Coping with Temporal Constraints in Multimedia Presentation Planning

Elisabeth André and **Thomas Rist**
German Research Center for Artificial Intelligence (DFKI)
Stuhlsatzenhausweg 3, D-66123 Saarbrücken
Email: <name>@dfki.uni-sb.de

Abstract

Computer-based presentation systems enable the realization of effective and dynamic presentation styles that incorporate multiple media. Obvious examples are animated user interface agents which verbally comment on multimedia objects displayed on the screen while performing cross-media and cross-window pointing gestures. The design of such presentations must account for the temporal coordination of media output and the agent's behavior. In this paper we describe a new presentation system which not only creates the multimedia objects to be presented, but also generates a script for presenting the material to the user. In our system, this script is forwarded to an animated presentation agent running the presentation. The paper details the kernel of the system which is a component for planning temporally coordinated multimedia.

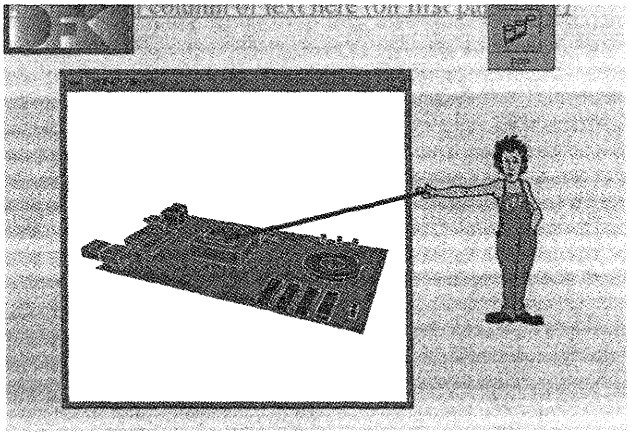

Figure 1: Verbal Annotation of Graphical Objects

Introduction

The success of human-human communication undisputably depends on the rhetorical and didactical skills of the speaker or presenter. Surprisingly enough, little attention has been paid to this aspect of computer-based presentation systems. Up to now, research has mainly focused on content selection and content encoding. Although multimedia documents synthesized by these systems might be coherent and even tailored to a user's specific needs, the presentation as a whole may fail because the generated material has not been presented in an appealing and intelligible way. This can often be observed in cases where multimedia output is distributed on several windows requiring the user to find out herself how to navigate through the presentation.

To enhance the effectivity of computer-based communication, we propose the use of a user interface agent which, in our case, appears as an animated character, the so-called PPP Persona. This character acts as a presenter, showing, explaining, and verbally commenting on textual and graphical output on a window-based interface. The use of such an animated agent to present multimedia material provides a good means of:

- establishing cross-references between presentation parts which are conveyed by different media possibly being displayed in different windows
- guiding a user through a presentation and thus preventing her from orientation and navigation problems and
- realizing new presentation styles that are dynamic and multimodal in nature.

For illustration, let's look at an example from one of our current application domains: the generation of instructions for the maintenance, service and repair of technical devices such as a modem. Suppose the PPP system is requested to explain the internal parts of a modem. A strategy to accomplish this task is to generate a picture showing the modem's circuit board and to introduce the names of the depicted objects. Unlike conventional static graphics where the naming is usually done by drawing text labels onto the graphics (often together with arrows pointing from the label to the object), the PPP Persona enables the realization of dynamic annotation forms as well. The system first creates a window showing the circuit board. After the window has appeared on the screen, the PPP Persona takes up a suitable position for carrying out pointing gestures. It points to the single objects one after the other and (cf. the screen shot shown in Fig. 1) utters the object names verbally (using a speech synthesizer). The advantage of this method over static annotations is that the system can influence the temporal order in which the user processes an illustration. Of course, it is also possible to combine this dynamic style with the standard annotation style; i.e. the PPP Persona attaches text labels to the depicted parts before the user's eyes.

The aim of this paper is to show (a) that such dynamic presentation styles can be handled in a common framework for describing the structure of multimedia presentations, and (b) that a plan-based approach can be used to design such presentations automatically - provided it is able to handle timing constraints.

The Structure of Dynamic Multimedia Presentations

In our previous work, we have developed principles for describing the structure of coherent text-picture combinations (cf. (André & Rist 1993)). Essentially, these principles are based on a generalization of speech act theory (Searle 1980) to the broader context of communication with multiple media, and an extension of RST (Rhetorical Structure Theory, (Mann & Thompson 1987)) to capture relations that occur not only between presentation parts realized within a particular medium but also those between parts conveyed by different media. The *rhetorical structure* can be represented by a directed acyclic graph (DAG) in which communicative acts appear as nodes and relations between acts are reflected by the graph structure. While the top of such a DAG is a more or less complex communicative act (e.g. to introduce an object), the lowest level is formed by specifications of elementary presentation tasks (e.g., pointing to an object). Coping with dynamic presentation styles as illustrated in the previous section requires an extension of this framework.

First of all, we have to address the *temporal structure* of a dynamic presentation as well. Like other authors in the Multimedia community, e.g. (Buchanan & Zellweger 1993; Hardman, Bulterman, & van Rossum 1994; Hirzalla, Falchuk, & Karmouch 1995), we start from an ordered set of discrete timepoints. The temporal structure is represented by timelines which position events along a single time axis where, in our case, an event corresponds to the start or the end of a communicative act.

Unlike systems where a human author has to specify the multimedia material and the timing constraints, we are concerned with the automated generation of the presentation parts, too. Therefore, we refine the notion of *communicative act* by explicitly distinguishing between *production* and *presentation acts*.[1] Whereas production acts refer to the creation of material, presentation acts are display acts, such as *S-Display-Text*, or acts which are carried out by the PPP Persona, e.g. *S-Point*. Let's consider the example shown in Fig. 1. The multimedia material was created by performing the following production acts:

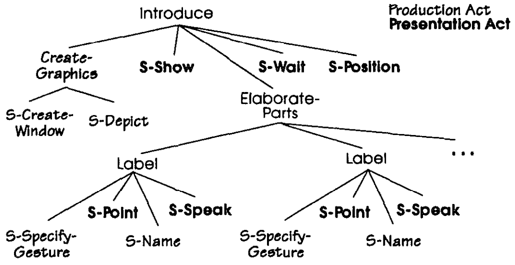

Figure 2: Rhetorical Structure

prod-act-1: Create a window containing a depiction of the circuit board
prod-act-2: Create a specification for a pointing gesture referring to the transformer
prod-act-3: Produce the sentence "This is the transformer"
...

To present the outcome of these production acts, the following presentation acts were carried out:

pres-act-1: Expose the created window
pres-act-2: Walk to the window
pres-act-3: Point to the transformer
pres-act-4: Say: "This is the transformer."
...
pres-act-13: Wait a moment

Fig. 2 exhibits the rhetorical structure of the sample presentation. The presentation as a whole serves to introduce the circuit board of a modem. It consists of the presentation acts *S-Show*, *S-Position* and *S-Wait* and the complex communicative acts *Create-Graphics* and *Elaborate-Parts*. *Create-Graphics* is composed of two production acts (*S-Create-Window* and *S-Depict*). *Elaborate-Parts* is defined by several labeling acts, which in turn are composed of two production acts (*S-Name* and *S-Specify-Gesture*) and two presentation acts (*S-Speak* and *S-Point*).

Now let's turn to the temporal structure of a multimedia presentation. Of course, to run the presentation in an intelligible way, the presentation acts need to be temporally coordinated. For example, pointing to an object depiction requires the exposure of the window containing the depiction. In addition, the pointing gesture should be maintained while the name of the respective object is uttered. Concerning the temporal relation between presentation acts and production acts, it is clear, that the presentation of material cannot start before its production. However, sometimes

[1] Note that this distinction provides a further dimension for characterizing communicative acts which has not been considered in previous classifications, e.g. (Maybury 1993b).

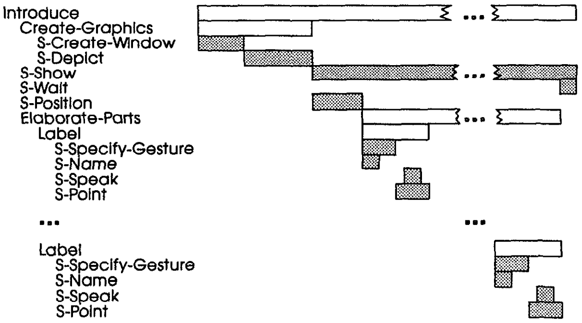

Figure 3: Temporal Structure

material is generated incrementally so that the presentation may start before the production is completed. For some purposes, it can also be reasonable to specify the temporal order in which a set of production acts should be carried out. For example, when generating a cross-media reference such as with *prod-act-2*, it wouldn't make much sense to start before the location (relative to the coordinate system of the display window) of the object depiction is known.

To sum up, there are usually many temporal constraints that must be satisfied by the communicative acts to produce and run a multimedia presentation. Fig. 3 shows a schedule for the communicative acts listed above. The durations of complex acts correspond to the length of the white bars, grey bars refer to durations of elementary acts.

While it is easy to characterize the temporal relations between communicative acts of a given presentation, it is much harder to do the same during the planning phase. The reason is that the temporal behavior of acts may be *unpredictable*. Among other things, it may depend on:

- *Resource limitations of the computing environment*
  For example, communicative acts such as *Create-Graphics* often involve the execution of programs with unpredictable runtimes. This may be aggravated by other factors such as network capacity and workload.
- *The temporal behavior of other acts*
  For instance, since we don't know when the graphics generation process will be finished, the startpoint of *S-Show*, which immediately comes after *Create-Graphics*, is unknown as well.
- *The current state of the presentation system*
  In PPP, the user can alter the system state at any time, e.g. by having the PPP Persona move to an arbitrary position on the screen. Thus, we cannot anticipate where the Persona will stand at presentation time. This makes it impossible to predict how long the Persona needs to walk to an appropriate position.

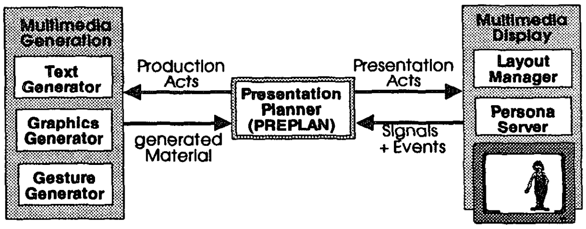

Figure 4: The PPP System

In the next three sections, we will describe how presentation schedules can be automatically built up starting from possibly incomplete timing information.

A View on PPP'S Architecture

PPP's major components are: a *presentation planner*, *medium-specific generators*, currently for graphics (cf. (Rist & André 1992)), text (cf. (Kilger 1994)) and gestures, the *Persona Server* and a *constraint-based layout manager* (cf. (Graf & Neurohr 1995)). The presentation planner (cf. (André & Rist 1993)) is responsible for determining the contents of multimedia material, selecting an appropriate medium combination and designing a script for presenting the material. Elementary production acts are sent to the corresponding generators which, in turn, inform the presentation planner when they have accomplished their task and how they have encoded a certain piece of information. The results of the generators are taken as input for the design of the presentation script which is forwarded to the display components for execution. The task of the layout manager is the determination of effective screen layouts and the maintenance of user interactions. The Persona Server (cf. (André, Müller, & Rist 1996)) carries out Persona actions which, among other things, includes assembling appropriate animation sequences. Both display components signal when they have accomplished their tasks and inform the presentation planner about the occurrence of interaction events, such as mouse-clicks on windows.

Representation of Design Knowledge

In order to build up multimedia presentations, we have defined a set of presentation strategies that can be selected and combined according to a particular task. These strategies reflect general design knowledge or they embody more specific knowledge of how to present a certain subject. They are characterized by a *header*, a set of *applicability conditions*, a collection of *inferior acts*[2], a list of *qualitative* and *metric* temporal constraints and a *start* and an *end interval*. The header corresponds to a complex presentation act. The applicability conditions specify when a strategy may be used

[2] For the sake of simplicity, we don't distinguish between main and subsidiary acts as we did in our previous work.

and constrain the variables to be instantiated. The inferior acts provide a decomposition of the header into more elementary presentation acts. Qualitative temporal constraints are represented in an "Allen-style" fashion which allows for the specification of thirteen temporal relationships between two named intervals: *before*, *meets*, *overlaps*, *during*, *starts*, *finishes*, *equal* and inverses of the first six relationships (cf. (Allen 1983)). Allen's representation also permits the expression of disjunctions, such as *(A (before after) B)*, which means that A occurs before or after B. Metric constraints appear as difference (in)equalities on the endpoints of named intervals. They can constrain the duration of an interval (e.g., $(10 \leq Dur\ A2 \leq 40)$), the elapsed time between intervals (e.g., $(4 < End\ A1 - Start\ A2 < 6)$) and the endpoints of an interval (e.g., $(Start\ A2 \leq 6)$).

Examples of presentation strategies are listed below. The first strategy may be used to build up the presentation shown in Fig. 1. It only applies if the system believes that *?object* is a physical object. Besides acts for the creation of graphics and natural language expressions, the strategy also comprises presentation acts to be executed by the PPP Persona, such as (*S-Show*, *S-Position* and *S-Wait*). Note that we are not forced to completely specify the temporal behavior of all production and presentation acts at definition time. This enables us to handle acts with *unpredictable* durations, start and endpoints, i.e. acts whose temporal behavior can only be determined by executing them. For example, in (S1) we only specify a minimal duration for act A2 and a fixed duration for act A4.

```
(S1) Header: (Introduce S U ?object ?window)
     Applicability Conditions:
     (Bel S (ISA ?object Physical-Object))
     Inferiors:
     ((A1 (Create-Graphics S U ?object ?window))
      (A2 (S-Show S U ?window))
      (A3 (S-Position S U)) (A4 (S-Wait S U))
      (A5 (Elaborate-Parts S U ?object ?window)))
     Qualitative:
     ((A1 (meets) A2) (A3 (starts) A2) (A3 (meets) A5)
      (A5 (meets) A4) (A4 (finishes) A2))
     Metric: ((10 ≤ Dur A2) (2 ≤ Dur A4 ≤ 2))
     Start: A1
     Finish: A2
```

To enable iteration over finite domains, we rely on a *Forall construct*, which is used in Strategy (S2) to indicate for all parts *?part* of an object *?object* one after the other to which class they belong.

```
(S2) Header: (Elaborate-Parts S U ?object ?window)
     Applicability Conditions:
     (Bel S (Encodes ?pic-part ?part ?window))
     Inferiors:
     ((A1 (Forall ?part With
                 (*And* (Bel S (Part-of ?part ?object))
                        (Bel S (I-ISA ?part ?class)))
                 Do Sequentially
                 (Ai (Label S U ?part ?class ?window)))))
```

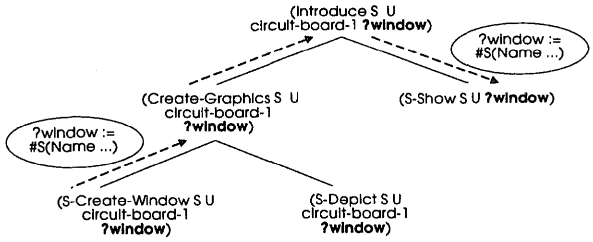

Figure 5: Propagating Multimedia Objects

Planning of Communicative Acts

To produce and present multimedia material, the strategies introduced above are considered operators of a planning system (cf. (André & Rist 1993)). Starting from a presentation goal, the presentation planner searches for plan operators and builds up a refinement-style plan in the form of a DAG. In the example shown Fig. 1, the system had to accomplish the task: *(Introduce S U circuit-board-1 ?window)*, selected strategy (S1) and set up the following subgoals: *(Create-Graphics S U circuit-board-1 ?window)*, *(S-Show S U ?window)*, *(S-Position S U)*, *(Elaborate-Parts S U circuit-board-1 ?window)* and *(S-Wait S U)*.

Whereas *S-Show*, *S-Position*, and *S-Wait* are elementary acts, *Create-Graphics* and *Elaborate-Parts* are further expanded by the presentation planner. The refinement of *Elaborate-Parts* results in several pointing gestures and speech acts. The speech acts are forwarded to the text generator which generates a name for each object to be introduced. The textual output is uttered using a speech synthesizer. Gestures are specified by the gesture generator which determines the gesture type (in our case pointing with a stick) and the exact coordinates. The expansion of *Create-Graphics* leads to a call of the graphics generator, and the creation of a window in which the resulting depiction of the modem's circuit board will appear.

During the planning process, multimedia objects are built up by performing production acts. These multimedia objects are bound to variables which are propagated in the DAG. In Fig. 5, the variable *?window* is instantiated with a window structure by *S-Create-Window*. Performing the act *S-Depict* causes an update of the data strucure bound to *?window*. The propagation mechanism ensures that the new value is accessible when executing the act *S-Show* which leads to the exposure of the window. To enable the processing of temporal constraints, we have combined PPP's presentation planner PREPLAN with RAT (cf. Fig. 6). RAT is a system for representing and reasoning about actions and plans, which relies on an extended version of Kautz's and Ladkin's MATS system (Kautz & Ladkin 1991). For each node of the presentation plan, the planner creates a local constraint network which includes the temporal constraints of the corresponding plan operators. During the planning process,

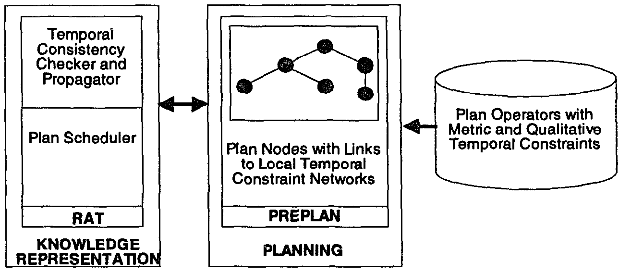
Temporal Consistency Checker and Propagator
Plan Scheduler
RAT
KNOWLEDGE REPRESENTATION
Plan Nodes with Links to Local Temporal Constraint Networks
PREPLAN
PLANNING
Plan Operators with Metric and Qualitative Temporal Constraints

Figure 6: Combining the Presentation Planner with a Temporal Description Logics

RAT checks these local constraint networks for consistency and computes numeric ranges on each endpoint and difference of endpoints and possible Allen relationships between each pair of intervals. In case of local inconsistencies, another presentation strategy is tried out. After the completion of the presentation planning process, a global temporal constraint network is built up by propagating the constraints associated with each planning node top-down and bottom-up in the DAG. If no global consistent temporal network can be built up, the presentation fails. Finally, a schedule is built up by resolving all disjoint temporal relationships between intervals and computing a total temporal order. For example, the following schedules would be created for a network containing the constraints *(A (before after) B)*, *(A (equals) C)*, $1 \leq$ Dur A ≤ 1 and $1 \leq$ Dur B ≤ 1:

Schedule 1	**Schedule 2**
1: Start A, Start C	*1: Start B*
2: End A, End C	*2: End B*
3: Start B	*3: Start A, Start C*
4: End B	*4: End A, End C*

As mentioned earlier, the planning is aggravated by acts with an unpredictable temporal behavior. Therefore, RAT only builds up a partial schedule which has to be refined when running the presentation. That is for some communicative acts, RAT only indicates an interval within which they *may* start or end instead of prescribing an exact timepoint.

The temporal behavior of a presentation is controlled by a *presentation clock* which is set to 0 when the system starts to show the planned material to the user and incremented by the length of one time unit[3] until the presentation stops. For each timepoint, RAT indicates which events must or can take place. For instance, a communicative act whose starting point is between 0 and 2, *may* start at timepoint 0 or 1, but *must* start at timepoint 2 in case it has not yet started earlier. Whether the event actually takes place or not is decided by the PPP Persona. Currently, the Persona chooses the earliest possible timepoint. In order to satisfy the temporal constraints set up by RAT, the Persona may have to shorten a presentation, to skip parts of it or to make a pause. In some cases, this may lead to suboptimal presentations, e.g. if the Persona stops speaking in the midst of a sentence. As soon as the Persona has determined that a certain event should take place, RAT is notified of this decision because it may have influence on further events. RAT adds a new metric constraint to the global temporal constraint network and refines the schedule accordingly.

Let's assume that RAT informs the Persona that the creation of the window may be finished at timepoint 1 and that the Persona may show the window to the user. However, it turns out that 10 seconds[4] are required for the creation of the window. The Persona forwards this information to RAT, which adds $10 \leq$ Dur Create-Window ≤ 10 to the global temporal constraint network. Since Create-Window meets S-Show, the display of the window can start only at timepoint 10.

Related Work

Efforts to develop time models for multimedia documents have been made by (Buchanan & Zellweger 1993; Hardman, Bulterman, & van Rossum 1994; Hirzalla, Falchuk, & Karmouch 1995). But, in all approaches the editing of a multimedia document is carried out by a human author who also has to specify the desired temporal relationships between the single document segments from which a consistent schedule is computed. Since there is no explicit representation of the contents of a document, it's not possible to automatically determine a high-level temporal specification for a document. In contrast to this, our system is not only able to design multimedia material, but also plans presentation acts and their temporal coordination.

A first attempt ot incorporate time into an automated presentation system has been made by Feiner and colleagues (cf. (Feiner *et al.* 1993)). However, they only investigate how temporal information can be conveyed by dynamic media and don't present a mechanism for synchronizing them.

A second research area which is of interest for our work is the creation of lifelike characters (see e.g. (Takeuchi & Nagao 1993; Badler, Phillips, & Webber 1993; Kurlander & Ling 1995; Lashkari, Metral, & Maes 1994)). The work closest to our own is that being carried out by Microsoft Research in the Persona project (cf. (Kurlander & Ling 1995)). In the current prototype system, a parrot called Peedy acts as a conversational assistant who accepts user requests for audio CDs. In contrast to PPP, the presentation of material in their system is restricted to playing the selected CDs.

[3] The length of a time unit can be interactively changed by the user.

[4] Since we rely on a time model with discrete timepoints, the actually needed time has to be rounded.

Conclusion

In this paper, we have presented a plan-based approach for the automatic creation of dynamic multimedia presentations. The novelty of PPP is that it not only designs multimedia material, but also plans presentation acts and their temporal coordination. This has been achieved by combining a presentation planning component with a module for temporal reasoning. To cope with unpredictable temporal behavior, we first build up a partial schedule while planning the contents and the form of a presentation which is refined when running it.

Our approach is particularly suitable for planning the behavior of animated user interface agents which have the potential of becoming integral parts of future intelligent presentation systems. However, it is not restricted to this class of applications. For instance, it can also be used for timing the display of static graphics and written text. In all existing presentation systems, document parts are either shown to the user immediately after their production (incremental mode) or the systems wait until the production process is completed and then present all the material at once (batch-mode). However, these systems are not able to flexibly change the presentation order depending on the current situation. In contrast, our approach makes it possible to influence the order and the speed in which a user processes a document by explicitly specifying the time at which information should be shown. Furthermore, multimedia material along with timing information specified by PPP can be used as input for existing presentation engines, such as the CMIFed environment (Hardman, Bulterman, & van Rossum 1994).

Future work will concentrate on more complex user interactions. Currently, the system clock is stopped when the user interrupts a presentation and started again when he resumes it. However, we also want to explicitly represent temporal relationships between presentation acts and interaction acts. For example, clicking on menu items is only possible as long as the menu is visible.

Acknowledgments

This work has been supported by the BMBF under grant ITW 9400. We would like to thank H.-J. Profitlich and M. Metzler for the development of RAT and J. Müller for his work on the Persona server and the overall system integration.

References

Allen, J. F. 1983. Maintaining Knowledge about Temporal Intervals. *Communications of the ACM* 26(11):832–843.

André, E., and Rist, T. 1993. The Design of Illustrated Documents as a Planning Task. In Maybury (1993a). 94–116.

André, E.; Müller, J.; and Rist, T. 1996. The PPP Persona: A Multipurpose Animated Presentation Agent. In *Advanced Visual Interfaces.* ACM Press.

Badler, N.; Phillips, C.; and Webber, B. 1993. *Simulating Humans: Computer Graphics, Animation and Control.* New York, Oxford: Oxford University Press.

Buchanan, M., and Zellweger, P. 1993. Automatically Generating Consistent Schedules for Multimedia Documents. *Multimedia Systems* 1:55–67.

Feiner, S. K.; Litman, D. J.; McKeown, K. R.; and Passonneau, R. J. 1993. Towards Coordinated Temporal Multimedia Presentations. In Maybury (1993a). 139–147.

Graf, W. H., and Neurohr, S. 1995. Constraint-based layout in visual program design. In *Proc. of the 11th International IEEE Symposium on Visual Languages.*

Hardman, L.; Bulterman, D.; and van Rossum, G. 1994. The Amsterdam Hypermedia Model: Adding Time and Context to the Dexter Model. *Communications of the ACM* 37(2):50–62.

Hirzalla, N.; Falchuk, B.; and Karmouch, A. 1995. A Temporal Model for Interactive Multimedia Scenarios. *IEEE Multimedia* 2(3):24–31.

Kautz, H. A., and Ladkin, P. B. 1991. Integrating metric and qualitative temporal reasoning. In *Proc. of AAAI-91*, 241–246.

Kilger, A. 1994. Using UTAGs for Incremental and Parallel Generation. *Computational Intelligence* 10(4):591–603.

Kurlander, D., and Ling, D. 1995. Planning-Based Control of Interface Animation. In *Proc. of CHI'95*, 472–479.

Lashkari, Y.; Metral, M.; and Maes, P. 1994. Collaborative interface agents. In *Proc. of AAAI-94*, 444–449.

Mann, W. C., and Thompson, S. A. 1987. Rhetorical Structure Theory: A Theory of Text Organization. Report ISI/RS-87-190, Univ. of Southern California, Marina del Rey, CA.

Maybury, M., ed. 1993a. *Intelligent Multimedia Interfaces.* AAAI Press.

Maybury, M. T. 1993b. Planning Multimedia Explanations Using Communicative Acts. In Maybury (1993a). 59–74.

Rist, T., and André, E. 1992. Incorporating Graphics Design and Realization into the Multimodal Presentation System WIP. In Costabile, M. F.; Catarci, T.; and Levialdi, S., eds., *Advanced Visual Interfaces.* London: World Scientific Press. 193–207.

Searle, J. 1980. *Speech Acts: An Essay in the Philosophy of Language.* Cambridge, England: Cambridge University Press.

Takeuchi, A., and Nagao, K. 1993. Communicative facial displays as a new conversational modality. In *Proc. of ACM/IFIP INTERCHI'93*, 187–193.