WebPersona: a lifelike presentation agent for the World-Wide Web

Elisabeth André, Thomas Rist, Jochen Müller

Angaben zur Veröffentlichung / Publication details:

André, Elisabeth, Thomas Rist, and Jochen Müller. 1998. "WebPersona: a lifelike presentation agent for the World-Wide Web." *Knowledge-Based Systems* 11 (1): 25–36. https://doi.org/10.1016/s0950-7051(98)00057-4.

WebPersona: a lifelike presentation agent for the World-Wide Web

Elisabeth André*, Thomas Rist, Jochen Müller

German Research Centre for Artificial Intelligence (DFKI), Stuhlsatzenhausweg 3, D-66123 Saarbrücken, Germany

Keywords: Presentation agent; Hypermedia; Navigation space

1. Introduction

The World-Wide Web 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 in the electronic market place will generate a 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 distribution on the web.

In the last few years, animated characters based either on cartoon-style drawings [20], real video [11] or geometric 3D models [5,7,17,21] 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 the raging debate on the sociological effects that lifelike characters may have, yet cannot have and perhaps never will have, it is safe to say that they enrich the repertoire of available options which can be used effectively to 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 lifelike 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

1. the media objects, along with a specification of how they need to be arranged and temporally scheduled,
2. a presentation runtime engine which displays the media objects according to the layout specification,

and ship them to the client.

In this paper, we present the PPP Persona (**P**ersonalized **P**lan-based **P**resenter [20]), a lifelike presentation agent which gathers relevant information from various web sources and data bases, restructures the information into self-contained units and presents them to the user. Unlike other approaches, e.g. that of Ball [6], we primarily employ lifelike characters for presenting information and don't support communication with lifelike characters via speech. Systems which allow for free-form natural-language input often frustrate users since it is difficult to predict whether a system will be able to handle a request or not. This problem is even aggravated if a lifelike character is used because it suggests intelligence. To avoid problems resulting from the deficiencies of current technology for the analysis of spoken

* Corresponding author.

language, we offer the user a hypermedia-style interface which allows him to influence the course of a presentation by making specific choices while it is run. The advantage of such an interface is that it makes explicit which kinds of interaction possibility the system offers without requiring the user to learn a specific language.

Fig. 1 shows an example. Suppose the user wants to spend a holiday in Finland and is looking for a lakeside 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 being presented, the Persona will interrupt the current presentation and run a script about 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.

In the following sections, we describe how to generate such presentations automatically. We first present the underlying knowledge base which maintains both domain and document knowledge. After that, we present a model for interactive hypermedia presentations which combines behaviour planning for lifelike characters with concepts from hypermedia authoring. This model forms the basis for our operationalization which will be described in the subsequent two sections. Finally, we report on the outcome of a recent empirical study which compared objective and subjective ratings of presentations with and without a Persona.

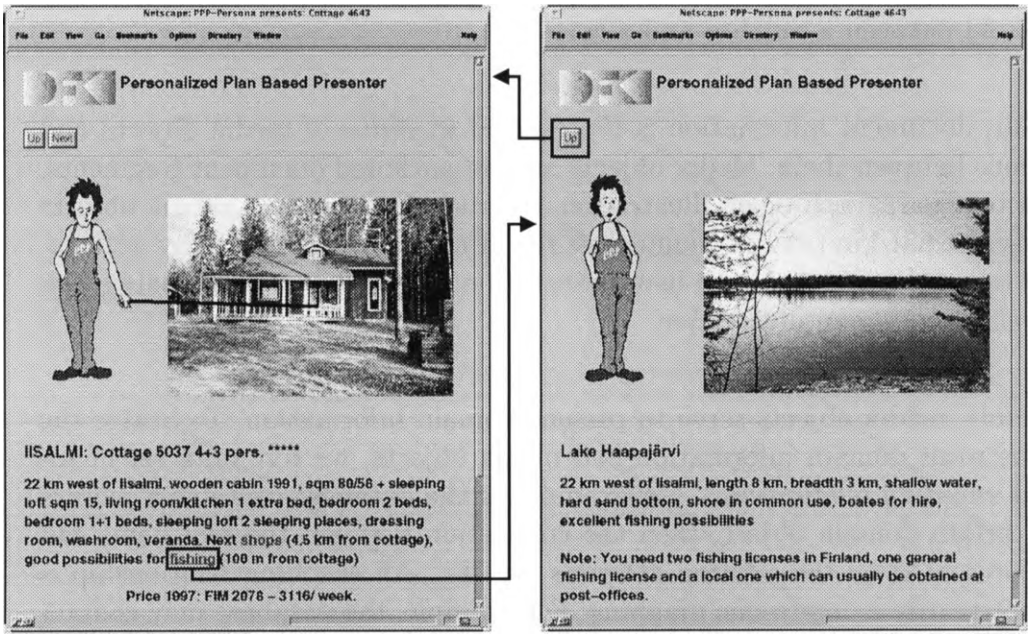
2. Representation of the underlying information

Our system does not store presentation scripts and navigation structures in advance, but generates them automatically from pre-authored document fragments and items stored in a data base. To integrate predesigned and automatically generated material, we start from a hybrid 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.

Similarly, 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 the kind of communicative role that 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 information. To bridge the gap between domain information and media objects, we rely on a set of so-called *encoding*


Netscape: PPP–Persona presents: Cottage 4643
File Edit View Go Bookmarks Options Directory Window Help
Personalized Plan Based Presenter
Up
Next
Netscape: PPP–Persona presents: Cottage 4643
File Edit View Go Bookmarks Options Directory Window Help
Personalized Plan Based Presenter
Up

IISALMI: Cottage 5037 4+3 pers. *****

22 km west of Iisalmi, wooden cabin 1991, sqm 80/56 + sleeping loft sqm 15, living room/kitchen 1 extra bed, bedroom 2 beds, bedroom 1+1 beds, sleeping loft 2 sleeping places, dressing room, washroom, veranda. Next shops (4.5 km from cottage), good possibilities for fishing (100 m from cottage)

Price 1997: FIM 2076 – 3116/ week.

Lake Haapajärvi

22 km west of Iisalmi, length 8 km, breadth 3 km, shallow water, hard sand bottom, shore in common use, boates for hire, excellent fishing possibilities

Note: You need two fishing licenses in Finland, one general fishing license and a local one which can usually be obtained at post–offices.

Fig. 1. An interactive presentation with the Persona.

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

Fig. 2. Representation of the underlying information.

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 various degrees of automation 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 rely heavily on prestored material, a shallow domain model was 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. [8].

3. The WebPersona presentation model

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

3.1. The behaviour model

What makes up a reasonable behaviour 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. This includes summarizing the contents of a webpage, directing the user's attention to parts of the document he might be interested in, providing additional information not listed on the webpage and recommending links to follow. To accomplish these tasks, the Persona relies on gestures that: express emotions (e.g., approval or disapproval), convey the communicative function of a presentation act (e.g., warn, recommend or dissuade), support referential acts (e.g., to look at an object and point at it), regulate the interaction between the Persona and the user (e.g., establishing eye contact with the user during communication) and indicate that the Persona is speaking. Of course, these gestures may also superimpose on each other. For example, to warn the user, the Persona lifts its index finger, looks towards the user and utters the warning.

However, the Persona's behaviour is not only determined by the directives (i.e., presentation tasks) specified in the script. Rather, the behaviour of the animated character follows the equation:

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

Such self-behaviours are indispensable in order to increase the Persona's vividness and believability. Self-behaviours

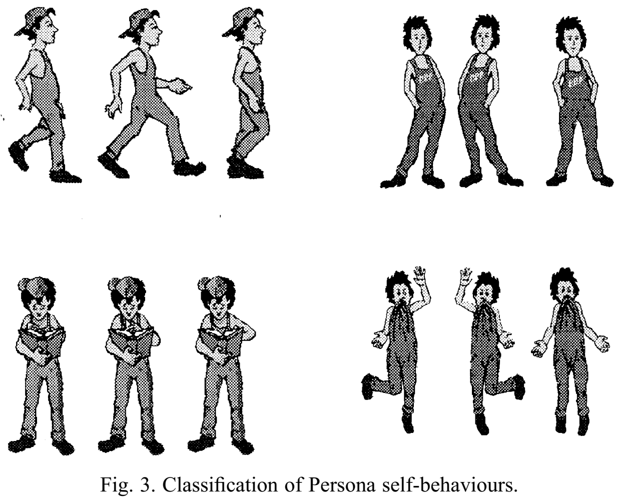


Fig. 3. Classification of Persona self-behaviours.

are compiled from different action types (see Fig. 3); they currently comprise the following actions:

- *Low-level navigation acts*. In some cases, the Persona has to move to an appropriate position on the screen before carrying out presentation tasks, such as pointing to an object. The kind of navigation act depends on the chosen metaphor. For example, human-like agents like the Persona walk or jump to an appropriate position on the screen while agents like Microsoft's parrot Peedy fly (see [7]).
- *Idle-time acts*. To ensure that the Persona exhibits life-like qualities, it has to stay 'alive' even in an idle phase, for instance, when waiting for material to be delivered by the generators or the retrieval components. Typical acts to span pauses are breathing or tapping with a foot.
- *Acts that indicate activity*. If the system is still working on a problem but the character doesn't have to carry out any presentation tasks, it may execute acts to visualize the system's state. For instance, the PPP Persona pulls out a book and starts turning over pages when information is retrieved from the web.
- *Reactive behaviours*. The Persona should be able to react to user interactions immediately and give visual feedback. For instance, if the user drags the Persona over the screen, the Persona fidgets.

The distinguishing feature of our model is the clear distinction between task-specific directives and character- and situation-specific self-behaviours. Though it is certainly possible to include appropriate enhancement of believability 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 since scripts can be formulated on a higher level of abstraction.

3.2. 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 interaction. In this section, we will present a model for describing such interactive presentations.

Inspired by the Amsterdam Hypermedia Model [12], we represent web presentation by a collection of presentation units and a set of transitions specifying how to get from one presentation unit to another.

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 first creates a window which includes a graphical object, shows it to the user and elaborates on the single parts of the graphical object by pointing to them and providing some additional information verbally. The durations of complex acts correspond to the length of the white bars, whereas the dark bars refer to durations of elementary acts.

Timeline diagrams enable us to represent the temporal behaviour of a presentation in an intuitive manner; however, they provide no means of describing the control flow of interactions. Therefore, we combine timeline diagrams with state-transition graphs. That is, timeline diagrams are used for describing the temporal behaviour of single presentation units whereas state-transition graphs serve to describe the navigation structure of a presentation.

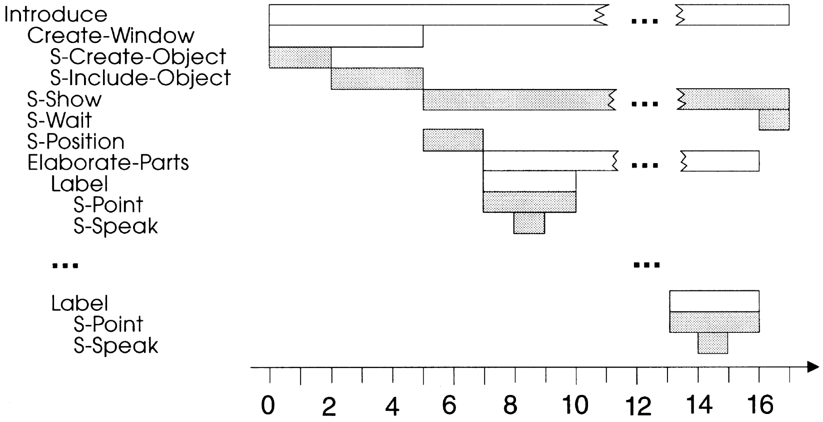


Fig. 4. Example of a timeline diagram.

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. $n: = (\langle duration \rangle \langle presentation\_unit \rangle)$. 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. $e: = (\langle from \rangle \langle to \rangle \langle condition \rangle \langle action \rangle)$.

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 transmission. 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 losing 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.

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 more information, e.g. about hiking possibilities. After the script for the second cottage has been run and the user hasn't asked for more information, a transition is made to the end node.

4. Automated creation of presentation scripts

In the previous 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 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 available cottages and also their

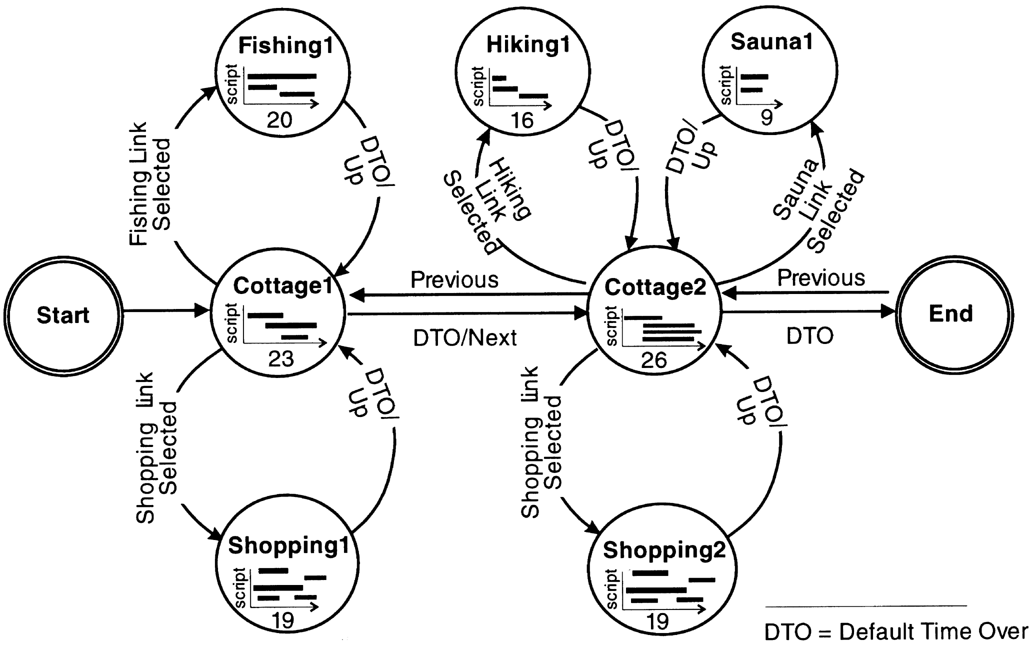


Fig. 5. Navigation graph for the cottage example.

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 build upon our previous work on the automated planning of presentation scripts for presentation agents (see [3]), and extend this work to interactive web presentations.

The main idea behind this approach is to formalize action sequences for composing multimedia material and designing scripts for presenting this material to the user as the 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), whereas 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 behaviour of these acts is specified by a list of qualitative and metric constraints. Like other authors in the Multimedia community, e.g. see [18], we represent qualitative constraints in an 'Allen-style' fashion (see [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. ($5 \leq$ *Duration Point2*).

The input to the presentation planner is a complex presentation goal. To accomplish this goal, the planner looks for operators whose headers 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 when 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 behaviour and to the information that has been conveyed so far.

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

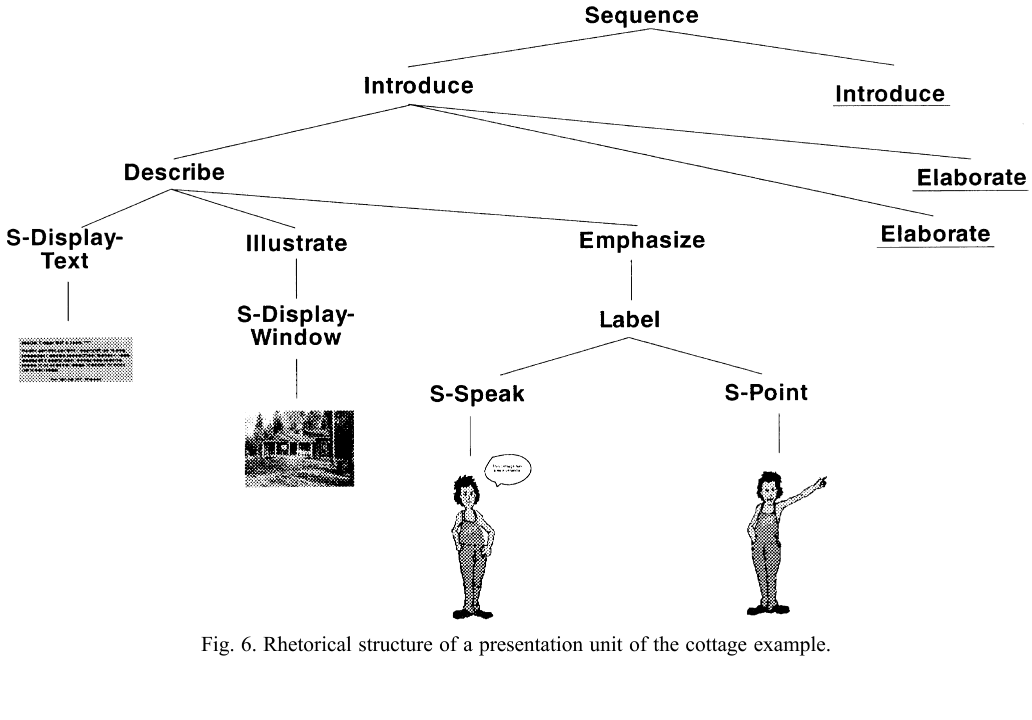


Fig. 6. Rhetorical structure of a presentation unit of the cottage example.

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 is 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, such as the acts *Illustrate* and *Emphasize* in the cottage example, are assigned to the same presentation unit.
- *Rhetorical relationships between items.* Elaborations are good candidates for hyperlinks, particularly if the information is of lower relevance to the user or if space is limited. For instance, all elaborations in the cottage example are realized by separate presentation units because the information is considered less relevant.
- *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. For instance, in the cottage example, the acts *S-Display-Text* and *Illustrate* are collected into one presentation unit because there might be crossreferences from text to graphics.
- *Layout constraints.* Items of lower relevance are realized as hyperlinks if the document parts to be presented don't fit on one screen page and scrolling should be avoided.
- *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.

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 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 a previous button.

After the rhetorical structure of a presentation unit has been determined, the planner creates a schedule. Since the temporal behaviour of each unit can be specified independently 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, a schedule is built up by resolving possibly occurring disjunctions and computing a total temporal order (see [3]). Since the temporal behaviour of presentation acts may be unpredictable at the design time, the schedule will be refined at runtime by adding new metric constraints to the constraint network.

5. Transformation of presentation scripts into animation sequences

The generated presentation scripts are forwarded to the so-called Persona engine (see Fig. 7) which has to accomplish the following tasks:

- *Determination of the next action to perform.* Responses to user interactions usually have the highest priority and may lead to the interruption of a presentation, e.g., if the user signals that he or she is no longer interested in a certain topic. Presentation tasks have a higher priority than idle-time scripts, which are run only if the Persona has no other tasks to perform.
- *Conversion of high-level presentation tasks into fine-grained animation sequences.* This includes the decomposition of high-level presentation tasks into elementary

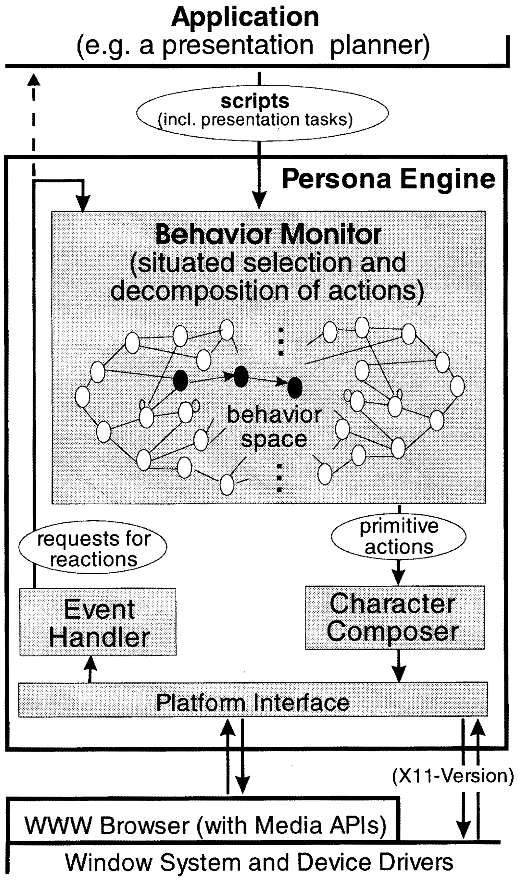


Fig. 7. Architecture of the Persona Engine.

gestures and the satisfaction of necessary preconditions for the execution of these gestures.

- *Augmentation of the Persona's behaviour by believability-enhancing behaviours.* In order not to distract the user, idle-time gestures with a high visual impact should be executed only in rare cases. Furthermore, monotonous repetitions should be avoided since they destroy a character's believability.

To support the specification of the Persona's behaviours, we have defined a declarative authoring language which represents Persona behaviours as plan operators. However, while planning techniques have proven useful for the generation of high-level presentation scripts, animation sequences have to be created in realtime and thus require a method which is computationally less expensive. To solve this problem, we follow an approach by Ball et al. [7] and precompile the declarative behaviour specifications into a finite state machine. That is, we compute beforehand, for all possible situations, which animation sequence to play. As a result, the system just has to follow the paths of the state machine when making a decision at runtime instead of starting a complex planning process (see [4]).

The compiled state machine forms the so-called *behaviour monitor* (see Fig. 7). Besides the behaviour monitor, the Persona Engine also comprises an *event handler*, a *character composer* and an *interface* which is tailored to the target platform (currently either X11 or a Java™ interpreter). The task of the event handler is to recognize whether input derived from the platform interface needs immediate responses from the Persona. That is, for each input message the event handler checks whether the message triggers one of the so-called 'reactive behaviours' stored in an internal knowledge base. If this is the case, the selected behaviour is made accessible to the behaviour monitor. Depending on the application, notifications may also be forwarded to the application program. For example, in our PPP system some events are interpreted as requests for the satisfaction of new presentation goals and thus activate a presentation planner (thus the dotted line in Fig. 7). The postures determined by the behaviour monitor are forwarded to a *character composer* which selects the corresponding frames (video frames or drawn images) from an indexed database and forwards the display commands to the window system.

6. 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 with 30 adult participants (see also [15]).

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 presentations (an objective measure). We thought that three effects on comprehension and recall might occur:

1. The Persona contributes to the comprehension and recall of presentations because of its strong motivational impact.
2. The Persona has a negative effect on the comprehension and recall of presentations because it distracts the user.
3. There is no effect of the Persona on comprehension or recall because the Persona neither motivates nor distracts the user or these two factors compensate for each other.

Since earlier studies already provided evidence that a lifelike character has a strong effective impact on the user (e.g., see [14,22,24]), we expected that this effect also occurs in our case.

6.1. Experimental setting

Participants were 15 females and 15 males, on average 28 years of age, all native speakers of German and recruited from the Saarbrücken university campus. Most of them were not computer specialists, but all of them had some experience in using computers for web surfing and editing purposes.

The subjects were confronted with five 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 spent 45 min on 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. That is, the Non-Persona version conveyed exactly the same information as the Persona version. This was important because we were interested in the effect of the mere presence of a Persona.

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). In the first case, we showed the subjects illustrations of four different pulley systems and conveyed information concerning the parts of the pulley systems and their kinematics auditorially. Whenever a part of the pulley system was mentioned, a pointing gesture was performed. For the condition with non-technical material, we designed a presentation in which ten fictitious office employees were introduced. For each employee, his or her photograph was shown and information concerning his or her name and occupation conveyed auditorially. Furthermore, his or her office was pointed at on an office floor layout.

The first variable was manipulated between subjects, while the second variable was manipulated within subjects.

Table 1
Means for the general questions asked in the questionnaire (Part A). Ratings range from 0 (negative answer, i.e. indicating disagreement) to 4 (positive answer, i.e. indicating agreement)

Question	Type of information			
	Technical information		Person descriptions	
	Persona condition No Persona	Persona	Persona condition No Persona	Persona
Presentation difficult	1.63	1.09	2.07	2.14
Presentation entertaining	1.28	2.07	1.78	2.0
Test difficult	2.00	1.50	2.86	2.93
Presentation interesting	1.71	2.21	2.0	2.28
Information overload	1.43	1.14	2.50	2.86

Thus, each subject viewed either presentation with or without the Persona, but each subject was confronted with both kinds of presentation. Neither of the two groups knew about the existence of the other.

The Persona's learning effect was measured by comprehension and recall questions following the presentations. For the technical scenario, the subjects had to answer questions such as: ''Which objects does the red rope touch?'' or ''In which direction does the lower pulley move if the free end of the red rope is pulled down?''. For the office experiment, we presented the subjects with photographs of office employees and a layout of the office floor. The subjects had to recall the employees' name, occupation and office number.

The Persona's affective impact was measured through a questionnaire at the end of the experiment. Part A of the questionnaire contained general questions on the presentation, such as ''Was the presentation difficult to understand?'' or ''Did you find the presentation entertaining?'', while part B contained specific questions on the Persona, such as ''How appropriate was the Persona's behaviour?'', ''Did the Persona help you to concentrate on relevant information?'' or ''Would you prefer presentations with or without a Persona in the future?''.

6.2. Experimental results

Regarding our first objective, the evaluation of the Persona's affective impact, our study revealed a positive effect (see Table 1, Table 2). Only one subject indicated that he would prefer presentations without a Persona in the future. t-tests on the data listed in Table 1 shows that subjects confronted with technical descriptions[1] found the subject matter significantly less difficult ($t(26) = -2.51$; $p = 0.0186$) and the presentation more entertaining ($t(26) = -2.38$; $p = 0.0247$) with the Persona present.

In the case of the office experiment, we found no significant difference between the ratings of the difficulty of the presentation and its entertainment value. Also, subjects found the Persona's behaviour less appropriate in this domain and felt that the Persona was less helpful as a concentration aid (see Table 2). We hypothesize that the less positive result for the non-technical domain is due to the fact that the Persona's realization as a workman is more appropriate to technical descriptions than to person-related descriptions.

Concerning the Persona's learning effect, we found no significant difference between the Persona and the No-Persona version ($t(26) = -0.73$; $p = 0.47$ for the technical domain, $t(26) = 0.82$; $p = 0.42$) for the non-technical domain). That is, the Persona contributed neither to the students' comprehension of the technical matters in the pulley experiment nor to their recall capabilities in the second experiment (see Table 3).

As a possible reason, we indicate that we only exploited Persona behaviours 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

[1] Because of technical difficulties, the data for two subjects had to be discarded.

Table 2
Means for the Persona-specific questions asked in the questionnaire (Part B)

Question	Type of information	
	Technical information	Person descriptions
Persona's behaviour is tuned to presentation	3.00	1.64
Persona helps concentration on relevant parts	2.70	2.00
Persona distracts subject from relevant information	1.00	0.93
Persona encourages subject to pay more attention to presentation	2.21	2.00

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 (third row of Table 2).

7. Related work

Efforts toward the creation of dynamic hypermedia presentations have also been made by the adaptive hypertext and natural-language generation community. As examples, we refer to the ELM-ART [8], ILEX [13] and PEBA-II [9] systems. Like PPP, these systems automatically compose hypertext from canned text and items from a knowledge base. The links between the single pages are automatically created on the basis of the user profile and the current situation. However, unlike PPP, they don't allow for the integration of dynamic media, and don't use a lifelike character to guide the user through a presentation.

A second line of research which is of interest to our work consists of approaches to control the behaviour of animated interface agents.

Closely related to our work is Microsoft's Persona project in which the interface agent is a parrot named Peedy (cf. [7]). Nevertheless, Peedy is an anthropomorphic character since it interacts with the user in a natural-language dialogue, and also mimics some non-verbal (human) communicative acts, e.g., Peedy raises a wing to the ear in case speech recognition fails. Since Peedy is to act as a conversational assistant (at least for the sample application, a retrieval system for music CDs), the system comprises components for processing spoken language, dialogue management and the generation of audiovisual output. However, the system doesn't have to create presentation scripts since the presentation of material is restricted to playing the selected CDs.

Lester and Stone [21] have combined a coherence-based behaviour-sequencing engine to control the behaviour of *Herman the Bug*, the pedagogical agent of *Design a Plant*. This engine dynamically selects and assembles behaviours from a behaviour space consisting of animated segments and audio clips. This material has been manually designed by a multidisciplinary team of graphic artists, animators, musicians and voice specialists. On the one hand, this allows the authoring of high quality presentations as the human author has much control over the material to be presented. On the other hand, enormous effort by the human author is required to produce the basic repertoire of a course. In contrast to their work, our approach aims at a higher degree of automation. The basic animation units from which a presentation is built correspond to very elementary actions, such as taking a step or lifting one's arm, which are flexibly combined by the Persona Engine. Furthermore, we don't rely on prestored audio clips, but use a speech synthesizer to produce verbal output.

Table 3
Means for comprehension and recall performance by Persona condition and type of information

Type of information	Persona condition	
	No Persona	Persona
Technical material	36.14	37.57
Person descriptions	11.43	10.35

Rickel and Johnson [19] have developed a pedagogical agent called Steve based on the Jack software, a tool for modelling 3D virtual humans [5]. Instead of creating animation sequences for a course offline and putting them dynamically together as in Design a Plant, the 3D character Steve is directly controlled by commands such as ''look at'', ''walk to'' or ''grasp an object''. In this case, the character interacts with virtual objects in the same way as a human will do in a real environment with direct access to the objects. In contrast to this, our system strictly distinguishes between domain and presentation objects. That is, the PPP Persona is part of a multimedia presentation and interacts with domain objects via their depictions or descriptions. This setting is similar to a setting in which a tutor presents and comments on slides or transparencies.

Similar applications have been described by Noma and Badler [16], who developed a virtual human-like presenter based on the Jack software, and by Thalmann and Kalra [23], who produced some animation sequences for a virtual character acting as a television presenter. While the production of animation sequences for the TV presenter requires a lot of manual effort, the Jack presenter receives input at a higher level of abstraction. Essentially, this input consists of text to be uttered by the presenter and commands, such as *pointing* and *rejecting*, which refer to the presenter's body language. Nevertheless, the human author still has to specify the presentation script, while our system computes this automatically, starting from a complex presentation goal. However, since our presentation planner is application independent, it may also be used to generate presentation scripts for the Jack presenter or the TV presenter.

Perlin and Goldberg [17] have developed an 'English style' scripting language called IMPROV for authoring the behaviour of animated actors. To a certain extent, the library of agent scripts in their approach can be compared to the repertoire of presentation strategies in our approach since they both allow for the organization of behaviours into groups. However, their scripts are represented as a sequence of actions or other scripts while we exploit the full set of Allen relationships. A novelty of our system is that it doesn't require the human author to specify the desired temporal constraints between the single presentation acts, but computes this information dynamically from a complex presentation goal. Furthermore, our system not only designs presentation scripts, but also assembles the multimedia material to be presented to the user.

8. Technical data

The Persona Engine and the Presentation Planner have been implemented in Java™ and C++. While the Presentation Planner resides on the server side, the Persona Engine is downloaded as a Java applet to the user's web browser and receives presentation scripts from the server. Vice versa, user input can be sent to the presentation server to be considered in the further design process.

The Persona Engine relies on about 250 frames for each Persona. Currently we use two cartoon personas and three real personas composed of grabbed video material. To control the behaviour of the personas, more than 150 different behaviours and about 70 presentation strategies have been defined.

9. Conclusion

We have argued that the use of lifelike characters is a promising option for presentations on the web. In order to describe such presentations, we combined a behaviour model for lifelike 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 lifelike characters and navigation structures to allow the user to dynamically change the course of a presentation at runtime.

While our evaluation study did not support the assumption that lifelike agents improve task comprehension and information recall capabilities of human presentation consumers, it clearly revealed a strong affective impact. Our subjects rated learning tasks presented by the Persona as less difficult than presentations without a lifelike character. Obviously, however, this effect does not occur in all applications, and users seem to have clear preferences about when to have a personified agent in the interface. Thus, user interface designers should not only take into account inter-individual but also intra-individual differences.

So far, the focus of our work has been on presentation design. Clearly, the overall quality of the Persona's presentations does not depend solely on its appearance and skills, but to a large extent also on the information gathered from the web. Unfortunately, a presentation agent cannot anticipate which information will be available on the web; i.e., it also has to operate in unknown environments. There are several approaches to tackle this issue. One direction is to rely on methods for information retrieval and extraction. However, we are still far away from robust approaches capable of analysing arbitrary webpages consisting of heterogeneous media objects such as text, images and video. Another approach uses so-called annotated environments (see [10]) which provide the knowledge agents need to perform their tasks appropriately. These annotations can be compared to markups of a webpage. Our hope is that, with the increasing popularity of agents, a standard for such annotations will be developed which will significantly ease the presentation planning process.

We plan to extend our work on presentation agents in several directions. First of all, we are currently defining strategies for presentations involving more than one agent. This extension allows 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 lifelike agents and so-called avatars will become the inhabitants of these worlds. While the audiovisual 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 started with the experimental design of a study to evaluate the Persona's effect on the user's navigation behaviour. In particular, we would like to investigate how far recommendations given by different Personas are followed by the user.

Acknowledgements

This work has been supported by the BMBF under contracts ITW 9400 7 and 9701 0. We would like to thank Peter Rist for drawing the cartoons, H.-J. Proftlich and M. Metzler for the development of the temporal reasoner, Frank Biringer 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) (1983) 832–843.

[2] E. André, W. Finkler, W. Graf, T. Rist, A. Schauder, W. Wahlster. WIP: the automatic synthesis of multimodal presentations. In: M. Maybury (Ed.), Intelligent Multimedia Interfaces, AAAI Press, 1993, pp. 75–93.

[3] E. André, T. Rist. Coping with temporal constraints in multimedia presentation planning. In: Proceedings of AAAI–96, vol. 1. Portland, Oregon, 1996, pp. 142–147.

[4] E. André, T. Rist, J. Müller. Integrating reactive and scripted behaviors in a lifelike presentation agent. In: Proceedings of the Second International Conference on Autonomous Agents (Agents '98), Minneapolis/St Paul, 1998, pp. 261–268.

[5] N.I. Badler, C.B. Phillips, B.L. Webber. Simulating Humans: Computer Graphics, Animation and Control. Oxford University Press, New York, Oxford, 1993.

[6] G. Ball. Dialogue initiative in a web assistant. In: Proceedings of Life-Like Computer Characters '96, Snowbird, UT, 1996.

[7] G. Ball, D. Ling, D. Kurlander, J. Miller, D. Pugh, T. Skelly, A. Stankosky, D. Thiel, M. van Dantzich, T. Wax. Lifelike computer characters: the persona project at Microsoft. In: J.M. Bradshaw (Ed.), Software Agents. AAAI/MIT Press, Menlo Park, CA, 1997, pp. 191–222.

[8] P. Brusilovsky, E. Schwarz, G. Weber. Elm-art: an intelligent tutoring system on world wide web. In: C. Frasson, G. Gauthier and A. Lesgold (Eds.), Intelligent Tutoring Systems (Lecture Notes in Computer Science, vol. 1086). Springer, Berlin, New York, Heidelberg, 1996, pp. 261–269.

[9] R. Dale, M. Milosavljevic. Authoring on demand: natural language generation in hypermedia documents. In: Proceedings of the First Australian Document Computing Symposium (ADCS '96), Melbourne, March 1996, pp. 20–21.

[10] P. Doyle, B. Hayes-Roth. Agents in annotated worlds. In: Proceedings of the Second International Conference on Autonomous Agents (Agents '98), Minneapolis/St Paul, 1998, pp. 173–180.

[11] S. Gibbs, C. Breiteneder, V. de Mey, M. Papathomas. Video widgets and video actors. In: Proceedings of the UIST '93 (ACM SIGGRAPH Symposium on User Interface Software and Technology), Atlanta, GA, 1993, pp. 179–184.

[12] L. Hardman, D.C.A. Bulterman, G. van Rossum, The Amsterdam hypermedia model: adding time and context to the Dexter model, Communications of the ACM 37 (2) (1994) 50–62.

[13] A. Knott, C. Mellish, J. Oberlander, M. O'Donnell. Sources of flexibility in dynamic hypertext generation. In: Proceedings of the 8th International Workshop on Natural Language Generation, Sussex, 1996.

[14] J.C. Lester, S. Converse, S. Kahler, T. Barlow, B. Stone, R. Bhogal. The persona effect: affective impact of animated pedagogical agents. In: Proceedings of CHI '97, Atlanta, GA, 1997, pp. 359–366.

[15] S. Van Mulken, E. André, J. Müller. The persona effect: how substantial is it? In: Proceedings of HCI '98, Sheffield, UK, to appear.

[16] T. Noma, N.I. Badler. A virtual human presenter. In: Proceedings of the IJCAI–97 Workshop on Animated Interface Agents: Making Them Intelligent, Nagoya, 1997, pp. 45–51.

[17] K. Perlin, A. Goldberg, Improv: a system for scripting interactive actors in virtual worlds, Computer Graphics 28 (3) (1996).

[18] C.S. Pinhanez, K. Mase, A. Bobick. Interval scripts: a design paradigm for story-based interaction systems. In: CHI97 Conference Proceedings, Atlanta, GA, March 1997, pp. 287–294.

[19] J. Rickel, W.L. Johnson. Integrating pedagogical capabilities in a virtual environment agent. In: Proceedings of the First International Conference on Autonomous Agents, Marina del Rey, ACM Press, 1997, pp. 30–38.

[20] T. Rist, E. André, J. Müller. Adding animated presentation agents to the interface. In: J. Moore, E. Edmonds and A. Puerta (Eds.), Proceedings of the 1997 International Conference on Intelligent User Interfaces, Orlando, FL, 1997, pp. 79–86.

[21] B.A. Stone, J.C. Lester. Dynamically sequencing an animated pedagogical agent. In: Proceedings of AAAI–96, vol. 1. Portland, OR, 1996, pp. 424–431.

[22] A. Takeuchi, K. Nagao. Communicative facial displays as a new conversational modality. In: Proceedings of ACM/IFIP INTERCHI '93, 1993, pp. 187–193.

[23] N. Magnenat Thalmann, P. Kalra. The simulation of a virtual TV presenter. In: Computer Graphics and Applications. World Scientific, 1995, pp. 9–21.

[24] J.M. Walker, L. Sproull, R. Subramani. Using a human face in an interface. In: Proceedings of CHI–94, Boston, MA, 1994, pp. 85–91.