is displayed. Manual laser disk controls include absolute frame indexing.

The services of the multimedia server are used to display clips contained in an editlist. Currently, all media associated with the Hyperbrochure is in the form of (synchronized) analog video and audio stored on CAV laserdiscs. Analog signals from two laserdisc players are routed to a video digitizer board within the NeXT computer, and the resulting digital video is displayed in a window on the NeXT’s display. An example frame from a campus sporting event is shown in Figure 6.4.

The needs of individual users are met by referring to the user model, which is arrived at by the reasoning method outlined earlier in this thesis. As discussed earlier, the user is given the opportunity to critique a selected subset of the model



Figure 6.4: **A frame from the Departmental Hyperbrochure**

via *Valhalla*'s user-model window, shown in Figure 6.5. The hypotheses actually displayed to the user are context-dependent, selected and ranked by a sensitivity analysis algorithm (described in Section 5.5) that reflects the degree of importance to the design, of each assumption in the model. In addition, the techniques employed to display these assumptions reflect their relative importance; quantities to which the design is most sensitive can be shown, for example, in bolder fonts, brighter colors, larger characters, and so on (*cf.* perceptual salience, Section 5.4.1). Every effort is made to sanction the further assumption by the system that the user has actually seen and attended to the display in the user model window. The *Valhalla* User Model window implements the interactivity paradigm advocated in Section 5.5. Clicking on any element of the User Model window first sends a message to the Reasoner which causes it to succeed in its processing loop; this message is followed by an instruction that encodes the change to the user model that the user has just specified. The Reasoner makes the requested change to the user model and then calculates a new presentation. In Figure 6.5, the screen shows a number of sets of radio buttons, which is *Valhalla*'s display technique for

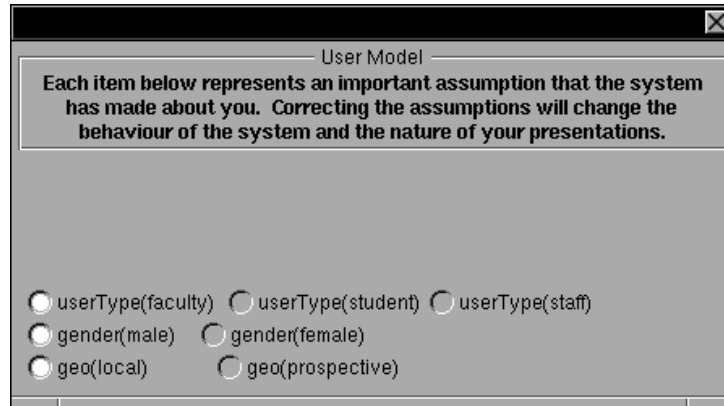


Figure 6.5: **The *Valhalla* User Model Window**

variables whose values are drawn from an exclusive (disjoint) set. Here, *Valhalla* believes the user is a faculty member; a student can correct the system’s misconception with a single click.

The pertinent elements of the user model selected by the reasoning engine are transferred to the hyperbrochure application in an abstract form. Instead of specifying a particular GUI “widget”, a particular hypothesis may be specified as an element of a (finite) discrete set, as possessing a value within a particular range, or as having a boolean value. It is left to the hyperbrochure to determine an applicable “widget” with which to display this to the user. This method of providing abstract, indirect control over the reasoning process adds to the flexibility of the framework.

6.1.1 The Reasoner

The Reasoner is a best-first Sicstus Prolog implementation of the assumption-based reasoning framework introduced in this paper. The knowledge bases are all written in a Prolog-like Horn clause language extended with assumptions (as described

in Section 5.4), and the annotation database consists of only definite clause assertions.

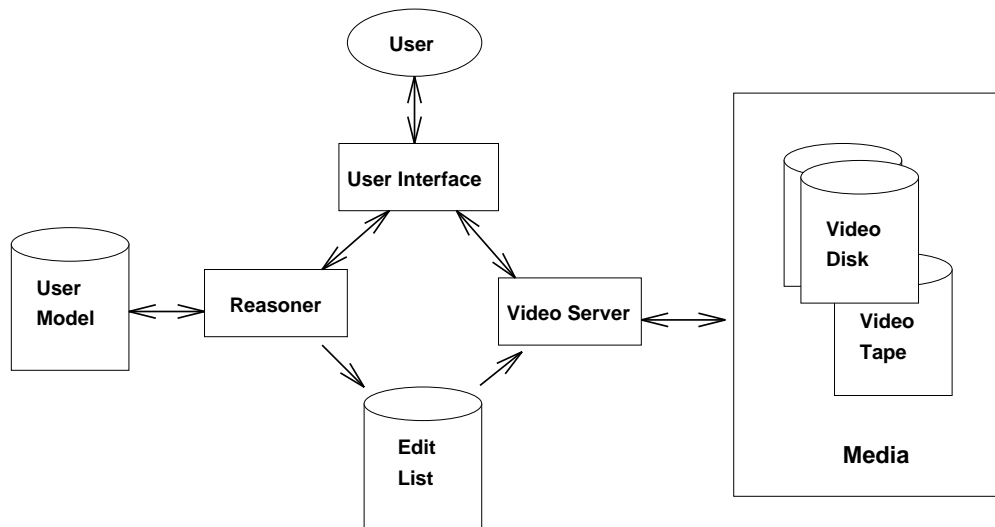


Figure 6.6: **Valhalla implementation**

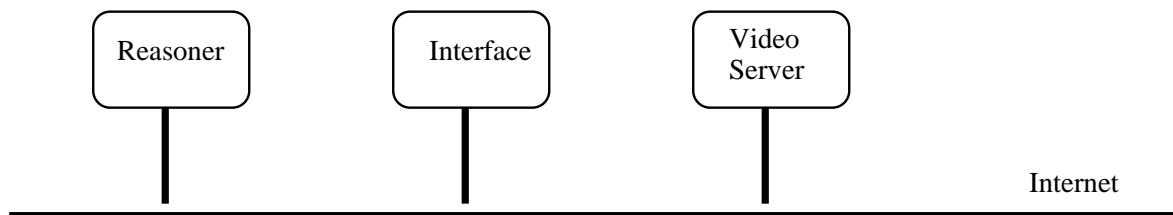


Figure 6.7: **The Valhalla Agents**

These aspects of the design can be seen in Figure 6.6. Connections between the video server, user interface and reasoning engine are all client/server (TCP/IP) links using standard Unix sockets, giving flexibility and platform independence. Figure 6.7 illustrates these relationships.

6.2 Functionality

```
? area(Student, graphics),           % Student studies graphics,
  supervises(FacultyMember, Student), % is supervised by FacultyMember
  relevant(FacultyMember, Topic),     % to whom Topic is relevant
  editList([], Presentation, Topic, 0, L), % Get a video edit-list
  costLength(L, 300).                 % close to 300 seconds
```

Figure 6.8: **Authorial Intent as a Prolog query**

Presentation is decoupled from specification by having the system prepare an edit list of relevant events and intervals subject to the constraints in the available knowledge bases. The generation of this edit-list is performed at run-time, rather than compile-time, so that the author need not be physically present to ensure that the presentation is suitable. The intent of the author is currently encapsulated in a distinguished predicate that is attached to a **Show** button on the interface, whose intended interpretation is that viewers should be given a basic overview of the material available, followed by a body which is relevant to their immediate information retrieval goals, and then by a conclusion; other author intentions could be similarly encapsulated and connected to the **Show** button, or to other buttons on some custom interface.

A number of constraints are applied to the design of the presentation, including for instance that its length not exceed a certain amount of time. The user's interaction with the system is restricted in this way to factor out variables that would make it difficult to test the impact of our user modelling approach. Note, however, that there is an additional window, not pursued in this dissertation, that can be used to make arbitrary queries of the reasoning engine, in the underlying representation language described in this thesis; the user can take the role of intent-based author by specifying his or her own intent in this window and instructing the system to find an appropriate presentation. For example, in Figure 6.8, an au-

thor might form this query to ask for a presentation of footage (optimal length of five minutes), relevant in some way to a departmental supervisor of a graduate student associated with the Computer Graphics research laboratory... Obviously, the full power of intent-based authoring is not realizable in the prototype without some facility with Prolog, and with the underlying reasoning and representation methodology; this is why we expect the user testing of *Valhalla* to make use of the **Show** button, which abstracts away from these complexities.

The system has been tested with the Departmental Hyperbrochure, introduced in Section 1.4. Potential viewers of the material are prospective and current graduate and undergraduate students, faculty and staff, funding agencies and industrial collaborators. All these are potential users of *Valhalla*, and each brings idiosyncratic goals and interests that the system attempts to meet with tailored presentations.

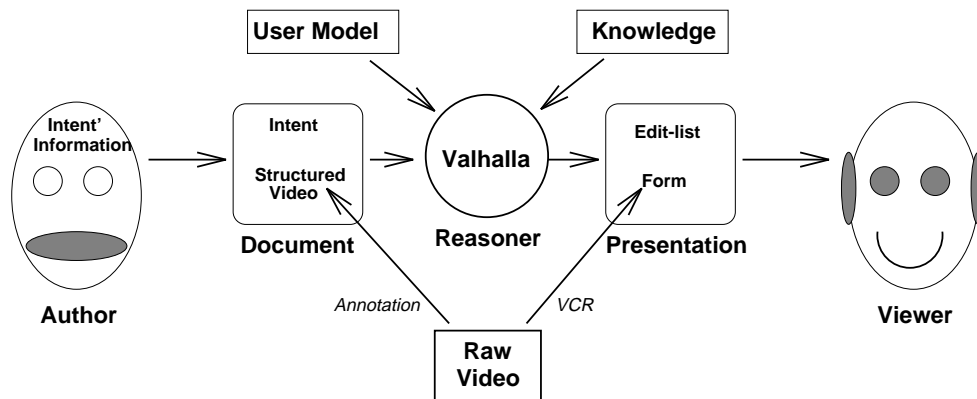


Figure 6.9: **Knowledge-based Video Annotation and Presentation**

The way the intent-based approach is mapped into the video authoring domain is shown schematically in Figure 6.9: the author brings an intent, and information he or she thinks will be relevant to the eventual presentation. After annotation,

a representation of this intent, and a set of indices into the raw video reside in a “document.” This is all done at compile-time, in the absence of the viewer. Later, at run-time, the reasoner uses the document, along with the user model and other knowledge, to produce an edit-list. The happy viewer, even in the absence of the author, sees only relevant portions of the raw video.

Although the current implementation has the Reasoner acting as a server to the client Interface, the simple protocol underlying communications between these agents permits the Reasoner to view the Interface as just another Prolog predicate: an “ok” from the Interface (caused by successful completion of a presentation to the user) allows successful completion of the Reasoner’s evaluation loop; any other message from the Interface (caused, for instance, by the user expressing dissatisfaction with the current presentation by clicking on the **No!** button in the Control window) forces the Reasoner to initiate a Prolog backtrack to find other solutions. Clicking on the **Show** button in the first place initiates calculation of the first presentation, using only prior probabilities for recognition assumptions.

Scenario: Company Meeting

Company XYZ holds a meeting to decide whether or not to build a new production facility. The meeting is supported by advanced Group Support System (GSS) tools, and complete records (minutes, keystroke logging, video, etc.) are kept.

Someone subsequently gets the job of creating a presentation of the meeting in order to document and justify the process by which the decision to build the new plant was reached. The intent of the ensuing presentations is to convince the viewer that the decision was well motivated; different viewers get different renditions of the argument. The marketing manager, for instance, gets a tailored account that focusses on sales forecasts for new plant operations. The accounting manager sees a spreadsheet with emphasis on cash-flow and detailed cost-benefit analyses. Potential investors see glossy images and video backup, along with statistics demonstrating improved delivery schedules—but nothing that would be useful to a competitor.

Any individual viewer of one of these presentations might be inspired to query the system for further information on any subject. He thereby becomes an author in his own right and specifies his own intent. A customer, for instance, might want to see how the decision to build the new plant will affect prices. He authors his own presentation of this information; whether or not the intent of the original author (to convince) plays any role in this last presentation is an ethical and practical issue of some interest, but unfortunately outside the scope of this thesis.

An example:

InterSpect Consulting Company's 1999 Yearly Report includes the following fiscal information (in millions of dollars):

- *Land usage costs: 12*
- *Data storage costs: 75*
- *Doughnuts and coffee: 105*
- *Communications costs: 23*

The presentation system has available to it knowledge about graphical presentation strategies and languages [129] [194], and reasons its way to an optimal

presentation for the president of InterSpect based upon this as well as a continually updated model of the president's goals, beliefs, desires and preferences. If the author's intention in preparing the year-end report was to fairly and accurately portray the company's expenses, the president might see a simple bar chart in which the suspiciously high doughnut and coffee bill would be immediately apparent. If the author's intention was actually an elaborate argument for even more resources to be allotted to the doughnut and coffee account, such a presentation element may not be prudent. Instead, another graphic would be designed at presentation time in which the offending quantity is de-emphasized, taking full advantage of the equipment available at run-time.

Part IV

Conclusions

Chapter 7

Conclusions

The whole duty of a writer is to please and satisfy himself, and the true writer always plays to an audience of one.

—The Elements of Style, page 85.

Most systems which are currently available to support authoring inherit the limitations of the traditional model of authoring, which were presented at length in Section 1.1 and recapped in Section 5.2. The foremost such limitation stems from the early commitment to content, which makes it impossible to provide user-tailored presentations at run-time. The problems are exacerbated in the video medium because it is temporally linear (and because humans have so little time), and because current techniques for automatic speech and visual recognition leave the contents uninterpreted.

A knowledge-based solution called intent-based authoring was described in Chapter 5 which mitigates the serious effects of these problems, by separating *intent* from content, analogously to how the structured document paradigm now separates content from form. The separation of intent from content and form enforced by the intent-based authoring paradigm frees the intent-based author to work at compile-time in the absence of the viewer to specify the intent, or communicative

goal underlying the eventual presentation. A run-time presentation system then makes the presentation by first deriving a model of the viewer and then tailoring a presentation that meets the author's objectives as given in the specification of intent, and the interests of the user as given in the model. The entire framework is represented in a Horn clause logic dialect, described in Section 5.4, which supports the recognition of user models, and the design of presentations, by abduction.

Other components of the framework described include a scrutability desideratum, to offset the possibility that the model contains errors. This desideratum requires that users be shown relevant aspects of the user model, so that they can understand the (mis)behavior of the system. Because the model may be very large, only a salient subset can be shown in general; this subset is determined by a sensitivity analysis, described in Section 5.5.

A prototype called *Valhalla* was implemented and tested, which manifests the intent-based authoring framework.

The central focus of this project continues to be the deployment of artificial intelligence techniques for user modelling. This work is performed within the limits of what has been called "minimal AI" to explore the simplest useful applications of probability, logic and decision theoretic reasoning strategies to the problems of modelling users of computer systems. The approach is very simple, based as it is upon well-tested notions from decision theory and the AI literature.

The intent-based authoring paradigm described in this thesis can be applied to different media, domains, and tasks. It has potential to circumvent limitations of the traditional model of authoring. Intent-based authoring *requires* the application of computational intelligence, a prospect which is only today becoming realistic; the future of authoring looks promising from this vantage point.

7.1 Implications

Authoring is hard, which is why technology has been used in the past to aid authors with their toil, and why this research has been undertaken to investigate the issues surrounding the task. However, it seems on the surface that intent-based authoring as advocated in this thesis is hard too—harder than traditional authoring, which is at least well understood, if limited in the aforementioned ways. So why should modern-day authors engage in the intent-based authoring paradigm? In addition to the reasons already discussed (*cf.* individualized presentations for individuals), is the reuse factor.

Intent-based schemata, once authored, can be reused. Knowledge, once appropriately indexed, can be reused, often in unanticipated ways. The principle of code re-use in software engineering is motivated in the same way [119]. While it is true today that an author will need to invest a great deal more work into the specification of an intent-based presentation than in the design of a traditional book or conference paper, the intent-based author and reader benefits from a more effective deliverable: the dynamic, intent-based document.¹

7.2 Future Work

Intent-based authoring captures the notion that authoring is distinct from viewing/reading, that the *design* of the document is not the same as its presentation. These processes have been separated and the functions of distinct sources of knowledge and information elaborated in the authoring and reading cycles. The author provides some of the information to be presented, and describes his intention as regards this information, but we can go further.

¹Intent-based authors of tomorrow will benefit from the presentation strategies and knowledge bases of all of his or her forebears, and, one day, the n -th intent-based presentation may take less effort from a human author than the corresponding traditional presentation would have.

The authoring system can support the author in this design or specification task by referring to knowledge sources that may prompt further elaboration on the part of the author. If the authoring system knows, for example, that the presentation facilities include video imaging capabilities, it may prompt the author to provide video information, if available. The authoring system may even have a partially instantiated model, or an aggregate model of the set of intended viewers of the presentation; this too may solicit additional contributions from the author.

Viewing, in turn, can be mediated by these same information sources. The model of the reader is consulted at run-time to design the presentation, and specific presentation elements are determined then by available resources: if the color printer is down, the presentation may have to be optimized for delivery on a black-and-white printer; presentations on an interactive display might differ significantly from those destined for hard-copy devices.²

The effectiveness of these and other techniques will be evaluated in future work. Empirical testing of the *Valhalla* interface is being undertaken to see if the user modelling techniques it encapsulates help users accomplish certain well-defined information retrieval tasks, as it is believed it will.

Empirical studies will also yield insight into the notion of perceptual salience advanced in this dissertation.

7.2.1 Learning: Updating Prior Probabilities

It is possible within the representation described in this thesis (see Section 5.4) to encode dependencies between random variables [161]. We could render explicit the relationship between sex and category, for instance, and continue to add other dependencies we consider important. The problem is that there may be many such relationships, and it will be difficult to decide which are important enough to en-

²The WIP system uses just this approach; combined with the intent-based paradigm, it is a promising direction. See also the work of Mackinlay [129].

code, and which can be safely ignored. Not only is this a knowledge engineering task which it is desirable to avoid, but the computational complexity of the resulting Bayesian network rises dramatically with the number of arcs representing dependence.

Instead, probability values can be obtained from an episodic knowledge base (EKB) [53] that tracks the user population over time. Values thus obtained will reflect *actual* relationships in the user population, with a computational complexity much lower than that for a complete Bayesian analysis, and the significant costs of error-prone knowledge engineering are completely avoided.

At the end of a session, the EKB is supplied with a new data point, a new individual; the information supplied is in fact the current, most likely user model, consisting of the highest-probability assumptions. This data point in the EKB can move over time, with continued experience with the same user.

When the reasoning agent requires a prior value, a call could be made to the EKB (as a procedural attachment, perhaps). $P(\textit{male})$ would be originally supplied by the EKB as simply the number of individuals in the EKB who are male, divided by the total number of individuals. Later, after the user has indicated that he or she is a faculty member, the EKB would be queried for the value of $P(\textit{male}|\textit{faculty})$. This value is the number of individuals in the EKB who are male faculty members, divided by the number of individuals who are faculty members.

The EKB could be used as a kind of “user model server,” a repository of information about the user population that can be queried by all applications serving that population. Multiple local EKBs could be located within an institution to serve different subsets of the complete user population, and global EKBs could serve queries over the entire population by querying all of the relevant local EKBs.

The EKB would function as a separate agent, communicating with other agents via TCP/IP. Craddock [53] describes an EKB that might be employed in the manner suggested here, and Orwant [154] [155] has recently implemented a system called

Doppelgänger that uses a similar mechanism.

Since the information in the EKBs is intended to be persistent, it is a design issue of some social impact whether the individuals stored in the EKB should be identifiable. If they are, then persistent models of individuals would result in highly accurate priors at the start of a new session with a user that had already used the system, or who had used another system served by the same EKB. If not, then the system must begin to model the user anonymously, from scratch, albeit with the help of a growing EKB. Individual tracking will keep the number of individuals stored in the EKB down to the number of actual users accessing the systems served by the EKB, while anonymous storage will result in a new entry in the EKB for every session by every user at every system served by the EKB. There are clearly significant advantages to individual tracking, but these must be weighed against its (still unclear) social repercussions (but see Section 7.2.2).

There is an unexplored relationship between the notion, well-entrenched in the user modelling literature, of *user stereotypes* [169] [45] [44] and the EKB approach to determination of prior probabilities. This relationship is hinted at by Rich: "...stereotypes can be viewed as concepts, and then they can be learned with statistical concept-learning methods." [171] The current discussion can be seen in just this light.

Once something is known about the user, this information can be used as in conventional stereotyping approaches to "trigger" the application of a stereotype. Say, for instance, that the user has indicated, or we have reasoned by plan recognition, that he or she is a faculty member. The traditional approach to the use of stereotypes would have us apply the 'faculty-member stereotype' and attribute thereby to the user the defaults therein. The approach described in this thesis computes using best-first abduction other consistent aspects of the user model as needed. The EKB approach would accomplish the same result, by finding the closest point in the feature space.

As with the approach used in this thesis, there is no need in the EKB approach to foresee all the possible combinations of triggers; the EKB can provide, on demand, a ‘stereotype’ for any conceivable combination of ‘triggers.’ Thus, we may encounter, in the course of interaction with the user population, an individual who is both a faculty member and a smoker, as well as an avid baseball fan. Although multiple inheritance from separate stereotypes for faculty, smoker and baseball can accommodate this particular example, considerable attention must still be given to preference for attribute values when these disagree between stereotypes. The EKB is immune from these problems, and provides values for any conjunction of attributes on demand.

A simpler approach

In this subsection a simpler approach is described that does not require the mechanism of an EKB, but which does not capture all the dependencies that may exist in a user population.

The knowledge engineer can “seed” the representation of prior probabilities. For instance, he or she can set the value of the random variable describing whether a user is a student or faculty member as follows: $\{\langle student, a/c \rangle, \langle faculty, b/c \rangle\}$; instead of storing priors as single real numbers, he can store pairs of numerators and denominators whose quotients are the priors. The value of the denominator can be interpreted as the size of the sample space from which the prior is determined, and the numerator as the number of positive instances found in that sample.

Over usage, additional instances are encountered, which can be seen as extending the sample space, and consequently having an effect on the values of the priors. A positive instance increments the numerator of the corresponding alternative, and the denominator of all alternatives.

There are obviously many pairs of numbers which, when divided, result in the same quotient. For instance, there are many a, b such that $a/b = 0.5$, but the mag-

nitudes of the numbers chosen to represent the priors influences their sensitivity to new information. The larger the denominator applied to the representation of a random variable, the more confidence implied, and the less sensitive it will be to observations. (See [201], *cf.* Dirichlet Priors.)

7.2.2 Privacy

The text of this thesis has consistently evaded the serious issue of privacy, with all of its complex social and ethical ramifications. Though outside the scope of the current work, any serious elaboration or broader application of the technology of user modelling will bring attention to the matter. Obvious questions include: *Who or what has access to the models? Are the models anonymous, or are they individual? Are they persistent, or discarded after each session with the system? Is the user aware of the modelling activity, and has he or she consented to such activity?*

The design of any user modelling system must necessarily answer these and other questions. In this research, the model is derived during the user's session, is available only to the reasoning engine and the user, and is discarded when the user terminates the session. All users of the system are made aware of the nature and scope of the modelling activity taking place, and they are, because of the scrutability desideratum advanced in Section 5.5, able to investigate the model itself via the user model window of the application interface.

As mentioned in Section 7.2, there is considerable advantage to maintaining persistent, individual models of users. If users return for multiple sessions, the modelling activity can begin where it left off at the end of the last session, thereby making the user's time at the interface more efficient, and more effective. Even anonymous models can have this salutary effect, to a lesser degree.

User models are likely to become important elements of future applications, as well as marketable commodities in themselves, very much like conventional mail-

ing lists are today. Today, it is polite and correct to ask people if they wish to be on a list before it is released and used to send them solicitations. Some people feel that their membership in a mailing list is to their advantage, others feel it is an intrusion into their private lives; this dichotomy will likely apply to the use of user models as well, but with greater potential, perhaps, for misuse.

Conceivably, models acquired over long periods could contain extremely detailed information about interests, habits and beliefs of individuals which, although valuable in the service of a cooperative application interface, could result in adverse effects ranging from unsolicited (though hopefully well-targeted) junk mail through to discrimination based on any of the many attributes of the model.

The aforementioned concerns are addressed by one of the objectives of the privacy protection policy enunciated by the privacy protection study commission [2], referred to as the principle of maximizing fairness. Government databases are subject to these and other guidelines, but no laws have yet been passed in North American courts to protect the rights of individual users of computer systems (see Rosenberg [174] for a broader view of privacy considerations in legal and social contexts).

Solutions to the problems created by the use of persistent, individual models could take many directions. Users can be given control over access to their models by having them specify which attributes or groups of attributes should be made available to which services and at what degree of authentication [58]. Users could maintain physical control of their models by carrying them on smart cards, PCMCIA cards (as in Doppelgänger [155]) or similar technology. Advanced encryption techniques with multiple levels of security may become accepted for networked transactions involving user models.

Privacy is an important issue that must be addressed not only from legal and ethical imperative, but because users themselves will be reluctant to use systems that do not treat sensitive personal data with due attention. Emerging secure tech-

nologies will hopefully be up to this task.

7.2.3 Future Development

A number of extensions to the hyperbrochure client are being planned. In addition to directly gaining knowledge via the user model window, details of the user's interaction with a presentation are provided to the reasoning engine. The reasoner does not currently use this information, but could bring resources to bear on it. For instance, if a user uses the navigation buttons in order to skip the viewing of the remainder of a particular clip, it may be deduced that the user doesn't have any interest in the contents of that clip and the user model can be updated accordingly (*cf.* plan recognition).

As was previously mentioned, items in the user model window are chosen based on their degree of sensitivity. Sensitivity analysis is defined (See Definition 9) as the act of determining how much a presentation will be changed when the user modifies a given set of assumptions in the user model. Given a set of user model items, sensitivity analysis currently provides a quantitative measure of these items, which is used by the client application to present the user model items in a reasonable, sorted order.

Other visual clues from the reasoning engine can also be considered. Linking the sensitivity metric with colour (for instance) might help to persuade the user to notice and correct faulty assumptions. As an example, the client application might choose to present all assumptions with a high sensitivity in bright red, thereby providing the suggestion of uncertainty or danger.

These perceptual salience metrics need to be determined by empirical study.

One of the goals of the intent-based authoring paradigm was to save time and effort by reducing the impact of the temporal nature of the video medium. However, viewers must still watch entire presentations in order to gauge their relevance and provide useful feedback to the reasoning engine. If the temporal portions of

presentations could be summarized in a non-temporal format, the viewer may be able to form an opinion of the presentation in a more timely manner. One possibility is to construct a graphical representation of the presentation and use a fisheye view (Noik [150]) of the representation in order to highlight the relevant features. If the user is in the process of browsing multiple presentations, the differences between presentations may be candidates for relevance.

The current system does not take into account what the user has already seen; there is no notion of “viewing history,” but something like it could be added. Users could then be given summaries of clips they had already seen, or shown relevant alternatives instead to avoid any form of repetition and boredom.

The system could be pressed into service in at least two different ways. First, after the fashion of information kiosks, *Valhalla* could be used on-line to provide information at varying levels of detail from overview to in-depth, in accordance with the dynamically evolving model of the information-seeking user. Second, after the fashion of a montage table, *Valhalla* can support a (human) video editor in the preparation of a video presentation that may be intended for a third party. These usage styles are distinguished for a number of reasons.

Information seekers will be limited in the time available, so the system must be real-time. The quality of the presentation is not paramount, as it will be seen only once by a single viewer. Response must be very good, however, if the typical kiosk user is to be expected to wait for the presentation, let alone watch it. In this usage pattern, the system is modelling the user-viewer.

Editors may be willing to wait for the system to search huge video indices and knowledge bases if the result is better content selection, or more consistent presentation style. The effort will pay off with multiple viewings by multiple viewers. In this usage pattern, the system is modelling the editor’s model of the eventual group of users.

Only the first of these ways of using the system is illustrated in this thesis with

a number of realistic usage scenarios.

A new Valhalla client is under development at the Department of Computer Science of the University of British Columbia. It is being implemented by Kurt Hoppe, a masters student there, as a CGI-compliant set of HTML files to be used in conjunction with industry standard World Wide Web browser programs like Netscape and Mosaic. The video server agent has been extended to interact with industry-standard http server programs; in particular, it has been tested with the NCSA 1.3 http daemon, and can deliver digital video to client applications via the daemon. An alternate reasoner is also under development by other researchers in the department, also for use with the Hyperbrochure.

Postscript

Content is everywhere. There is a widespread media-induced misconception that a lack of content lies behind current disenchantment with the world wide web, and that the remedy is in the hands of Hollywood conglomerates who will spew out incredible volumes of this content. What is in fact missing is a recognition of the primacy of intent. There are not enough authors in existence to tailor the presentation of all that content for individual readers. This is why intent-based authoring is pursued in this dissertation, and why the world wide web, which connects authors and readers as never before in human literacy, will one day become a delivery mechanism not for content, but for intent.

Bibliography

- [1] The Role of Intelligent Systems in the National Information Infrastructure. *Artificial Intelligence Magazine*, 16(3), 1995.
- [2] Public Law 93-579. Section s.2(b) of the privacy act of 1974, 5 united states code, section s.552a, 1974.
- [3] Robert M. Akscyn, Donald L. McCracken, and Elise A. Yoder. KMS: A Distributed Hypermedia System for Managing Knowledge in Organizations. *Communications of the Association for Computing Machinery*, 31(7):820–835, 1988.
- [4] Elisabeth André, Robin Cohen, Winfried Graf, Bob Kass, Cecile Paris, and Wolfgang Wahlster. Proceedings of the third international workshop on user modelling. Proceedings D-92-17, Deutsches Forschungszentrum für Künstliche Intelligenz, Stuhlsatzenhausweg 3, D-6600 Saarbrücken 11, Germany, August 1992.
- [5] Elisabeth André and Thomas Rist. Towards a plan-based synthesis of illustrated documents. Research Report RR-90-11, Deutsches Forschungszentrum für Künstliche Intelligenz, Stuhlsatzenhausweg 3, D-6600 Saarbrücken 11, Germany, September 1990.
- [6] Elisabeth André and Thomas Rist. The design of illustrated documents as a planning task. In M. Maybury, editor, *Intelligent Multimedia Interfaces*. AAAI Press, 1993.
- [7] J. André, R. Furuta, and Quint V., editors. The Cambridge Series on Electronic Publishing. Cambridge University Press, 1989.

- [8] Douglas E. Appelt and Martha E. Pollack. Weighted abduction for plan ascription. *User Modeling and User-Adapted Interaction*, 2(1-2):1–25, 1992.
- [9] Apple Computer, Inc., 20525 Mariani Ave., Cupertino, CA 95014. *HyperCard User's Guide*, 1987.
- [10] Yigal Arens and Eduard Hovy. How to describe what? towards a theory of modality utilization. In *Cognitive Science Society Meeting*, Cambridge, MA, August 1990.
- [11] Yigal Arens, Eduard Hovy, and Susanne van Mulken. Structure and Rules in Automated Multimedia Presentation Planning. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, pages 1253–1259, Chambéry, France, September 1993.
- [12] Paul Van Arragon. User modeling bibliography. Technical Report CS-87-22, Department of Computer Science, University of Waterloo, March 1987.
- [13] Paul Van Arragon. *Nested Default Reasoning for User Modelling*. PhD thesis, University of Waterloo, 1990.
- [14] J. L. Austin. *How to do Things with Words*. Oxford University Press, 1962.
- [15] Kent Bach and Robert M. Harnish. *Linguistic Communication and Speech Acts*. MIT Press, Cambridge, MA, 1979.
- [16] Afzal Ballim. *ViewFinder: A Framework for Representing, Ascribing and Maintaining Nested Beliefs of Interacting Agents*. PhD thesis, l'Université de Genève, 1992.
- [17] Afzal Ballim and Yorick Wilks. Beliefs, stereotypes and dynamic agent modelling. *User Modelling and User-Adapted Interaction*, 1(1):33–65, 1991.
- [18] Roland Barthes. The death of the author. In *Image, Music, Text*. Hill and Wang, New York, NY, 1977.
- [19] L. Bartram, R. Ovans, J. Dill, M. Dyck, A. Ho, and W.S. Havens. Contextual assistance in user interfaces to complex, time-critical systems: The intelligent zoom. In *GI '94: Graphics Interface 1994*, Banff, AL, Canada, May 1994.

- [20] M. Bauer, S. Biundo, D. Dengler, M. Hecking, J. Koehler, and G. Merziger. Integrated plan generation and recognition. Research Report RR-91-26, Deutsches Forschungszentrum für Künstliche Intelligenz, Stuhlsatzenhausweg 3, D-6600 Saarbrücken 11, Germany, August 1991.
- [21] Richard J. Beach. Setting tables and illustrations with style. CS 45, University of Waterloo Computer Science Department, Waterloo, Canada, May 1985. PhD thesis.
- [22] Izak Benbasat, Gerardine DeSanctis, and Barrie R. Nault. Empirical research in managerial support systems: A review and assessment, June 1991.
- [23] Izak Benbasat, Albert S. Dexter, and Peter Todd. The influence of color and graphical information presentation in a managerial decision simulation. *Human-Computer Interaction*, 2:65–92, 1986.
- [24] Jacques Bertin. *La Graphique et le Traitement Graphique de l'Information*. Flammarion, 1977.
- [25] Jacques Bertin. *Graphics and Information Processing*. Walter de Gruyter, 1981.
- [26] Jacques Bertin. *Semiology of Graphics: diagrams, networks, maps*. University of Wisconsin Press, 1983.
- [27] Sara Bly. Shared workspaces: A look at supporting distributed workgroups. CICSIR invited lecture, September 1991.
- [28] Sara A. Bly and Scott L. Minneman. Commune: A shared drawing surface. In *Proceedings of the Conference on Office Information Systems*, pages 184–192, Boston, MA, April 1990.
- [29] Kellogg S. Booth and W. Morven Gentleman. A symbiotic approach to visualization and the user interface. *Unpublished*, 1990.
- [30] G. Brajnik and C. Tasso. A flexible tool for developing user modeling applications with non-monotonic reasoning capabilities. In *Proceedings of the Third International Workshop on User Modelling*, pages 42–66, August 1992.

- [31] Giorgio Brajnik, Carlo Tasso, and Antonio Vaccher. *UMT User Manual – Version 1.0, September 1992*. Universita degli Studi di Udine, Laboratorio di Intelligenza Artificiale, Dipartimento di Matematica e Informatica, Via Zanon, 6, 33100 Udine, ITALY, 1992.
- [32] Gerhard Brewka. Preferred subtheories: An extended logical framework for logical reasoning. *IJCAI*, 1989.
- [33] Vannevar Bush. *Pieces of the Action*. William Morrow and Company, 1970.
- [34] W. Buxton and T. Moran. EuroPARC's Integrated Interactive Intermedia Facility (IIIF): Early Experiences. In S. Gibbs and A.A. Verrijn-Stuart, editors, *Multi-user Interfaces and Applications*, pages 11–34. 1990.
- [35] Sandra Carberry. Modelling the user's plans and goals. *Computational Linguistics*, 14(3):23, September 1988.
- [36] Sandra Carberry. *Plan Recognition in Natural Language Dialogue*. MIT Press (A Bradford Book), 1990.
- [37] T.A. Cargill. A view of source text for diversely configurable software. Technical Report CS-79-28, University of Waterloo Computer Science Department, 1979.
- [38] Stephen M. Casner. A task-analytic approach to the automated design of graphic presentations. *Association for Computing Machinery Transactions on Graphics*, 10(2), April 1991.
- [39] Timothy Catlin, Paulette Bush, and Nicole Yankelovich. Internote: Extending a hypermedia framework to support annotative collaboration. In *Hyper-text '89 Proceedings*, pages 365–378, 1989.
- [40] P. Cheeseman. On finding the most probable model. In J. Shriringer and P. Langley, editors, *Computational Models of Scientific Discovery and Theory Formation*, chapter 3, pages 73–95. Morgan Kaufmann, San Mateo, 1990.
- [41] P.P. Chen. The entity-relationship model: Toward a unified view of data. *ACM Transactions on Database Systems*, 1(1):9–36, 1976.

- [42] Mourad Cherfaoui and Christian Bertin. Video Documents : Towards Automatic Summaries. In *Workshop Proceedings of IEEE Visual Processing and Communications*, pages 295–298, Melbourne, Australia, September 1993.
- [43] H. Chernoff. The use of faces to represent points in k-dimensional space graphically. *Journal of the American Statistical Association*, 68:361–368, 1973.
- [44] David N. Chin. KNOME: Modeling What the User Knows in UC. In A. Kobsa and W. Wahlster, editors, *User Models in Dialog Systems*, pages 74–107. Springer-Verlag, 1989.
- [45] David Ngi Chin. *Intelligent Agents as a Basis for Natural Language Interfaces*. PhD thesis, Computer Science Division (EECS), University of California, Berkeley, CA 94720, jan 1988.
- [46] William S. Cleveland. *The Elements of Graphing Data*. Wadsworth and Brooks/Cole Advanced Books and Software, Pacific Grove, CA, 1985.
- [47] William S. Cleveland. A model for graphical perception, June 1990. Personal correspondence.
- [48] William S. Cleveland and Robert McGill. Graphical perception: theory, experimentation and application to the development of graphical methods. *American Statistical Association*, 79:531–554, September 1984.
- [49] E.F. Codd. A relational model of data for large shared data banks. *Communications of the Association for Computing Machinery*, 13(6):377–387, June 1970.
- [50] Robin Cohen and Marlene Jones. Incorporating User Models Into Educational Diagnosis. Technical Report CS–86–37, Faculty of Mathematics, University of Waterloo, Waterloo, Ontario, September 1986. Also published as Chapter 11 of *User Models in Dialog Systems*, edited by Kobsa and Wahlster (Springer-Verlag, 1989).
- [51] Jeff Conklin. Hypertext: An introduction and survey. In Irene Greif, editor, *Computer Supported Cooperative Work: A Book of Readings*, pages 423–475. Morgan Kaufman, 1988.

- [52] R. Cook and J. Kay. The Justified User Model. In *Proceedings of the Fourth International Conference on User Modelling*, pages 145–150, Hyannis, MA, August 1994.
- [53] A. Julian Craddock. *Induction and the Reference Class Problem*. PhD thesis, University of British Columbia, 1993.
- [54] A. Csinger, H. da Costa, and B. Forghani. A general-purpose programmable command decoder. In *IEEE Proceedings, Conference Compint*, pages 139–41, November 1987.
- [55] Andrew Csinger. From utterance to belief via presupposition: Default reasoning in user-modelling. In S. Ramani, R. Chandrasekar, and K.S.R. Anjaneyulu, editors, *Lecture Notes in Artificial Intelligence*, number 444, pages 407–417. Springer-Verlag, 1989.
- [56] Andrew Csinger. Implementing a theory of communications in a default reasoning framework. TR 91-30, Department of Computer Science, 1991.
- [57] Andrew Csinger. The Psychology of Visualization. Technical Report 28, Department of Computer Science, November 1992.
- [58] Andrew Csinger. OpenMed: Open Systems for Secure Health Care Information Transaction. A Joint CANARIE/TAD Proposal submitted by Cyberstore Systems Inc., CHARA Health Care Society, Lions Gate Hospital, Mission Memorial Hospital, InterSpect Systems Consulting Corp., Network Systems Group Inc., Health Informatics Research Group of UBC, September 1995.
- [59] Andrew Csinger and Kellogg S. Booth. Reasoning about Video: Knowledge-based Transcription and Presentation. In Jay F. Nunamaker and Ralph H. Sprague, editors, *27th Annual Hawaii International Conference on System Sciences*, volume III: Information Systems: Decision Support and Knowledge-based Systems, pages 599–608, Maui, HI, January 1994.
- [60] Andrew Csinger and David Poole. From utterance to belief via presupposition: Default reasoning in user-modelling. In *Proceedings of the Conference for Knowledge Based Computing Systems, KBCS-89*, pages 408–419, Bombay, India, December 1989. Also appeared in Springer-Verlag *Lecture Notes in Artificial Intelligence*, number 444, pages 407-417.

- [61] Andrew Csinger and David Poole. Hypothetically Speaking: Default Reasoning and Discourse Structure. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, pages 1179–1184, Chambéry, France, September 1993.
- [62] Glorianna Davenport, Ryan Evans, and Mark Halliday. Orchestrating Digital Micromovies. *Leonardo*, 26(4):283–288, 1993.
- [63] Glorianna Davenport, Thomas Aguiere Smith, and Natalio Pincever. Cinematic Primitives for Multimedia. *IEEE Computer Graphics and Applications*, pages 67–74, July 1991.
- [64] P. Edwards, editor. *The Encyclopaedia of Philosophy*. Macmillan and The Free Press, 1967.
- [65] Douglas C. Engelbart. Knowledge-domain interoperability and an open hyperdocument system. In *Proceedings of the Conference on Computer-Supported Cooperative Work, October 7-10, 1990 Los Angeles, CA*, pages 143–156, 1990.
- [66] James T. Enns. The promise of finding effective geometric codes. *Conference Presentation, SIGGRAPH'90*, 1990.
- [67] David W. Etherington. Formalizing non-monotonic reasoning systems. Technical Report 1, University of British Columbia, Vancouver, Canada, V6T 1W5, 1983.
- [68] Ryan George Evans. LogBoy Meets FilterGirl: A Toolkit for Multivariant Movies. Master's thesis, MIT, February 1994.
- [69] Steven Feiner. Apex: An experiment in the automated creation of pictorial explanations. *IEEE Computer Graphics and Applications*, 5(11), November 1985.
- [70] Steven Feiner. Research issues in generating graphical explanations. In *Proceedings Graphics Interface '85*, pages 117–122, 1985.
- [71] Steven Feiner and Clifford Beshers. Worlds within worlds: Metaphors for exploring n-dimensional worlds. In *Proceedings Symposium on User Interface Software and Technology*, Snowbird, UT, October 1990.

- [72] Steven Feiner, Sandor Nagy, and Andries van Dam. An experimental system for creating and presenting interactive graphical documents. *Association for Computing Machinery Transactions on Graphics*, 1(1):59–77, January 1982.
- [73] Steven K. Feiner and Kathleen R. McKeown. Coordinating text and graphics in explanation generation. In *Proceedings AAAI*, pages 442–449, Boston, MA, July 1990.
- [74] J. J. Finger and M. R. Genesereth. Residue: A Deductive Approach to Design Synthesis. Technical Report STAN-CS-85-1035, Department of Computer Science, Stanford University, Stanford, Cal., 1985.
- [75] T.W. Finin. GUMS: A General User Modelling Shell. In A. Kobsa and W. Wahlster, editors, *User Models in Dialog Systems*, pages 411–430. Springer, 1989.
- [76] Gerhard Fischer. Shared knowledge in cooperative problem-solving systems: Integrating adaptive and adaptable systems. In *Proceedings of the Third International Workshop on User Modelling*, pages 148–161, Dagstuhl, Germany, August 1992.
- [77] Gerhard Fischer, Raymond McCall, and Anders Morch. Janus: Integrating hypertext with a knowledge-based design environment. In *Hypertext '89 Proceedings*, pages 105–117, Pittsburgh, PA, November 1989.
- [78] Edward A. Fox. The coming revolution in interactive digital video. *Communications of the Association for Computing Machinery*, 32(7):794–801, July 1989.
- [79] Mark Friedell. Context-sensitive, graphic presentation of information. *ACM Computer Graphics*, 16(3):181–188, July 1982.
- [80] Mark Friedell. Automatic synthesis of graphical object descriptions. *ACM Computer Graphics*, 18(3):53–62, July 1984.
- [81] Hector Geffner. Default reasoning, minimality and coherence. In *KR89*, page 137, Toronto, Canada, May 1989.

- [82] Ricki Goldman-Segall. Thick Descriptions: A Tool for Designing Ethnographic Interactive Videodiscs. *SIGCHI Bulletin*, 21(2), 1989.
- [83] Ricki Goldman-Segall. A Multimedia Research Tool for Ethnographic Investigation. In I. Harel and S. Papert, editors, *Constructionism*. Ablex Publishing Corporation, Norwood, NJ, 1991.
- [84] Robert C. Goldstein, V. Srinivasan Rao, and Andrew W. Trice. Meeting-place: The customizable group support environment, 1993.
- [85] Bradley A. Goodman. Multimedia Explanations for Intelligent Training Systems. In Mark T. Maybury, editor, *Intelligent Multimedia Interfaces*, chapter 7, pages 148–171. AAAI Press – MIT Press, 1993.
- [86] Bradley A. Goodman and Diane J. Litman. On the interaction between plan recognition and intelligent interfaces. *User Modeling and User-Adapted Interaction*, 2(1-2):83–115, 1992.
- [87] Irene Greif, editor. *Computer Supported Cooperative Work*. Morgan-Kaufmann, 1988.
- [88] Steven Gribble, Andrew Csinger, and Kellogg S. Booth. A Distributed Multimedia Architecture for Intent-based Video Authoring and Presentation. In *Proceedings of MultiComm'94*, Vancouver, Canada, November 1994.
- [89] H.P. Grice. Logic and conversation. In P. Cole and J.L. Morgan, editors, *Syntax and Semantics: Speech Acts, vol 3*, pages 47–58. Academic Press, New York, 1975.
- [90] Georges Grinstein, Ronald Pickett, and Marian G. Williams. Exvis: An exploratory visualization environment. In *Graphics Interface '89*, London, Canada, 1989.
- [91] B.J. Grosz and C.L. Sidner. Attention, Intentions, and the Structure of Discourse. *Computational Linguistics*, 12(3):175–204, 1986.
- [92] Bernard J. Haan, Paul Kahn, Victor A. Riley, James H. Coombs, and Norman K. Meyrowitz. Hypermedia services. *Communications of the Association for Computing Machinery*, 35(1), January 1992.

- [93] Frank G. Halasz. Reflections on notecards: Seven issues for the next generation of hypermedia systems. *Communications of the Association for Computing Machinery*, 31(7):836–852, 1988.
- [94] Lynda Hardman, Guido van Rossum, and Dick C. A. Bulterman. Structured Multimedia Authoring. In *Proceedings ACM Multimedia 93*, pages 283–289, August 1993.
- [95] Beverly L. Harrison and Ronald M. Baecker. Designing Video Annotation and Analysis Systems. In *Graphics Interface '92 Proceedings*, pages 157–166, Vancouver, BC, May 1992.
- [96] Rich Helms. Distributed Knowledge Worker (DKW): A Personal Conferencing System. In *Proceedings of the 1991 CAS Conference*, pages 115–125, Toronto, Canada, October 1991. IBM Canada, Ltd. Laboratories, Center for Advanced Studies.
- [97] Tyson R. Henry and Scott E. Hudson. Interactive graph layout. In *Proceedings User Interface Software and Technology '91*. ACM Press, 1991.
- [98] Xueming Huang, Gordon I. McCalla, and Jim E. Greer. Student model revision: Evolution and revolution. In *Proceedings of the Eighth Biennial Conference of the Canadian Society for Computational Studies of Intelligence*, pages 98–105, 1990.
- [99] Hiroshi Ishii. Teamworkstation: Towards a seamless shared workspace. In *Proceedings of the Conference on Computer-Supported Cooperative Work, October 7-10, 1990 Los Angeles, CA*, pages 13–26, 1990.
- [100] Hiroshi Ishii and Naomi Miyake. Toward an open shared workspace: Computer and video fusion approach of teamworkstation. *Communications of the Association for Computing Machinery*, 34(12):37–50, December 1991.
- [101] P. N. Johnson-Laird. Mental models of meaning. In Joshi, Webber, and Sag, editors, *Elements of Discourse Structure*. Cambridge University Press, 1981.
- [102] Phillippe Joly and Mourad Cherfaoui. Survey of automatical tools for the content analysis of video. IRIT 93-36-R, Bibliotheque de l'IRIT, UPS, 118 route de Narbonne, 31062 TOULOUSE CEDEX, 1993. Also available

by anonymous FTP from ftp.irit.fr in PostScript, ascii and MS Word formats as private/svideo.[ps,as,wd], or by email direct from the authors (cherfaoui@ccett.fr or joly@irit.fr).

- [103] Marlene Jones and David Poole. An expert system for educational diagnosis based on default logic. In *Proceedings Expert Systems and their Applications*, pages 673–683, 1985.
- [104] William P. Jones. How do we distinguish the hyper from the hype in non-linear text? In *INTERACT'87*, pages 1107–1113, 1987.
- [105] Peter Karp and Steven Feiner. Issues in the automated generation of animated presentations. In *Proceedings Graphics Interface*, pages 39–48, Halifax, May 1990.
- [106] Peter Karp and Steven Feiner. Automated Presentation Planning of Animation Using Task Decomposition with Heuristic Planning. In *Graphics Interface '93*, pages 118–127, Toronto, Canada, May 1993.
- [107] Robert Kass and Tim Finin. Modelling the user in natural language systems. *Computational Linguistics*, 14(3):5, September 1988.
- [108] Robert Kass and Irene Stadnyk. Using user models to improve organizational communication. In *Proceedings of the Third International Workshop on User Modelling*, pages 135–147, Dagstuhl, Germany, August 1992.
- [109] Henry A. Kautz. A formal theory of plan recognition. Technical Report 215, Dept. of Comp. Sci., University of Rochester, Rochester, NY 14627, 1987.
- [110] A. Kay and A. Goldberg. Personal dynamic media. *IEEE Computer*, 10(3):31–42, March 1977.
- [111] J. Kay. um: A toolkit for user modelling. In *Proc. of the Second International Workshop on User Modelling*, pages 1–11, Honolulu HI, 1990.
- [112] D. Knuth. *The TeX Book*. Addison-Wesley, Reading, MA, 1984.
- [113] A. Kobsa. Modeling the user's conceptual knowledge in bgp-ms, a user modeling shell system. *Computational Intelligence*, 6:193–208, 1990.

- [114] A. Kobsa. Towards inferences in bgp-ms: Combining modal logic and partition hierarchies for user modeling. In *Proceedings of the Third International Workshop on User Modelling*, Dagstuhl, Germany, 1992.
- [115] Alfred Kobsa. User modelling: Recent work, prospects and hazards. In *Proceedings of the Workshop on User Adapted Interaction*, Bari, Italy, May 1992. Also available as a June 1992 Technical Report from Universität Konstanz Informationswissenschaft.
- [116] Alfred Kobsa and Wolfgang Wahlster, editors. *User Models in Dialog Systems*. Springer-Verlag, 1989.
- [117] Kurt Konolige. A computational theory of belief introspection. In *IJCAI85*, pages 502–508, 1985.
- [118] Robert R. Korfhage. Intelligent information retrieval: Issues in user modelling. Technical Report 85–CSE–9, Dept. of Computer Science and Engineering, Southern Methodist University, Dallas, Texas, May 1985.
- [119] C.W. Krueger. Models of reuse in software engineering. Technical Report CMU-CS-89-188, Carnegie Mellon U., December 1989.
- [120] David Kurlander and Steven Feiner. A visual language for browsing, undoing, and redoing graphical interface commands. In S. K. Chang, editor, *Visual Languages and Visual Programming*, pages 257–275. Plenum Press, New York, 1990.
- [121] L. Lamport. *A Document Preparation System. User's Guide And Reference Manual*. Addison-Wesley, Reading, MA, 1986.
- [122] Brenda Laurel. Issues in multimedia interface design: Media integration and interface agents. In *CHI'90 Proceedings*, pages 133–139, 1990.
- [123] Brenda Laurel. *Computers as Theatre*. Addison-Wesley, 1991.
- [124] Jintae Lee. Sibyl: A tool for managing group decision rationale. In *Proceedings of the Conference on Computer-Supported Cooperative Work, October 7-10, 1990 Los Angeles, CA*, pages 79–92, 1990.

- [125] Francis J. Lim and Izak Benbasat. A communication-based framework for group interfaces in computer-supported collaboration. In *24th Hawaii Conference on System Sciences*, Kauai, Hawaii, January 1991.
- [126] Lai-Huat Lim and Izak Benbasat. The effects of group support system on group meeting process and outcomes: A meta-analysis. UBC Working Paper 91-MIS-020, 1991.
- [127] W.E. MacKay and D.G. Tatar. Special issue on video as a research and design tool. *ACM SIGCHI Bulletin*, 21(2), October 1989.
- [128] Wendy E. Mackay and Glorianna Davenport. Virtual video editing in interactive multimedia applications. *Communications of the Association for Computing Machinery*, 32(7):802–810, July 1989.
- [129] Jock D. Mackinlay. Automatic Design of Graphical Presentations. Technical Report STAN-CS-86-1138, Stanford University Department of Computer Science, December 1986.
- [130] Jock D. Mackinlay. Automating the Design of Graphical Presentations of Relational Information. *Association for Computing Machinery Transactions on Graphics*, 5(2):110–141, April 1986.
- [131] Mitchell P. Marcus. *A Theory of Syntactic Recognition for Natural Language*. The MIT Press, 1980.
- [132] Joseph Marks and Ehud Reiter. Avoiding unwanted conversational implicatures in text and graphics. In *AAAI-90*, pages 450–456, Boston, MA, July 1990.
- [133] Catherine C. Marshall and Peggy M. Irish. Guided tours and on-line presentations: How authors make existing hypertext intelligible for readers. In *Hypertext '89 Proceedings*, pages 15–26, November 1989.
- [134] Christian Metz. *Film Language: A Semiotics of the Cinema*. Oxford University Press, 1974. Translated by Michael Taylor. .
- [135] Scott Minneman and Sara A. Bly. Managing à trois: a study of a multi-user drawing tool in distributed design work. In *CHI'91 Proceedings*, New Orleans, LA, 1991.

- [136] Fanya S. Montalvo. Knowledge visualizer: A graphic interface-building tool kit. Technical report, MIT Media Laboratory, 20 Ames Street, Cambridge MA 02139, September 1988.
- [137] Fanya S. Montalvo. Diagram understanding: The symbolic descriptions behind the scenes. In Tadao Ichikawa, Erland Jungert, and Robert R. Korfhage, editors, *Visual Languages and Applications*. Plenum Publishing Corporation, 1990.
- [138] J.D. Moore and C.L. Paris. Planning text for advisory dialogues. In *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics*, pages 203–211, 1989.
- [139] Johanna D. Moore and Cécile Paris. Exploiting User Feedback to Compensate for the Unreliability of User Models. *User Modeling and User-Adapted Interaction*, 2(4):287–330, 1992. .
- [140] Gerald M. Murch. Color graphics – blessing or ballyhoo? In Ronald M. Baecker and William A. S. Buxton, editors, *Readings in Human-computer Interaction: A Multidisciplinary Approach*, pages 333–341. Morgan Kaufmann, 1985.
- [141] Sung Myaeng and Robert Korfhage. Towards an intelligent and personalized information retrieval system. Technical Report 86–CSE–10, Dept. of Computer Science and Engineering, Southern Methodist University, Dallas, Texas, March 1986.
- [142] Nicholas Negroponte. *Soft Architecture Machines*. The MIT Press, Cambridge, Mass., 1975.
- [143] Daniel Neiman. Graphical animation from knowledge. In *AAAI '82*, pages 373–376, 1982.
- [144] U. Neisser. Decision time without reaction time: Experiments in visual scanning. *American Journal of Psychology*, 76:376–385, 1963.
- [145] Ted Nelson. A file structure for the complex, the changing, and the indeterminate. In *Proceedings of the ACM National Conference*, pages 84–100, 1965.

- [146] Steven R. Newcomb, Neill A. Kipp, and Victoria T. Newcomb. The “Hy-Time” Hypermedia/Time-based Document Structuring Language. *Communications of the Association for Computing Machinery*, 34(11):67–83, November 1991.
- [147] Jakob Nielsen. *Hypertext and Hypermedia*. Academic Press, Inc, San Diego, CA, 1989.
- [148] E.G. Noik. Automating the generation of interactive applications. Technical Report 90-21, U. of British Columbia, June 1990. Master’s Thesis.
- [149] E.G. Noik. Reducing the validation task by adding conformance at the implementation level. In *24th Annual Hawaiian Intl. Conf. on System Sciences*, Kauai, HI, January 1991.
- [150] Emanuel G. Noik. Layout-independent fisheye views of nested graphs. In *Proceedings IEEE/CS Symposium on Visual Languages*, Bergen, Norway, August 1993.
- [151] Donald A. Norman. *The Psychology of Everyday Things*. New York : Basic Books, 1988.
- [152] David Novitz. *Pictures and their Use in Communication*. Martinus Nijhoff, The Hague, 1977.
- [153] Margrethe H. Olson and Sara A. Bly. The portland experience: a report on a distributed research group. *International Journal of Man-Machine Studies*, 34:211–228, 1991.
- [154] Jon Orwant. Apprising the User of User Models: Doppelgänger’s Interface. In *Proceedings of the Fourth International Conference on User Modelling*, pages 151–156, Hyannis, MA, August 1994.
- [155] Jon Orwant. Heterogeneous Learning in the Doppelgänger User Modeling System. *User Modeling and User-Adapted Interaction*, 4(2):107–130, 1995.
- [156] Ronald Pickett. Integrated displays of multispectral imagery at air force geophysics laboratory. *Draft*, March 1991.

- [157] Ronald Pickett and Georges Grinstein. Iconographic displays for visualizing multidimensional data. In *Proceedings of the 1988 IEEE Conference on Systems, Man and Cybernetics*, pages 514–519, Beijing and Shenyang, People’s Republic of China, 1988.
- [158] David Poole. A logical framework for default reasoning. *Artificial Intelligence*, 36(1):27–47, 1987.
- [159] David Poole. Explanation and Prediction: an Architecture for Default and Abductive Reasoning. *Computational Intelligence*, 5(2):97–110, 1989.
- [160] David Poole. Normality and faults in logic-based diagnosis. In *IJCAI*, pages 1304–1310, Detroit, MI, August 1989.
- [161] David Poole. Probabilistic Horn Abduction and Bayesian Networks. *Artificial Intelligence*, 64:81–129, 1993.
- [162] David Poole, Randy Goebel, and Romas Aleliunas. A logical framework for default reasoning. In *The Knowledge Frontier: Essays in the Representation of Knowledge*, pages 331–352. Springer Verlag, New York, NY, 1987.
- [163] Bhavani Raskutti and Ingrid Zukerman. Query and Response Generation during Information- Seeking Interactions. In *Proceedings of the Fourth International Conference on User Modelling*, pages 25–30, Hyannis, MA, August 1994.
- [164] Susan E. Rathie. SAVANT: Video Annotation Support for Meeting Memory. Master’s thesis, University of British Columbia, 1995.
- [165] Brian K. Reid. *Scribe: A Document Specification Language and its Compiler*. PhD thesis, Carnegie-Mellon University, 1980. Also issued as Technical Report CMU-CS-81-100.
- [166] Brian K. Reid. Electronic Mail of Structured Documents: Representation, transmission, and archiving. In J. André, R. Furuta, and V. Quint, editors, *Structured Documents*. Cambridge University Press, 1989. .
- [167] R. Reiter. A logic for default reasoning. *Artificial Intelligence*, 13(1,2):81–132, 1980.

- [168] Ronald A. Rensink and Gregory Provan. The analysis of resource-limited vision systems. In *Proceedings of the 13th Annual Conference of the Cognitive Science Society*, pages 311–316, 1991.
- [169] Elaine Rich. User modelling via stereotypes. *Cognitive Science*, 3:329–354, 1979.
- [170] Elaine Rich. Users are individuals: Individualizing user models. *International Journal of Man-Machine Studies*, 18:199–214, 1983.
- [171] Elaine Rich. Stereotypes and User Modeling. In A. Kobsa and W. Wahlster, editors, *User Models in Dialog Systems*, pages 35–51. Springer, 1989.
- [172] Elaine Rich. *Artificial Intelligence*. McGraw Hill, 1991.
- [173] Thomas Rist and Elisabeth André. Incorporating graphics design and realization into the multimodal presentation system wip. In *Advanced Visual Interfaces 92*, Rome, 1992.
- [174] Richard Rosenberg. *The Social Impact of Computers*. Academic Press, 1992.
- [175] Steven F. Roth and Joe Mattis. Data Characterization for Intelligent Graphics Presentation. In *CHI'90 Proceedings*, pages 193–200, Seattle, WA, April 1990.
- [176] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, Englewood Cliffs, New Jersey 07632, 1995.
- [177] Sunil Sarin and Irene Greif. Computer-based real-time conferencing systems. *IEEE Computer*, 18(10):33–45, October 1985.
- [178] Margaret H. Sarner and Sandra Carberry. Tailoring definitions using a multifaceted user model. In *Proceedings of the Eighth Biennial Conference of the Canadian Society for Computational Studies of Intelligence*, pages 106–113, 1990.
- [179] Leonard J. Savage. *The Foundations of Statistics*. Dover Publications, Inc., New York, 1972.

- [180] John L. Schnase and John H. Leggett. Computational hypertext in biological modelling. In *Hypertext '89 Proceedings*, pages 181–197, Pittsburgh, PA, November 1989.
- [181] John R. Searle. A Taxonomy of Illocutionary Acts. In *Expression and Meaning*, pages 1–29. Cambridge University Press, London, 1979.
- [182] John R. Searle and Daniel Vanderveken. *Foundations of Illocutionary Logic*. Cambridge University Press, 1985.
- [183] Dorée Duncan Seligmann and Steven Feiner. Automated generation of intent-based 3d illustrations. *Computer Graphics*, 25(4):123–132, July 1991. Proceedings of SIGGRAPH '91 (Las Vegas, Nevada, July 28-August 2, 1991).
- [184] David Sewell. The usenet oracle: Virtual authors and network community, December 1992. To subscribe, send SUB EJRNL your name to LISTSERV@ALBANY.bitnet; to get contents or abstracts of previous issues send: GET EJRNL CONTENTS; to get Volume 1, Number 1, send GET EJRNL V1N1. EJournal is an all-electronic, Matrix distributed, peer-reviewed, academic periodical.
- [185] Stuart Smith, R. Daniel Bergeron, and Georges G. Grinstein. Stereophonic and surface sound generation for exploratory data analysis. In *CHI'90 Proceedings*, pages 125–132, April 1990.
- [186] Oliviero Stock. Natural Language and Exploration of an Information Space: the ALFresco Interactive System. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, 1991.
- [187] Oliviero Stock. ALFRESCO: Enjoying the Combination of Natural Language Processing and Hypermedia for Information Exploration. In Mark T. Maybury, editor, *Intelligent Multimedia Interfaces*, pages 197–224. AAAI Press/MIT Press, 1993.
- [188] L. Suchman and R. Trigg. Understanding Practice: Video as a Medium for Reflection and Design. In Greenbaum and Kyng, editors, *Design at Work: Cooperative Design of Computer Systems*, pages 210–213. 1991.

- [189] Markus A. Thies and Frank Berger. Intelligent user support in graphical user interfaces: Plan-based graphical help in object-oriented user-interfaces. Technical report, Deutsches Forschungszentrum für Künstliche Intelligenz, Stuhlsatzenhausweg 3, D-6600 Saarbrücken 11, Germany, March 1992.
- [190] Anne Treisman. Properties, parts, and objects. In Boff, Kaufmann, and Thomas, editors, *Handbook of Perception*, volume II, pages 35–1 to 35–70. 1986.
- [191] Anne Treisman. Features and objects in visual processing. In Irvin Rock, editor, *The Perceptual World: Readings from Scientific American*. W. H. Freeman and Company, New York, 1990. Originally November, 1986 Issue of Scientific American.
- [192] Anne Treisman, Patrick Cavanaugh, Burkhard Fischer, V.S. Ramachandran, and Rudiger von der Heydt. Form perception and attention: Striate cortex and beyond. In Lothar Spillmann and John S. Werner, editors, *Visual Perception: The Neurophysiological Foundations*, pages 273–316. Academic Press, 1990.
- [193] Anne Treisman and Stephen Gormican. Feature analysis in early vision: Evidence from search asymmetries. *Psychological Review*, 95:15–48, 1988.
- [194] Wolfgang Wahlster, Elisabeth André, Som Bandyopadhyay, Winfried Graf, and Thomas Rist. WIP: The Coordinated Generation of Multimodal Presentations from a Common Representation. Research Report RR-91-08, Deutsches Forschungszentrum für Künstliche Intelligenz, Stuhlsatzenhausweg 3, D-6600 Saarbrücken 11, Germany, February 1991.
- [195] Wolfgang Wahlster, Elisabeth André, Wolfgang Finkler, Hans-Jürgen Profitlich, and Thomas Rist. Plan-Based Integration of Natural Language and Graphics Generation. *Artificial Intelligence*, 63(1–2):387–427, October 1993. Also available from DFKI as a technical report. .
- [196] Wolfgang Wahlster and Alfred Kobsa. *User Models in Dialog Systems*. Springer-Verlag, 1990.
- [197] Colin Ware and John C. Beatty. Using color as a tool in discrete data analysis. CS 21, University of Waterloo Computer Science Department, Waterloo, Canada, August 1985.

- [198] Etienne Wenger. *Artificial Intelligence and Tutoring Systems: Computational and Cognitive Approaches to the Communication of Knowledge*. Morgan Kaufmann, Inc., 1987.
- [199] Robert Wilensky, David N. Chin, Marc Luria, James Martin, James Mayfield, and Dekai Wu. The berkeley unix consultant project. *Computational Linguistics*, 14(4):35–84, December 1988.
- [200] Dekai Wu. Active Acquisition of User Models: Implications for Decision-Theoretic Dialog Planning and Plan Recognition. *User Modeling and User-Adapted Interaction*, 1:149–172, 1991.
- [201] Yang Xiang, Michael P. Beddoes, and David Poole. Sequential Updating Conditional Probability in Bayesian Networks by Posterior Probability. In *Proceedings of the Eighth Biennial Conference of the Canadian Society for Computational Studies of Intelligence*, pages 21–27, 1990.
- [202] Nicole Yankelovich, Norman Meyrowitz, and Andries van Dam. Reading and writing the electronic book. *IEEE Computer*, 18(10):15–30, October 1985.
- [203] Elise Yoder, Robert Akscyn, and Donald McCracken. Collaboration in kms, a shared hypermedia system. In *CHI'89 Proceedings*, pages 37–42, May 1989.
- [204] Richard M. Young, T.R.G. Green, and Tony Simon. Programmable user models for predictive evaluation of interface designs. In *CHI'89 Proceedings*, pages 15–19, May 1989.
- [205] Frank Zdybel, Norton R. Greenfeld, Martin D. Yonke, and Jeff Gibbons. An information presentation system. In *IJCAI '81*, pages 978–984, 1981.
- [206] Polle T. Zellweger. Scripted documents: A hypermedia path mechanism. In *Hypertext '89 Proceedings*, pages 1–14, November 1989.
- [207] Ingrid Zukerman. Content planning based on a model of a user's beliefs and inferences. In *Proceedings of the Third International Workshop on User Modelling*, pages 162–173, Dagstuhl, Germany, August 1992.