

# Guiding The User Through Dynamically Generated Hypermedia Presentations with a Life-Like Character

Elisabeth André, Thomas Rist and Jochen Müller  
German Research Center for Artificial Intelligence (DFKI)  
Stuhlsatzenhausweg 3, D-66123 Saarbrücken, Germany  
Email: {andre,rist,jmueller}@dfki.de

## ABSTRACT

Rapid growth of competition on the electronic market place, will generate the demand for new innovative communication styles with web users. In this paper, we develop an operational approach for the automated generation of hypermedia presentations. Unlike conventional hypermedia, we use a life-like presentation agent which presents the generated material, and guides the user through a dynamically expanding navigation space. The approach relies on a model that combines behavior planning for life-like characters with concepts from hypermedia authoring such as timeline structures and navigation graphs.

## Keywords

Automated Presentation of Information, Hypermedia Generation, Life-Like Characters

## INTRODUCTION

The World-WideWeb has just begun to change a very broad range of business processes - from marketing and sales, to customer services, order management and distribution. Rapid growth of competition on the electronic market place, will generate the demand for new innovative communication styles with web users. Much effort has already been spent on the conversion of conventional documentation material, such as printed product brochures and instruction manuals, into HTML-hypertext for the distribution on the web.

In the last few years, animated characters based either on cartoon-style drawings [12], real video [8], or geometric 3D-models [4, 13, 10, 6] have become increasingly popular in user interfaces. For web applications, they are a promising option since they make presentations more lively and appealing. They even allow for the emulation of conversation styles common in human-human communication.

Despite of the raging debate on the sociological effects that life-like characters may have, yet can't have, and perhaps never will have, it is safe to say that they enrich the repertoire of available options which can be used to effectively communicate information to the user. Among other things, they can be employed to:

- attract the user's focus of attention
- guide the user through a presentation
- realize new presentation means, such as two-handed pointing
- convey additional conversational and emotional signals

With the advent of web-browsers which are able to execute programs embedded in webpages, the use of animated characters for the presentation of information over the web has become possible. A web presentation can now comprise dynamic media such as video, animation and speech, all of which have to be displayed in a spatially and temporally coordinated manner. Such a coordination is needed for dynamic presentations in which a life-like character points to and verbally comments on other media objects, such as graphics, video clips, or text passages. The principle is to pack a webpage with:

- (a) the media objects along with a specification of how they need to be arranged and temporally scheduled,
- (b) a presentation runtime engine (for example implemented as a Java applet) which displays the media objects according to the layout specification,

and ship them to the client.

Unlike other approaches, e.g., Ball [5], we primarily employ life-like characters for presenting information. We don't allow for communication with life-like characters via speech in order to avoid problems resulting from the deficiencies of current technology for the analysis of spoken language.

© ACM 1998. This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in IUI98, San Francisco, CA, USA, <http://dx.doi.org/10.1145/268389.268394>

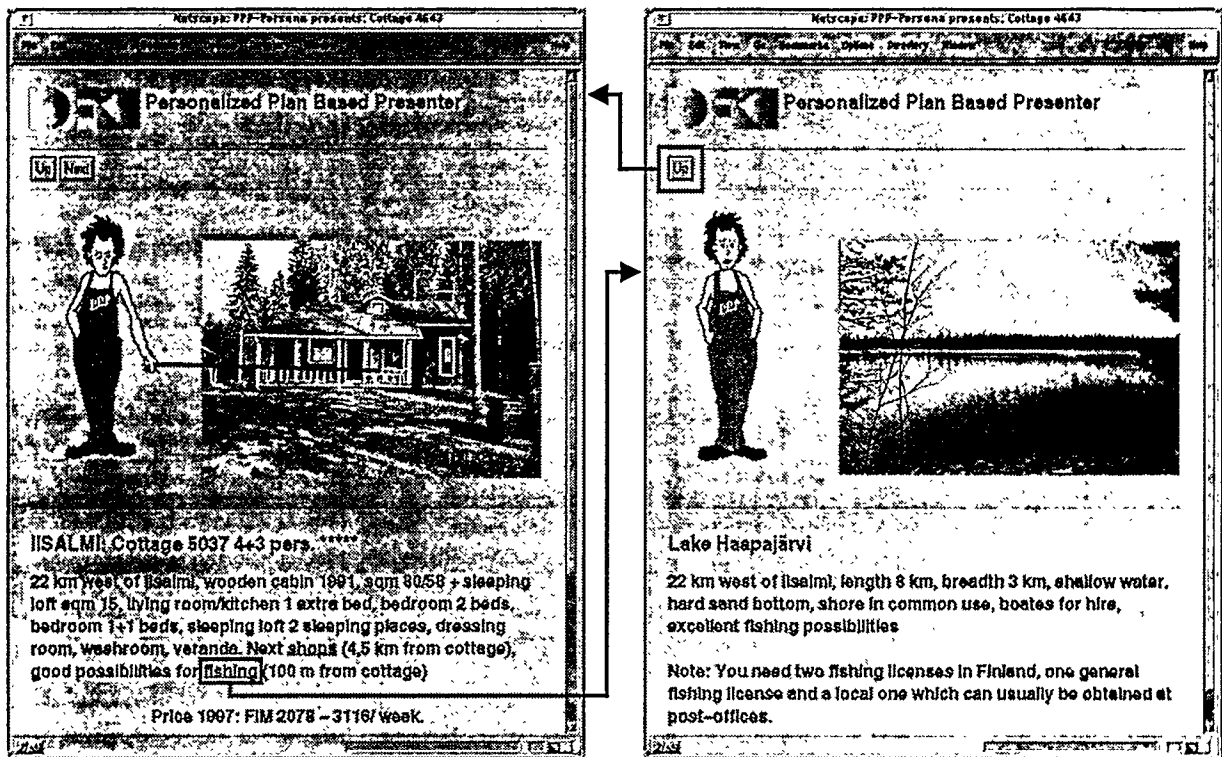


Figure 1: An Interactive Presentation with the Persona

Nevertheless, the user has the possibility of influencing the course of the presentation by making specific choices while it runs. The novelty of our system is that presentation scripts and navigation structures are not stored in advance, but generated automatically from pre-authored document fragments and items stored in a knowledge base.

Fig. 1 shows an example taken from the PPP (Personalized Plan-based Presenter, [12]) system. Suppose the user wants to spend holidays in Finland and is looking for a lake-side cottage. To comply with the user's request, the Persona retrieves matching offers from the WWW, selects one of them and presents it to the user. To give the user the possibility of asking for more information, several items in the text are made mouse-sensitive. Clicking on one of these items will lead to the insertion of a subscenario. For instance, if the user clicks on the fishing item while the first cottage is presented, the Persona will interrupt the current presentation and run a script with fishing possibilities. After that, it will continue with the main script and describe the next offer. However, following a navigation link does not cause paging as in the case of most conventional web presentations. Rather, a new presentation script for the agent along with the required textual and pictorial material is transferred to the client-side presentation runtime engine.

To generate such presentations automatically, we build upon our previous work on the automated planning of presentation

scripts for presentation agents (cf. [3]), and extend this work for interactive web presentations.

#### REPRESENTATION OF THE UNDERLYING INFORMATION

To integrate predesigned and automatically generated material, we start from a database which comprises both information about the domain and information about documents.

Domain information is represented in terms of objects and the relations between them. For example, in the "Cottage-Domain" objects are cottages, lakes, geographic locations, and also activities like hiking, fishing or shopping. A type hierarchy is used to allow for hierarchically structuring domain representations. The set of domain relations may comprise, for example, a part-of relation to express that a certain cottage has a sauna, or a price-relation which may hold between a number and a cottage.

Similar, document information is represented in terms of media objects and relations between them. Media objects are pre-authored document fragments, e.g. a text paragraph or an illustration. Relationships between media objects represent what kind of communicative role a media object may play with respect to another media object in a presentation. For instance, a text paragraph may *elaborate on* an illustration.

Certainly, media objects serve to present domain informa-

	Domain Knowledge	Document World	Encoding Relations
Domain Objects	(ISA C01 cottage)	(ISA pic01 picture)	(DEPICTS pic01 C01)
	(ISA C02 cottage)	(ISA pic02 picture)	(DEPICTS pic02 C01)
	(ISA C03 cottage)	.....	(DEPICTS pic03 C02)
	.....	(ISA text01 text)	.....
	(ISA S01 sauna)	(ISA text02 text)	.....
	.....	.....	.....
Relations	(PARTOF S01 C01)	(ISA video01 video)	(DESCRIBES text01 C01)
	.....	(ELAB text02 text01)	(DESCRIBES text02 S01)
	(PRICE C01 800)	.....	(DESCRIBES text03 C03)
	(PRICE C02 650)	(ILLU pic01 text01)	.....
	(PRICE C03 700)	.....	.....
	.....	(CONTR pic01 pic03)	(SHOWS video01 Hike01)
	.....	.....	.....

Figure 2: Representation of the Underlying Information

tion. To bridge the gap between domain information and media objects, we rely on a set of so-called *encoding relationships*. For example, if the database contains a picture of a certain domain object, then the connection between the two items can be represented by the relation (Depicts pic obj). An encoding relationship is not necessarily a one-to-one mapping. For example, the database may contain several graphics (media objects) for one and the same domain object. Conversely, one and the same media object may be employed for different purposes in different situations.

As Fig. 2 shows, the generation of multimedia presentations can start from information sources which are very different in nature. Note that our approach allows for flexibility concerning the degree of automatization by varying the relative proportion of domain and document knowledge. In the extreme case, the database comprises a very deep domain model from which all media objects can be generated on the fly. We followed this approach in our previous system WIP [2]. Since the current applications of PPP heavily rely on prestored material, a shallow domain model is usually sufficient. Here, explicit representations of the contents and the structure of the document fragments are required. However, we do not assume completeness of the database in the sense that all possible structural relations are represented. Such an assumption is simply unrealistic for most practical applications. Rather, we follow a principle that states: the more structural relations present in the database, the more the flexibility that can be embodied into the navigation structure of the resulting presentation. Such an approach has also been used for the generation of adaptive hypertext, see e.g. [7].

### THE WEBPERSONA PRESENTATION MODEL

Our presentation model has two main ingredients: A model which describes the behavior of the character, and a model for the description of hypermedia presentations.

#### The Behavior Model

What makes up a reasonable behavior for a character depends on a number of factors, such as the chosen *metaphor*,

its *purpose*, and the *conversational setting*. As shown in the example above, our Persona is a cartoon-style human-like figure. Its primary purpose is to execute presentation acts. However, the Persona's behavior is not only determined by the directives (i.e., presentation tasks) specified in the script. Rather, the behavior of the animated character follows the equation:

$$\text{Persona behavior} := \text{directives} + \text{self-behavior}$$

Such self-behaviors are indispensable in order to increase the Persona's vividness and believability. Self-behaviors are compiled from different action types (cf. Fig. 3), they currently comprise actions to enhance the character's vividness, e.g., to span over idle time, actions to move the character to appropriate screen positions, and immediate reactions to external events such as mouse gestures on the presented material.

Though it is certainly possible to include appropriate instructions directly in the presentation script, our approach has an important advantage. From a conceptual point of view, we consider it more adequate since a clear borderline is drawn between a "*what to present*"-part which is determined by the application, and a "*how to present*"-part which, to a certain extent, depends on the particular presenter. From the practical perspective, the approach considerably facilitates script production.

The Persona's behavior is coordinated by a so-called behavior monitor, which determines the next action to be executed and decomposes it into elementary postures. This also includes the augmentation of the Persona's behavior by believability-enhancing behaviors, such as idle-time acts. The postures determined by the behavior monitor are forwarded to a character composer which selects the corresponding frames (video frames or drawn images) from an indexed data-base, and forwards the display commands to the window system.


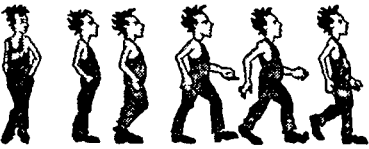

Action Type	Purpose	Sample Frames for Action Visualization
idle time acts	To achieve a more vivid, but not distracting behavior. Especially to avoid freezing during pauses which may occur between presentation jobs.	 e.g. looking and swinging around
navigation acts	To move the character to appropriate positions where it will carry out presentation jobs. (Definition of Actions depends on chosen character type, e.g. walking versus flying on the screen)	 e.g. turning around and walking
immediate reactions	To immediately react to external events, such as user interactions, which require direct visual feedback by the Persona.	e.g. dragging 

Figure 3: Classification of Persona Self Behaviors

### The Hypermedia Model

An important characteristic of our web presentations is that they are not just played back, but have a branching structure which allows the user to choose between different possibilities of navigation. That is, the course of a presentation changes at runtime depending on user interactions. In this section, we will present a model for describing such interactive presentations.

Inspired by the Amsterdam Hypermedia Model [9], we represent web presentation by a collection of presentation units and a set of transitions specifying how to get from one presentation unit to the other.

A presentation unit is defined by a collection of media objects together with a presentation script. We assume that a presentation unit is a self-contained part of a presentation whose media objects are placed in time independent of media objects corresponding to other presentation units.

Presentation scripts entail directions for the character concerning the presentation of media objects. As in other animation scripting systems, we visualize presentation scripts by timeline diagrams which position all actions to be executed by the character along a single time axis. According to the timeline diagram shown in Fig. 4, the Persona displays a text that is accompanied by an illustration and a text, points to an object in the illustration and verbally provides some additional information. The durations of complex acts correspond to the length of the darker bars, the lighter bars refer to durations of elementary acts.

Timeline diagrams enable us to represent the temporal behavior of a presentation in an intuitive manner, however, they provide no means of describing the control flow of interac-

tions. Therefore, we combine timeline diagrams with state-transition graphs. That is timeline diagrams are used for describing the temporal behavior of single presentation units while state-transition graphs serve to describe the navigation structure of a presentation.

A state-transition graph  $G$  is defined by a set of nodes and edges, i.e.  $G = \langle N, E \rangle$ . With each node  $n \in N$ , we associate a presentation unit, and a default duration, usually the duration of the presentation unit, i.e.  $n := \langle \text{duration} \rangle \langle \text{presentation\_unit} \rangle$ . Each node corresponds to the state of a presentation. If a node is entered, the corresponding presentation script is run. Consequently, being in a certain state means that the corresponding presentation unit is active. An edge  $e \in E$  is defined by its connecting nodes, a condition and an action, i.e.  $e := \langle \text{from} \rangle \langle \text{to} \rangle \langle \text{condition} \rangle \langle \text{action} \rangle$ .

A transition is made if one of predicates associated with the edges leading away from the node is satisfied or the default duration is over. Predicates usually refer to user interactions, such as clicking on mouse-sensitive icons in a presentation. An interesting question is the timepoint of transition. Should the system wait until the presentation is completed or interrupt and resume it later? Since a presentation unit may be rather long, we have chosen the second possibility. However, to avoid loosing the coherency of a presentation, we don't allow for the interruption of elementary presentation acts that vary in time, such as speaking or pointing, but wait until these acts are executed. When returning to a node, the system continues the presentation by playing only the remaining part of the script.

	0	2	4	6	8
Describe					
S-Display-Text					
Illustrate					
S-Display-Window					
Emphasize					
Label					
S-Speak					
S-Point					

Figure 4: Example of a Timeline Diagram

Each graph contains a starting and an end node with an empty presentation script and a default duration of 0. A path through a presentation graph is defined as a sequence of nodes  $n_i$ ,  $1 \leq i \leq m$  where  $n_1$  is the starting node and  $n_m$  is the end node. It corresponds to a specific way of viewing the presentation.

The concepts introduced above will be illustrated using the example presented in the introduction. The navigation graph of this example is shown in Fig. 5. The presentation is started by entering the starting node. Since the default duration of this node is 0, the first cottage node is entered immediately and the corresponding presentation script for the Persona is run. Let's suppose the user clicks on the shopping button while the Persona describes the first cottage. As a consequence, the presentation is interrupted and the shopping script is played. That is the Persona now informs the user about shopping possibilities. After that, the system returns to the first cottage node and plays back the remaining parts of the script. After the default time of 23 time units has passed, a transition is made to the second cottage node. Here, again the user has the possibility of requesting for more information, e.g., about hiking possibilities. After the script for the second cottage has been run and the user hasn't requested for more information, a transition is made to the end node.

## AUTOMATED CREATION OF WEB PRESENTATIONS

In the last section, we presented a model for describing interactive web presentations. However, the manual creation of navigation graphs and presentation scripts is tedious and error-prone. To satisfy the individual needs of a large variety of users, the human author would have to prepare a large number of presentations in advance and to hold them on stock. In the rapidly growing field of online presentation services, the situation is even worse. If live data has to be communicated there is simply not enough time to manually create and continuously update presentations. For example, the nodes of the navigation graph in Fig. 5 correspond to cottages which have been selected for the user on the fly. Since the number of the available cottages and also their features may change at any time, it doesn't make sense to rely on predesigned navigation graphs or presentation scripts. In the following, we will discuss how to automate the generation process. This process comprises the following tasks:

- (1) the design of a multimedia discourse structure reflecting how the single parts of a presentation are related to each other
- (2) the decomposition of the presentation into self-contained presentation units
- (3) the design of a navigation graph
- (4) the design of presentation scripts for each presentation unit

To accomplish these tasks, we extend our previous work on the automatic generation of non-interactive multimedia presentations. The main idea behind this approach was to formalize action sequences for composing multimedia material and designing scripts for presenting this material to the user as operators of a planning system. The effect of a planning operator refers to a complex communicative goal (e.g. to provide information about a cottage) while the expressions in the body indicate which acts have to be executed in order to achieve this goal (e.g., to show an illustration and to describe it). The temporal behavior of these acts is specified by a list of qualitative and metric constraints. Like other authors in the Multimedia community, e.g. see [11], we represent qualitative constraints in an "Allen-style" fashion (cf. [1]) which allows for the specification of thirteen temporal relationships between two named intervals, e.g. (*Speak1 (During) Point2*). Quantitative constraints appear as metric (in)equalities, e.g. ( $5 \leq \text{Duration Point2}$ ).

The input to the presentation planner is a complex presentation goal, e.g., to present a set of cottages. To accomplish this goal, the planner looks for operators whose header subsume it. If such an operator is found, all expressions in the body of the operator will be set up as new subgoals. The planning process terminates if all subgoals have been expanded to elementary production/retrieval or presentation tasks or to goals that will be realized by hyperlinks in the final presentation. The result of the planning process is a refinement-style plan which reflects the rhetorical structure of the presentation (see Fig. 6). For example, there is a sequence relationship between the single cottage presentations and elaboration relationships between a cottage presentation and the corresponding subscenarios. Furthermore, this plan specifies how the single parts should be temporally coordinated. For instance, the text and the corresponding illustration should be displayed at the same time.

To allow for the dynamic expansion of the navigation space, we do not expand goals corresponding to hyperlinks at presentation design time, but only if the user selects the corresponding button at presentation runtime. For example, the underlined Elaborate- and Introduce-nodes in Fig. 6 have not yet been expanded since the system has decided to realize them as hyperlinks. This method has the advantage that presentations can be adapted to the user's previous navigation behavior and to the information that has been conveyed so far.

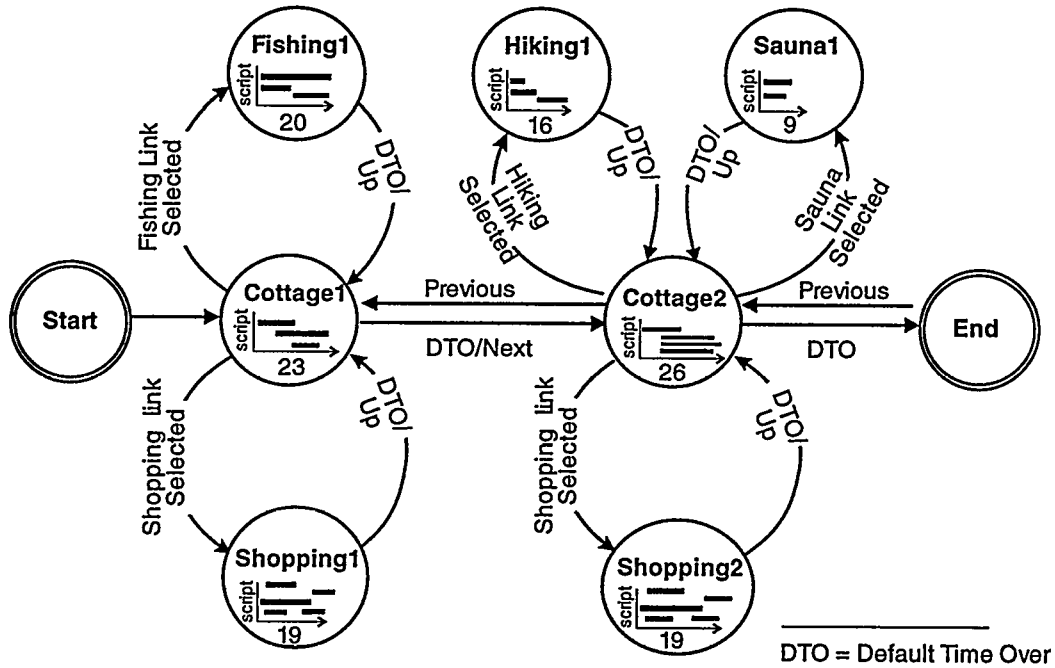


Figure 5: Navigation Graph for the Cottage Example

During the planning process, relevant knowledge units for achieving the goals are retrieved from the domain and document knowledge bases and distributed onto different webpages that will be connected by hyperlinks. The presentation of one of these webpages then corresponds to a presentation unit. When creating this network of webpages, the following criteria are considered:

- *user characteristics (such as user goals, knowledge and interests)*  
User characteristics are considered by ranking domain objects and media objects according to their relevance to a particular user or user group. All items of low relevance are realized as hyperlinks. If it's unclear whether an item is of relevance to a particular user, the item becomes a candidate for a hyperlink as well.
- *temporal relationships between items*  
Temporally overlapping presentation parts are assigned to the same presentation unit.
- *rhetorical relationships between items*  
Elaborations are good candidates for hyperlinks, in particular if the information is of lower relevance to the user or space is limited. Items of a sequence relationship are assigned to different presentation units if the selection of items depends on the user's interest in previously presented items.
- *cohesive links between items*  
The distribution of material onto different webpages

should not disturb the user's viewing process. For example, an illustration should not be separated from its accompanying text if the text contains crossreferences to the illustration.

- *layout constraints*  
Items of lower relevance are realized as hyperlinks if the document parts to be presented don't fit on one screen page.
- *optional information*  
In our approach, the author of plan operators has the possibility to annotate some presentation acts as optional. On the one hand, this method gives the human author more control over the final presentation. On the other hand, the prespecification of hyperlinks reduces the adaptive capabilities of the system at runtime.

In the cottage example, the acts *S-Display-Text* and *Illustrate* are collected into one presentation unit because there may be crossreferences from text to graphics. The acts *Illustrate* and *Emphasize* cannot be assigned to different presentation units since they refer to each other and temporally overlap. In contrast to this, all elaborations are realized by own presentation units because the information is considered less relevant. Finally, the two introductions are assigned to different presentation units since they refer to different matching offers which are described in sequence.

For each new presentation unit, the planner creates a node in the navigation graph and specifies how this node can be reached from other nodes and vice versa. These conditions

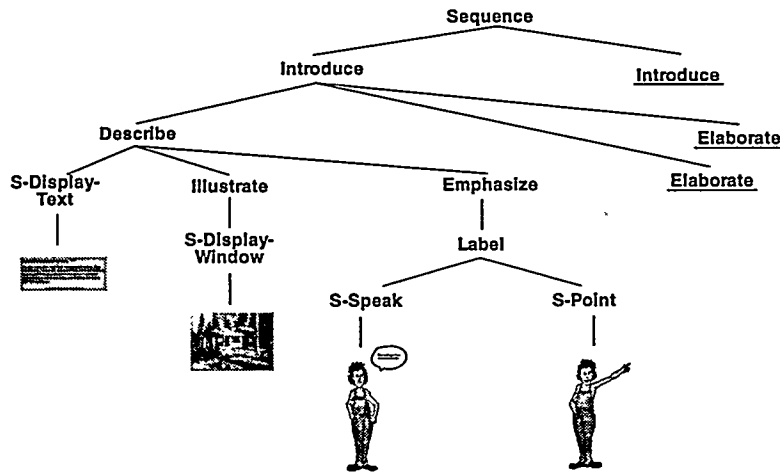


Figure 6: Rhetorical Structure of a Presentation Unit of the Cottage Example

then correspond to the predicates associated with the edges of the navigation graph. For instance, to get from a scenario to an elaborating subscenario, a specific button has to be selected. If the presentation associated with the subscenario is over or the user clicks on an up button, the system returns to the main scenario. To jump back and forth between scenarios connected via a sequence relationship, the user may select a next or previous button resp.

After the rhetorical structure of a presentation unit has been determined, the planner creates a schedule. Since the temporal behavior of each unit can be specified independent of other units, the system can start with this task without knowing which links the user will eventually follow. It first collects all constraints on and between actions of a unit. After that, it determines the transitive closure over all qualitative constraints and computes numeric ranges over interval endpoints and their difference. Finally, possibly occurring disjunctions are resolved and a total temporal order is computed (see [3]).

## EVALUATION

Our research on animated interface agents was motivated by the assumption that they make man-machine communication more effective. In order to find empirical support for this conjecture, we conducted a psychological experiment in which 28 subjects (15 females, 13 males, average age: 28) were confronted with 5 web-based presentations that they are subsequently asked questions about. Subjects were allowed to spend as much time as they required to answer the questions, but not to watch a presentation several times. On the average, each subject spend 45 minutes on the experiment. Our study focused on two issues:

- 1) the effect of a Persona on the subject's rating of the presentations (a subjective measure), and
- 2) its effect on the subject's comprehension of presenta-

tions (an objective measure).

The first issue was measured through comprehension and recall questions following the presentations. The second issue was measured through a questionnaire at the end of the experiment.

In the experiment, two variables were varied. The first variable referred to the Persona itself. The Persona was either absent or present. In the experiment without the Persona, a voice spoke the same explanations as in the Persona-version and pointing gestures by the Persona were replaced with an arrow. The second variable was the information type. Subjects were confronted with technical descriptions of pulley systems and with person descriptions (i.e., information about DFKI employees). The first variable was manipulated between-subjects, while the second variable was manipulated within-subjects. Thus, each subject viewed either presentations with or without the Persona, but each subject was confronted with both kinds of presentation.

Concerning our first objective, the evaluation of the Persona's affective impact, our study revealed a positive effect. Subjects confronted with the Persona-based presentations found the explanations less difficult to understand and perceived the Persona as being helpful and entertaining. Only one subject indicated that he would prefer presentations without a Persona in the future. The subjective rating of the subjects was more positive in the case of the pulley experiment than in the case of the DFKI experiment. We hypothesize that this result is due to the fact that the Persona's realization as a workman is more appropriate to technical descriptions than to institute descriptions.

Concerning the second objective, the evaluation of the Persona's learning effect, we didn't find a significant difference between the Persona and the No-Persona version. That is the Persona did neither contribute to the students' comprehension of the technical matters in the pulley experiment, nor

to the students' recall capabilities in the second experiment. As a possible reason, we indicate that we only exploited Persona behaviors that can be easily replaced with other means of communication not necessarily requiring the existence of a Persona. In our experiments, Persona gestures were restricted to neutral facial expressions (i.e. head and eye movements towards the objects currently being explained and lip movements indicating that the Persona is speaking), pointing gestures and simple idle time actions, such as breathing or tapping with a foot.

On the other hand, initial concerns that people would be distracted by the Persona and concentrate too much on the Persona's facial expressions instead of looking at the referent of the pointing gestures were not confirmed. In the questionnaire, all subjects indicated that the Persona did not distract them.

## CONCLUSION

We have argued that the use of life-like characters are a promising option for presentations on the web. In order to describe such presentations, we combined a behavior model for life-like characters with concepts from hypermedia authoring. Since the manual specification of such presentations would be too labour intensive and error-prone, we also showed how to automate this process. Our current prototype is capable of generating both presentation scripts for life-like characters, and navigation structures to allow the user to dynamically change the course of a presentation at runtime.

We plan to extend our work on presentation agents in several directions. First of all, we intend to employ more than one agent in a presentation. This extension would allow for different role castings, consider for example two agents discussing the pros and cons of a certain product. A new line of research will be opened up with the dissemination of virtual worlds via the web, as life-like agents and so-called avatars will become the inhabitants of these worlds. While the audio-visual realization of these agents will be facilitated by the emerging VRML 2 standard, our technology may be used to have them perform presentation tasks. Finally, we intend to evaluate the Persona's effect on the user's navigation behavior. In particular, we intend to investigate in how far recommendations given by different Personas are followed by the user.

## ACKNOWLEDGMENTS

This work has been supported by the BMBF under the contracts ITW 9400 7 and 9701 0. We would like to thank Peter Rist for drawing the cartoons, H.-J. Profitlich and M. Metzler for the development of the temporal reasoner, Frank Biring for implementing the Persona Compiler, and Susanne van Mulken for supervising the empirical evaluation. In addition, we are grateful for the comments of the anonymous reviewers.

## REFERENCES

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

1983.

- 2 E. André, W. Finkler, W. Graf, T. Rist, A. Schauder, and W. Wahlster. WIP: The Automatic Synthesis of Multimodal Presentations. In M. Maybury, editor, *Intelligent Multimedia Interfaces*, pages 75–93. AAAI Press, 1993.
- 3 E. André and T. Rist. Coping with temporal constraints in multimedia presentation planning. In *Proc. of AAAI-96*, volume 1, pages 142–147, Portland, Oregon, 1996.
- 4 N.I. Badler, C.B. Phillips, and B.L. Webber. *Simulating Humans: Computer Graphics, Animation and Control*. Oxford University Press, New York, Oxford, 1993.
- 5 G. Ball. Dialogue initiative in a web assistant. In *Proc. of Life-Like Computer Characters '96*, Snowbird, Utah, 1996.
- 6 G. Ball, D. Ling, D. Kurlander, J. Miller, D. Pugh, T. Skelly, A. Stankosky, D. Thiel, M. van Dantzich, and T. Wax. Lifelike computer characters: the persona project at microsoft. In J.M. Bradshaw, editor, *Software Agents*. AAAI/MIT Press, Menlo Park, CA, 1997.
- 7 P. Brusilovsky, E. Schwarz, and G. Weber. Elm-art: An intelligent tutoring system on world wide web. In C. Frasson and G. Gauthier and A. Lesgold, editor, *Intelligent Tutoring Systems (Lecture Notes in Computer Science, Vol. 1086)*, pages 261–269, Berlin, New York, Heidelberg, 1996. Springer.
- 8 S. Gibbs, C. Breiteneder, V. de Mey, and M. Papathomas. Video Widgets and Video Actors. In *Proc. of the UIST'93 (ACM SIGGRAPH Symp. on User Interface Software and Technology)*, pages 179–184, Atlanta, GA, U.S.A., 1993.
- 9 L. Hardman, D.C.A. Bulterman, and G. van Rossum. The Amsterdam Hypermedia Model: Adding Time and Context to the Dexter Model. *Communications of the ACM*, 37(2):50–62, 1994.
- 10 K. Perlin and A. Goldberg. Improv: A System for Scripting Interactive Actors in Virtual Worlds. *Computer Graphics*, 28(3), 1996.
- 11 Claudio S. Pinhanez, Kenji Mase, and Aaron Bobick. Interval Scripts: a Design Paradigm for Story-Based Interactive Systems. In *CHI97 Conference Proc.*, pages 287–294, Atlanta, GA, Mar. 1997.
- 12 T. Rist, E. André, and J. Müller. Adding Animated Presentation Agents to the Interface. In *Proceedings of the 1997 International Conference on Intelligent User Interfaces*, pages 79–86, Orlando, Florida, 1997.
- 13 B.A. Stone and J.C. Lester. Dynamically sequencing an animated pedagogical agent. In *Proc. of AAAI-96*, volume 1, pages 424–431, Portland, Oregon, 1996.