AI Meets Authoring: User Models for Intelligent Multimedia

Andrew Csinger Kellogg S. Booth David Poole
csinger@cs.ubc.ca ksbooth@cs.ubc.ca poole@cs.ubc.ca

Department of Computer Science
University of British Columbia, Vancouver, Canada, V6T 1Z4

Abstract

Authoring is a complex, knowledge-intensive activity which until recently has been performed exclusively by humans. New computer-based techniques have added horsepower rather than intelligence to traditional approaches, and have not addressed their principal limitations, chief of which is the inability to tailor presentations to individual users at run-time.

We believe a model of the user is needed to support this kind of run-time determination of form *and* content. We describe our approach to the acquisition, representation and exploitation of user models: the *most plausible* user model is the result of an abductive *recognition* process and it incorporates assumptions about the user which are then used to constrain the *design* by abduction of the best presentation. Both recognition and design processes are performed at run-time. We describe a prototypical implementation designed to demonstrate these ideas in the domain of video authoring.

Our approach to authoring is intended to apply across multiple media; we have demonstrated these ideas with video because authoring in the video medium with traditional approaches inherits and exacerbates the problems from traditional media, and because the popularity of video as an authoring medium continues to grow.

Introduction

Authoring is the honorable tradition of collecting, structuring and presenting information in the form of a static "document" rendered in some medium or media. Promising new technologies have recently come to light that could alleviate some of the limitations of this difficult, knowledge-intensive undertaking.

As an offshoot of his semiological analyses of the cinema, Metz [1974, p45] wrote that "the spectatorial demand cannot mould the particular content of each film . . . ." Such statements—though accurate in 1974—are representative of now 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 stuck with since well before Gutenberg. 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.[1] We now have fast graphics, powerful reasoning engines and other technology, but what are we going to do with it?

In order to tailor presentations to the needs and desires of individual readers, we believe 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.

We are developing techniques for user modelling that we apply to the authoring problem. Thinking of authoring in terms of the knowledge required to support the activity has resulted in a new approach that

[1] We use "binding" here in the sense of associating values with variables. The pun was unintended.

we call "intent-based authoring," which we believe can ultimately resolve the principal problems with the traditional approach.

We apply our approach to video authoring in particular, because authoring in the video medium is even harder than in conventional media, and we think that our approach can eventually work across media boundaries.

We begin in this article with an overview of authoring, distinguishing between traditional and the proposed intent-based paradigms, situating video authoring in these contexts. The problems with traditional authoring manifest themselves in the video domain in both of the distinct phases of transcription and presentation. We focus in this article on how the form and content of the presentation can be determined at run-time by reference to a model of the user/reader, and we discuss the acquisition, exploitation and representation of this model; a more detailed exploration of the issues involved in transcription is given by Csinger and Booth [1994] who discuss problems associated with the processing of information before its presentation.

Finally, we describe *Valhalla,* a prototypical system implemented to demonstrate our ideas, and we provide our conclusions, drawn from our initial experiences using Valhalla.

1 Authoring

We first describe the traditional notion of authoring, independent of the target media for which the presentation is being designed, and then show how current approaches to video authoring lie within this traditional view, and thus suffer its drawbacks.

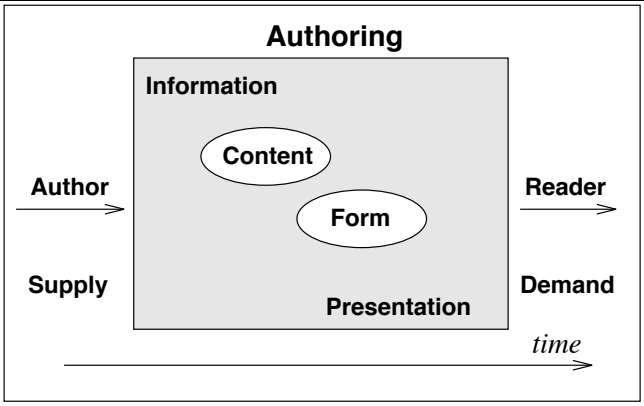

Figure 1: **The Traditional Approach to Authoring.** There is no clear separation of information from presentation, and authors are committed to both the form and content of their message. Structured-document approaches separate form and content, but user-tailored presentation is still not possible; reader "demand" only indirectly affects the authoring process.

1.1 Traditional Approaches

Acquiring and presenting information are knowledge-intensive activities that in the past have been performed exclusively by humans. These activities in combination have come to be known as *authoring*. The task of an author 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 (see Figure 1). The familiar book format conveys this point; 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. The author must both select and

order 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.

Hypermedia: Although tables of contents and elaborate indexes are intended as remedies to this static format, the burden of the "one size fits all" approach falls heavily upon the reader. For instance, an encyclopedia 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 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 paper.

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 [Newcomb *et al*., 1991] 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).

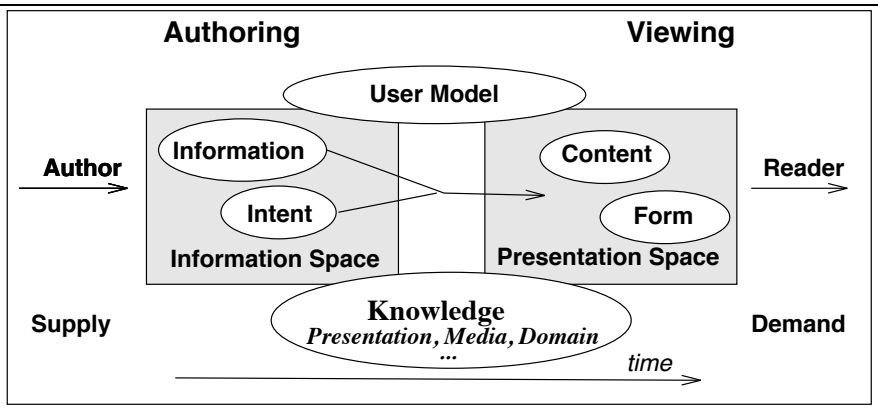

Figure 2: **The Intent-based Approach to Authoring.** Information and presentation spaces are clearly separated, bridged by various knowledge sources. In particular, a model of the viewer permits user-tailored determination of content at run-time; supply meets demand.

1.2 Intent-based Authoring

A more complete de-coupling of specification and presentation processes is required, however, before truly personalized presentations are possible. We argue that in addition to the content of the document, the author must also supply an *intent*. This authorial intent is usually implicit in the work; a newspaper article is (sometimes) written to inform, an editorial to convince, a journal article such as this to argue for the acceptance of a new authoring paradigm, and so on.

Making explicit this intention at the time the document is specified opens the door to truly user-specific document presentation. Illustrated in Figure 2, we call this approach to authoring *intent-based authoring*, and describe here an application of the approach to the authoring of video documents. MacKinlay [1986], Karp and Feiner [1993] 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 [Mackinlay, 1986, Roth and Mattis, 1990]. Semantic qualities of the data to be displayed are seldom considered.

The domain in which we are demonstrating our approach is the presentation of video data. Unlike Karp and Feiner [1993], 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 *presentation* for our purposes is an edit decision list which specifies the order in which a selection of video clips is to be played.[2]

Recently, other researchers have considered related problems. Hardman *et al.* [1993] 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 *et al.* [1993] also build presentations on-the-fly from canned video clips and other information. Our work reported in this article focusses on user modelling, rather than the media and domain concerns that motivate most other work.

1.3 Video Authoring

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. Unfortunately, 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" [Goldman-Segall, 1989]) 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.

On the other hand, the limitations of the traditional approach to authoring are most obvious when these processes are applied in non-traditional media, such as in the video medium. The raw material must first be acquired, which involves filming or digitizing. From this raw source, video authors must assemble cuts into a cohesive presentation. The raw footage can be very voluminous, and very sparse. 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 that may not be relevant to any conceivable presentation. Nevertheless, it takes an author at least ten hours 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 [1990], Goldman-Segall [1991], Harrison and Baecker [1992], MacKay and Tatar [1989], MacKay and Davenport [1989] and Suchman and Trigg [1991].)

When the author has finally identified a useful set of cuts, 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. In the case of video tape, it is a crippling limitation, due to the temporal linearity of the medium.

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

[2]Such a characterization deliberately excludes from discussion 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.

mechanisms for content-searching video data with the precision and semantic exactness of generalized textual search.[3]

Unlike video data, both graphics and animation can be searched for information using existing computer tools. Graphics usually rely on an underlying model or database that can be queried, while animations also have a model or database with temporal information added, which 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 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 hypervideo documents.

We 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 as low as the keystroke level and as high as the level of abstraction embedded into the GSS (*e.g.,* "brain-storming session," "open discussion," etc.) 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.

Two central problems relating to annotation are the *semantic unpredictability* and the *syntactic ambiguity* problems, defined as follows and further described by Csinger and Booth [1994].

> *The Semantic Unpredictability Problem is that unanticipated uses of the database may not be supported by the procedures used to acquire the database.*

The semantic unpredictability problem occurs when it is not known *a priori* what will turn out eventually to be important (*i.e.,* figuring out what events, and at what level of abstraction they should be recorded in the log). This problem is not solved by recording everything, nor by annotating every event in the record.

> *The Syntactic Ambiguity Problem is that the names which refer to elements of the information record may not be used consistently within the record, and may not be consistent with systems external to the record.*

The syntactic ambiguity problem applies to bodies of annotated video, and is not solved by adding a translation mechanism, or a lookup table; there is no necessary limit to the number of synonyms that would be required and the lookup table would need to be context sensitive.

This definition covers a variety of cases: incorrect or unintended *synonymy* is when there are multiple references to an individual (*e.g.,* 'Smith,' 'smith,' 'jsmith,' 'John' . . .). *Homonymy* is when it may be impossible to resolve a reference (*e.g.,* does 'smith' refer to John Smith or Mary Smith?). *Hypernymy* and *Hyponymy* are when references are at an inappropriate level in an abstraction hierarchy (*e.g.,* 'sports-event,' 'baseball-game,' 'montreal-vs-boston'. . .). An example of where these kinds of problems can arise is when a computer-generated log is used to index a video record which has also been manually annotated; the log may employ Unix user-id's to refer to individuals (*e.g.,* 'jsmith'), while the annotator may have used the more familiar 'John,' or 'Smith.' Some means of reconciling the references must be provided. Another example is where multiple annotators, or even a single annotator, use different labels to refer to the same event in the record.

Systems now available for video annotation do not address these problems. Each of the systems listed in the references to this article have their merits, but none of them deals directly with the syntactic ambiguity or the semantic uncertainty problem. Csinger and Booth [1994] describe how a knowledge based approach can at least mitigate some of the effects of these problems.

Even given a reasonable body of annotations, we believe navigation through a hypervideo document is more difficult than with other types of hypermedia. Pieces of information are difficult to extract from video

[3]See, however, Cherfaoui and Bertin [1993], who use digital image processing techniques to extract some types of information from video. Refer to Joly and Cherfaoui [1993] for a recent survey of related approaches.

because they often are meaningless when taken out of context (*i.e.,* it usually does not make sense to view a non-consecutive subset of frames nor does it make sense to view only disconnected pieces of frames). Thus there is a major bottleneck in the presentation of hypervideo documents. The goal of our research is to alleviate the difficulties associated with both the annotation and presentation phases of video authoring, within the context of the intent-based authoring paradigm described above.

Davenport *et al*. [1993] 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.

2 Intelligent User-tailored Presentation

Presentation is the process of transforming information into stimuli that map into the perceptual space of human beings [Csinger, 1992]. There are innumerable problems associated with automating presentation processes [Karp and Feiner, 1993, Mackinlay, 1986]. Basic issues include: 1) *What*: selecting the contents of the presentation, 2) *How*: determining the form and style of the presentation[4] and 3) *When*: deciding the temporal order of presentation events.

In the video medium, selecting the intervals of the record to be displayed, and the order in which they are to be displayed, are both serious problems.[5] Rendering the information in appropriate ways, relevant to the needs of individual users, will be achieved by recourse to models of these users, to knowledge about the media to be employed, and to other knowledge bases of various origin.

A rule-base of facts and assumables is supplied by a *knowledge engineer*. Some part of this database consists of intentional schema that intent-based *authors* can compose into high-level intentional representations. Other parts are world knowledge that the system uses in domain dependent and independent ways, and there can be knowledge as well that describes characteristics of the medium. The video record itself is provided by an *archiver*, and annotations that describe the contents of the video record are provided by an *annotator*. These databases, along with observations of the viewer's behavior, comprise the inputs to the intent-based authoring system.[6]

The outputs of the system are 1) an edit decision list of video clips which are played under user control on a video display device, and 2) a presentation to the user of the user model derived by the system. (Refer to Figure 3).

These processes, knowledge-bases and system outputs are described briefly in the following subsections.

2.1 User Models

Models of the viewers for whom the presentation is being prepared [Kobsa, 1992, Wahlster and Kobsa, 1990] are the most important knowledge-based ingredients in the recipe for intent-based presentation. Accordingly, our minimalist artificial intelligence approach [Poole, 1990] to user modelling is to first build the most likely user model, and then from this user model to prepare the best presentation. This is a *recognition* task followed by a *design* task.

---

[4]There is also the important issue of "allocating the media," [Arens *et al*., 1993], which involves hard-to-automate decisions about what medium in a multimedia environment best conveys the information value of a datum. Such decisions are very context-dependent, and are prone to unforeseen cross-modal effects [Csinger, 1994, Wahlster *et al*., 1991].

[5]Although this discussion proceeds in terms of the video medium, the intent-based authoring framework is more general and widely applicable. Presentation plans can be elaborated not only with video sequences but with graphics, text and whatever other media and information resources are at the disposal of the presentation system.

[6]Note that a single agent may assume more than one of the roles mentioned (knowledge engineer, author, archiver, annotator, viewer); a user agent can author his own presentations by specifying or choosing an intention, and then viewing the ensuing presentation. An agent may be both author and knowledge engineer, annotator or archiver, and so on, at different times.

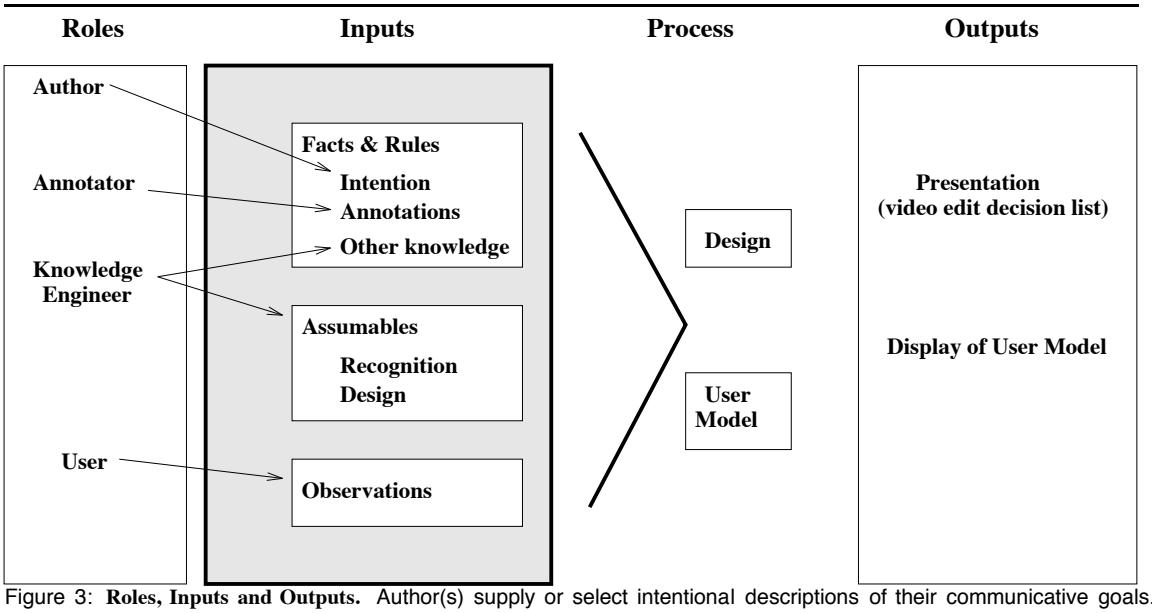

Figure 3: **Roles, Inputs and Outputs.** Author(s) supply or select intentional descriptions of their communicative goals. Knowledge engineers provide general and specific knowledge, as well as the assumables for model recognition and presentation design. The system calculates the most likely user model from observations of the user's activity and uses that to design the best presentation. Both the presentation and components of the user model are displayed to the viewer.

We use probabilities as the measure of likelihood of the user models, and use an instance of probabilistic Horn abduction to model the recognition task, finding the most likely explanation of the user's behavior in terms of a set of recognition assumptions which have associated probabilities.

We also need a way to perform the design task, and a way to evaluate designs (presentations): We use an abductive design approach [Finger and Genesereth, 1985] with costs associated with design assumptions.

These recognition and design processes can be combined into a single abductive reasoning component, where some assumptions are recognition assumptions (with probabilities) and some are design assumptions (with costs). This scheme leads to a preference ordering over explanations which is described in Section 2.1.3.

Kass and Finin [1988] and others have suggested various dimensions along which user models might be categorized. We discuss in the rest of this section how our user models are acquired, exploited and represented by the system.[7]

2.1.1 Acquisition

We regard the acquisition of a model of a user as a *recognition* task [Poole, 1989, Csinger and Poole, 1993], driven by the system's *observations* of the user's behavior at an interface, and by other information available in the user's run-time environment.

Acquisition techniques are often categorized by where they fall along a continuum from 'implicit' to 'explicit' in 'adaptive' and 'adaptable' systems, respectively [Fischer, 1992]. We employ a number of techniques that cover different parts of this spectrum.

At the explicit end of the continuum, we use a simple form-filling operation designed to elicit interest metrics. The system chooses *at run-time* both the fields and the layout of the form that is presented to the user, with the initial values determined by the system's current hypotheses about the user, which in turn are derived from the environment maintained by the resident operating system and other services (*e.g.*, `finger`, `gopher`, `.plan` file, etc.), before the user has taken any action. The user implicitly validates the

[7] Refer to Csinger and Poole [1993] for more on the minimal artificial intelligence approach to deriving models of users.

system's assumptions by his or her inaction, and explicitly corrects them by selecting new values or typing over the values displayed. Ideally, the system's hypotheses about the user are correct, and the user merely verifies them. In any case, the system makes 'observations' about the user consisting of what the user does when presented with a display of some of the hypotheses in the user model.[8]

The explicit presentation of the assumptions as a list of attribute values avoids complexities of natural language understanding and generation, and addresses the complaint that the operation of systems that employ user models is opaque to their users.

However, this approach is sensitive to the choice of elements of the user model which are shown to the user. Some algorithm to make this selection is needed, as there may be very many hypotheses in the system's theory about the user. We are experimenting with a number of sensitivity-analytic measures that are compatible with the representation we are using.

At the implicit end of the continuum, we use recognition techniques to infer the user's goals and plans from observation of his or her actions at a virtual VCR interface panel (described later). This approach has the virtue of being completely unintrusive, but is difficult to implement because of well-known problems having to do with the large plan-library for realistic problems.

In summary, we build the most plausible model of the user based upon all the available evidence, which includes explicit, direct feedback from the user via the user-model window, implicit feedback from the user's manipulation of the virtual VCR controls, and other contextual information like the user's login and group id's. We perform this recognition task using abductive reasoning techniques (see Section 2.1.3), where probability is taken to be the measure of plausibility; assumptions used for recognition have associated probabilities, and our algorithms find the most likely model that explains the observations.

2.1.2 Exploitation

The exploitation of the user model in the intent-based authoring domain can be seen as a design task, where the model is used to design a presentation that meets the intention specified by the author. The user model is a 'theory' with which the design must be consistent [Poole, 1989, Csinger and Poole, 1993]. Assumptions used for design have associated cost, and our algorithms minimize the cost of the design; the cost is a measure of the "badness" to the author of the design. We perform the design task by abduction.

The system tries to complete a constructive proof (from the user model, using design assumptions) that there exists an edit decision list that satisfies the intention of the author. When there are multiple proofs, the system selects the 'best' one, as described in Section 2.1.3.

For example, in the presentation of portions of a video record of a previously logged and annotated meeting, an important piece of information that might be found in the user model is the amount of time that a user is willing to devote to the presentation. This information will constrain the choice of cuts, and ensure that only the most important sequences are viewed.

The user model may not, of course, always be completely accurate, even after the user has been given the opportunity to modify selected fields in the acquisition phase. Dissatisfaction with the presentation may prompt the user to fast-forward, rewind, or simply stop the current presentation with the virtual VCR controls provided. The system can then reason about the user's goals and update the user model before designing a new presentation.

2.1.3 Representation

The contents of the knowledge-bases are expressed in a simple Horn-clause dialect [Poole, 1993b], and the reasoning engine is a best-first probabilistic Horn-clause assumption based system [Poole, 1993a].

Specifically, we use a variant of the Theorist framework for hypothetical reasoning [Poole, 1987], defined in terms of two sets of formulae: the "facts" F, and the *assumables* H. An *explanation* of a closed formula g is a consistent set $F \cup E$ that implies g, where E is a set of ground instances of elements of H, called *assumptions*. Such a g, the formula that is explained, is called the *explanandum*. When the reasoning

[8] The presentation of these hypotheses must be performed in perceptually salient ways. There is no point displaying this information with the aim of making things transparent for the user if the user is not going to understand it, or perhaps even worse, ignore it [Csinger, 1994].

system calculates explanations for an *explanandum* it regards as given, the system is performing *abduction*. We sometimes refer to the set E itself as the explanation of g.

This work extends Theorist to include both probabilities and costs (see also [Poole, 1993a]), and alters the notion of explanation to reflect a new combination of design and recognition.[9]). H is partitioned into the set R of assumables available for user model recognition, and into the set D available for presentation design. Every assumable r in R is assigned a prior probability $0 \leq P(r) \leq 1$. R is partitioned into disjoint and covering sets (which correspond to independent random variables as in Poole [1993b]). Every assumable d in D is assigned a positive cost $U(d)$.

A *model* of the user is a set of recognition assumptions $M : M \subset R, F \cup M \not\models \emptyset$ such that $F \cup M \models Obs$, where Obs is a set of observations of the user; in other words, M is an explanation of Obs. The probability of a user model is the product of the probabilities of its elements, assuming independence of recognition partitions: $P(M) = \prod_{r \in M} P(r)$. The best model is the one with the highest probability.

Given model M, a presentation *design* is a set of design assumptions $W : W \subset D, F \cup M \cup W \not\models \emptyset$ such that $F \cup M \cup W \models I(P)$, where $I(P)$ is a relation that is true when presentation P (a video edit decision list, for instance) satisfies the intention of the author. In other words, W, together with the model M, explains the existence of an edit decision list that satisfies the intention of the author; design W could be said to support presentation P in the context of model M. Note that here the user model M is treated as part of the facts for the design. The cost of a design is the sum of the costs of its constituent assumptions: $U(W) = \sum_{d \in W} U(d)$. The best presentation in the context of model M is the presentation supported by the least cost design.

Note that the partitioning of H partitions each explanation of $Obs \wedge \exists_P I(P)$ into a model and a design component which we denote as $\langle M, W \rangle$. We define a preference relation $\succ_p$ over explanations such that:

$$\langle M_1, W_1 \rangle \succ_p \langle M_2, W_2 \rangle$$

iff

$$(P(M_1) > P(M_2) \text{ or } (P(M_1) = P(M_2) \text{ and } U(W_1) < U(W_2)))$$

which 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.

In other words, an explanation is composed of a user model and a presentation design. We prefer the explanation whose recognition assumptions constitute the most plausible user model (the one with the highest probability), and whose design assumptions constitute the best presentation (the one with the lowest cost). Therefore, we prefer the explanation $\langle M, W \rangle$ with $\langle P(M), U(W) \rangle$ highest in a lexicographical ordering. Our algorithms find the explanation that represents the best design for the most plausible model.

A single abductive reasoning engine is employed for both recognition of the user model, and for design of the presentation. Separating the assumables for model recognition from the assumables for presentation design not only helps knowledge engineers express what they really mean, but has interesting ramifications in the way presentations are chosen; in particular, we do give up good models for which we can find only bad designs. For instance, consider the case where we have disjoint assumables *student* and *faculty*, where $P(student) >> P(Faculty)$, 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.*, $U(W_{best} | \cdots student \cdots) >> U(W_{best} | \cdots faculty \cdots)$. 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.

Decision-theory has been applied by others to design tasks under uncertainty. Some of this literature (see Cheeseman [1990] for a discussion) argues that the best design is the one that results from averaging over all models (probabilistically weighted), *i.e.*, that the expected value function is to be maximized to find the best presentation design:

[9]The distinction here between design and recognition turns on whether the system is free to choose any hypothesis that it wants (design) or whether it must try to "guess" some hypothesis that "nature" or an adversary has already chosen (recognition). Both recognition and design can be performed abductively or deductively; we use abduction here for both. See [Csinger and Poole, 1993] for details.

$$E(W) = \mu(W|M_1)P(M_1) + \mu(W|M_2)P(M_2) + \cdots + \mu(W|M_n)P(M_n) \quad (1)$$

where $\mu(P|M)$ is the cost of the best design that supports presentation P in the context of model M.

Our current research suggests that this approach may not always be the best one; in particular, in our application, we want the user model to be explicit, so that it can be critiqued by the user and by computational agents. We also desire visibility of the model so that the system's design rationale is evident to the user, who can correct the model if necessary. Using the expected value definition of utility shown in Equation 1 does not support these goals.

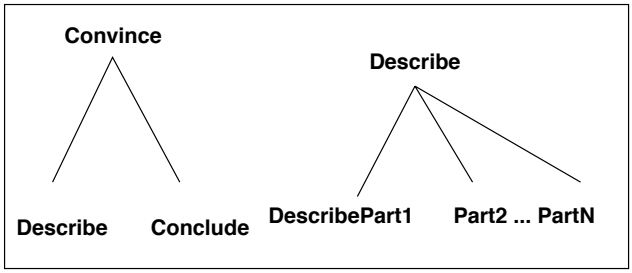

Figure 4: **Example Presentation Schemata.** The leaves of the schemata can be video edit-lists, or can themselves be further elaborated.

2.2 Domain-independent knowledge

We begin with a library of presentation plan schemata.[10] These schemata are variously elaborated, more or less domain-independent strategies or plans for delivering information in a variety of [stereo]typical scenarios. We could have, for instance, a "convince" plan, a plan intended to convince the viewer of a particular perspective, if this is deemed necessary. There are many ways in which viewers might become convinced of something; one strategy is to present examples as evidence in support of a conclusion. Stylistic or conventional factors might govern whether the evidence precedes the conclusion (prefix) or comes after it (postfix). We could also think of the "inform" plan, with its obvious speech-act connotations [Searle and Vanderveken, 1985]. A presentation plan is an arbitrarily complex argument structure designed to achieve a compound communicative goal.

A schema is chosen which is consistent with the model of the intention of the author. This schema must be refined, elaborated, and perhaps even modified non-trivially to arrive at the final presentation. Refinements take place by exploding the terminal leaves of the plan until they are instantiable components of the available (video) record. Structural changes to the tree can be made by extra-logical plan critics, or other generalized agents with an interest in the form or content of the presentation. Figure 4 illustrates two simple schemata, and Figure 5 shows some of the code used to implement the "describe" schema.

2.3 Domain-dependent knowledge

Here we refer to axioms supplied by a domain specialist, which describe the manner in which domain-independent elements may be instantiated with domain-specific types of information. So, for instance, where the schema indicates that the viewer should be impressed or entertained at a specific point in the presentation, this body of knowledge would define the kinds of information in this particular domain that might qualify. For example, impressing a viewer at some point would depend upon what it takes to impress

[10] This is similar in spirit to the presentation strategies employed in the WIP project at the German Research Center for Artificial Intelligence (DFKI) [Wahlster *et al*., 1991]. Some of our approach has its intellectual roots in the work being done at DFKI, where the first author was a visiting scholar in 1992.

```
% Descr is a description of Thing
describe(Thing, Descr) <=
   editList([],Descr,descr(Thing),0,_L).
describe(BigThing, Descr) <=
   bagof(Thing, partof(Thing, BigThing), Things),
   desc(Things, [], Descr, 0, _Length).

desc([], Descr, Descr, Length, Length).
desc([H|T], InD, Descr, InL, Length) <=
   editList(InD,OutD,descr(H),InL,OutLength),
   desc(T, OutD, Descr, OutLength, Length).
```

Figure 5: **Code sample: Part of the "Describe" Schema.** A description of a *Thing* can be a video edit-list, or it can be the descriptions of its parts, each of which can be a video edit-list. This code is interpreted by the Horn-clause meta-interpreter that performs the probabilistic and cost-based abductive reasoning described in this article. The reasoner itself is written in Prolog.

a user of a given type, as well as what the goals of the user are in consulting the presentation system. A set of axioms might permit the inference that a picture or a video of the latest results in computer-synthesized animation would impress a researcher in theoretical AI, although it might take more than that to impress the prospective graduate student in computer graphics.

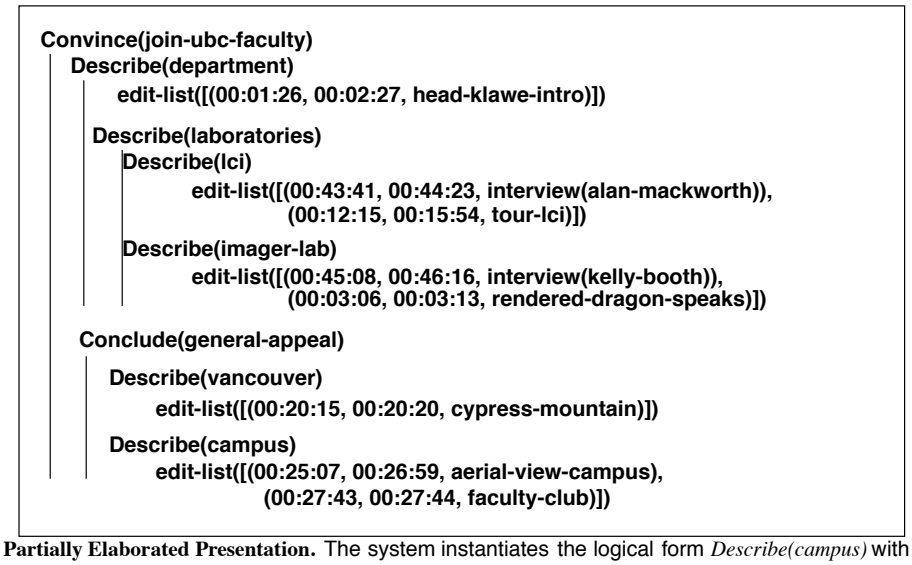

Figure 6: **A Partially Elaborated Presentation.** The system instantiates the logical form *Describe(campus)* with actual video footage represented by the in- and out-points given in absolute time codes.

Eventually, individual pieces of the video record must be chosen to fill the slots in the elaborated plan schema; the leaf nodes in the tree representing the plan schema for "convincing the user" are to be exploded (see Figure 6). We would be supplied with the knowledge required to retrieve instances of video that illustrate "graphics research results," and we would know at this point what label in the annotation label field identifies visuals of computer-synthesized graphics, and so on.

2.4 Media-specific knowledge

Knowledge of particular media characteristics must be encoded as well. It may not do, for instance, to present intervals whose length is less than, say, a second in duration, and there is little point in flashing video stills for a thirtieth of a second. There is a limit to the speed with which video can be meaningfully displayed to human viewers, and transitions between cuts should be pleasing and consistent. The presentation component of the system should know at least this much about video. Similar considerations apply for other media. Certain cross-modal issues can be handled with this knowledge-base as well [Csinger, 1994]. Synchronization of different tracks, for instance, is a difficult task. How are the audio and video tracks to be synchronized, when a tape is presented faster than normal speed? Beyond what speed do special measures need to be taken? The eventual answers to these kinds of questions will be included in the media-specific knowledge base.

3 *Valhalla*: A Prototype

Valhalla is a prototype implementation that addresses the problems associated with transcription [Csinger and Booth, 1994], and 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 document, along with the user model and other knowledge, to produce an edit-list. The viewer, even in the absence of the author, sees only relevant portions of the video.

```
? area(Student, graphics),                  % Student studies graphics
  supervises(FacultyMember, Student),       %  and 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 7: **Prolog query.** An author 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.

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. We restrict the user's interaction with the system 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 shown in this article, that can be used to make arbitrary queries of the reasoning engine; the user can take the role of intent-based author by specifying his own intent in this window and instructing the system to find an appropriate presentation. See Figure 7 for an example.

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 article. As discussed earlier, the user is given the opportunity to critique a

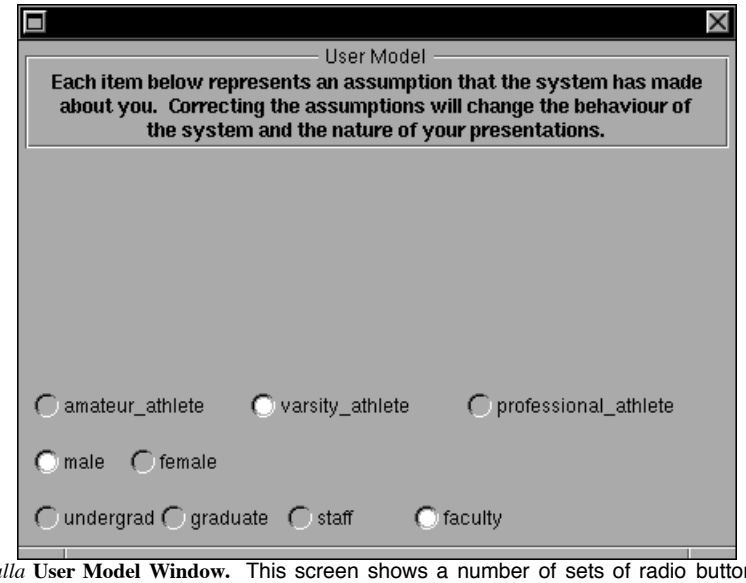

Figure 8: **The *Valhalla* User Model Window.** This screen shows a number of sets of radio buttons, which is *Valhalla*'s display technique for variables whose values are drawn from an exclusive (disjoint) set. Here, *Valhalla* believes the user is a faculty member; a student can correct the system's misconception with a single click.

selected subset of the model via *Valhalla*'s user-model window, shown in Figure 8. The hypotheses actually displayed to the user are context-dependent, selected and ranked by a sensitivity analysis algorithm (to be described in a forthcoming article) that approximates 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 are shown, for example, in bolder fonts, brighter colors, larger characters, and so on. 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* control window, shown in Figure 9, 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"). The "Show" button is a request that is passed on to the reasoning engine to calculate the next best presentation for the current model. "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.

The video delivery component of the system is designed to handle tape, video disk and digital video through a video server mechanism. Connections between the video server, user interface and reasoning engine are all client/server (TCP/IP) links, giving flexibility and platform independence; the user interface with form filling and virtual VCR control is implemented in Objective C and currently resides on a NeXT workstation, the Prolog reasoner as well as the video server on a Sparcstation. The knowledge bases are all written in a Prolog-like Horn clause language extended with assumptions (as described in Section 2.1.3), and the annotation database consists of only definite clause assertions.

We have begun testing the system with a body of video known as the UBC Computer Science Department Hyperbrochure, an hour-long video disk that includes 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. 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.

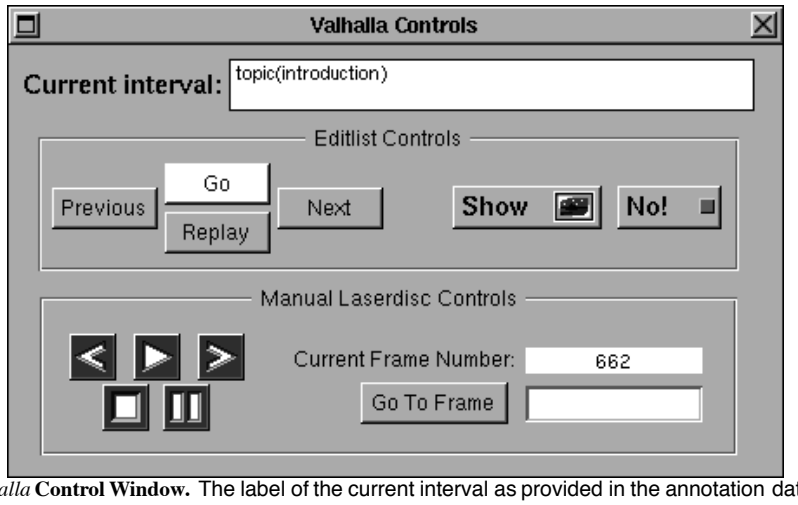

Figure 9: **The *Valhalla* Control Window.** The label of the current interval as provided in the annotation database is displayed. Manual laser disk controls include absolute frame indexing.

3.1 Scenario: Different Users, Different Models, Different Designs

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 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 a 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 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 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, information that 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 *are* other explanations, like "sheer disbelief," etc.) 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 accomodate Mary's optimal viewing time, as represented in the system's model of her.

Both John 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); the *a priori* probabilities of assumables can be upgraded according to well-known learning algorithms [Xiang *et al*., 1990]. 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.

4 Conclusion

Video annotation and presentation are characterized in this paper as members of a class of authoring tasks. Most systems which are currently available to support this task inherit the limitations of the traditional model of authoring. The foremost such limitation is that since the composition/specification and presentation phases are inseparable in the traditional model, there is no way 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 was proposed to mitigate the serious effects of these problems, and a prototype called *Valhalla* was implemented and tested.

The central focus of our project continues to be the deployment of artificial intelligence techniques for user modelling. We work within the limits of what has been called "minimal AI" to explore the simplest useful applications of probability and decision theoretic reasoning strategies to the problems of modelling users of computer systems. Our approach is very simple, based as it is upon well-tested notions from decision theory and the AI literature. We will be evaluating the effectiveness of these and other techniques in future work. We are undertaking empirical testing of the *Valhalla* interface to see if the user modelling techniques it encapsulates help users accomplish certain well-defined information retrieval tasks, as we believe it will.

The intent-based authoring paradigm described in this article can be applied to different media, domains, and tasks. We believe it has potential to circumvent limitations of the traditional model of authoring.

References

[Arens *et al*., 1993] 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.

[Buxton and Moran, 1990] 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.

[Cheeseman, 1990] P. Cheeseman. On finding the most probable model. In J. Shranger and P. Langley, editors, *Computational Models of Scientific Discovery and Theory Formation*, chapter 3, pages 73–95. Morgan Kaufmann, San Mateo, 1990.

[Cherfaoui and Bertin, 1993] 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.

[Csinger and Booth, 1994] 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.

[Csinger and Poole, 1993] 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.

[Csinger, 1992] Andrew Csinger. The Psychology of Visualization. Technical Report 28, The University of British Columbia, November 1992.

[Csinger, 1994] Andrew Csinger. Cross-modal Reference and the Attention Problem. In preparation., 1994.

[Davenport *et al*., 1993] Glorianna Davenport, Ryan Evans, and Mark Halliday. Orchestrating Digital Micromovies. *Leonardo*, 26(4):283–288, 1993.

[Finger and Genesereth, 1985] 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.

[Fischer, 1992] 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.

[Goldman-Segall, 1989] Ricki Goldman-Segall. Thick Descriptions: A Tool for Designing Ethnographic Interactive Videodiscs. *SigChi Bulletin*, 21(2), 1989.

[Goldman-Segall, 1991] Ricki Goldman-Segall. A Multimedia Research Tool for Ethnographic Investigation. In I. Harel and S. Papert, editors, *Constructionism*. Ablex Publishing Corporation, Norwood, NJ, 1991.

[Goodman, 1993] 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.

[Hardman *et al*., 1993] Lynda Hardman, Guido van Rossum, and Dick C. A. Bulterman. Structured Multimedia Authoring. In *Proceedings ACM Multimedia 93*, pages 283–289, August 1993.

[Harrison and Baecker, 1992] 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.

[Joly and Cherfaoui, 1993] 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).

[Karp and Feiner, 1993] 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.

[Kass and Finin, 1988] Robert Kass and Tim Finin. Modelling the user in natural language systems. *Computational Linguistics*, 14(3):5, September 1988.

[Kobsa, 1992] 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.

[Mackay and Davenport, 1989] 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.

[MacKay and Tatar, 1989] W.E. MacKay and D.G. Tatar. Special issue on video as a research and design tool. *ACM SIGCHI Bulletin*, 21(2), October 1989.

[Mackinlay, 1986] 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.

[Metz, 1974] Christian Metz. *Film Language: A Semiotics of the Cinema*. Oxford University Press, 1974. Translated by Michael Taylor. .

[Newcomb *et al*., 1991] Steven R. Newcomb, Neill A. Kipp, and Victoria T. Newcomb. The "HyTime" Hypermedia/Time-based Document Structuring Language. *Communications of the Association for Computing Machinery*, 34(11):67–83, November 1991.

[Poole, 1987] David Poole. A logical framework for default reasoning. *Artificial Intelligence*, 36(1):27–47, 1987.

[Poole, 1989] David Poole. Explanation and Prediction: an Architecture for Default and Abductive Reasoning. *Computational Intelligence*, 5(2):97–110, 1989.

[Poole, 1990] David Poole. Hypo-deductive Reasoning for Abduction, Default Reasoning and Design. In *Working Notes, AAAI Spring Symposium on Automated Abduction*, pages 106–110, March 1990.

[Poole, 1993a] David Poole. Logic Programming, Abduction and Probability: A Top-Down Anytime Algorithm for Computing Prior and Posterior Probabilities. *New Generation Computing*, 11(3–4):377–400, 1993.

[Poole, 1993b] David Poole. Probabilistic Horn abduction and Bayesian networks. *Artificial Intelligence*, 64(1):81–129, 1993.

[Roth and Mattis, 1990] Steven F. Roth and Joe Mattis. Data Characterization for Intelligent Graphics Presentation. In *CHI'90 Proceedings*, pages 193–200, Seattle, WA, April 1990.

[Searle and Vanderveken, 1985] John R. Searle and Daniel Vanderveken. *Foundations of Illocutionary Logic*. Cambridge University Press, 1985.

[Suchman and Trigg, 1991] 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*. 1991.

[Wahlster and Kobsa, 1990] Wolfgang Wahlster and Alfred Kobsa. *User Modelling in Dialog Systems*. Springer-Verlag, 1990.

[Wahlster *et al*., 1991] 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.

[Xiang *et al*., 1990] 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.