7

virtual environments

From a user perspective, virtual environments offer the most advanced interface to multimedia information systems. Virtual environments involve the use of (high resolution) 3D graphics, intuitive interaction facilities and possibly support for multiple users. In this chapter, we will explore the use of (desktop) virtual environments as an interface to (multimedia) information systems. We will discuss a number of prototype implementations illustrating, respectively, how paintings can be related to their context, how navigation may be seen as a suitable answer to a query, and how we can define intelligent agents that can interact with the information space. Take good notice, the use of virtual environments as an interface to information systems represents a major challenge for future research!

7.1 virtual context

Imagine that you walk in a museum. You see a painting that you like. It depicts the Dam square in 17th century Amsterdam. Now, take a step forwards and suddenly you are in the middle of the scene you previously watched from some distance. These things happen in movies.

Now imagine that you are walking on the Dam square, some sunday afternoon in May 2001, looking at the Royal Palace, asking yourself is this where Willem-Alexander and Maxima will get married. And you wonder, what did this building and the Dam square look like three centuries ago. To satisfy your curiosity you go to the Royal Museum, which is only a half hour walk from there, and you go to the room where the 17th century city-scape paintings are. The rest is history.

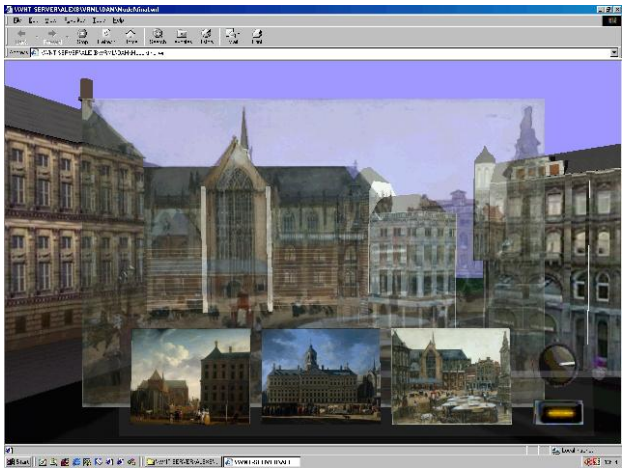
We can improve on the latter scenario I think. So let's explore the options. First of all, we may establish that the Dam square represents a rich information space. Well, the Dam Square is a 'real world' environment, with it has 700 years of (recorded) history. It has a fair amount of historical buildings, and both buildings and street life have changed significantly over time.

So, we can rephrase our problem as

how can we give access to the 'Dam square' information space

But now we forget one thing. The idea underlying the last scenario is that we somehow realize a seamless transition from the real life experience to the information space. Well, of course, we cannot do that. So what did we do?

Look at the screenshot from our *virtual context* prototype. You can also start the VRML demo version that is online, by clicking on the screenshot. What you see is (a model of) the Dam square, more or less as it was in 2001. In the lower part, you see a panel with paintings. When you click on one of these painting, your viewpoint is changed so that you observe the real building from the point of view from which the painting was made. Then using the controls to the right of the panel, you can overlay the real building with a more or less transparent rendering of the painting. You can modify the degree of transparency by turning the dial control. You may also make the panel of paintings invisible, so that it does not disrupt your view of the Dam and the chosen overlay.



In other words, we have a VR model of Dam square and a selection of related paintings from the Royal Museum, that are presented in aa panel from which the user can choose a painting. We deploy viewpoint adjustment, to match the selected painting, and we use overlay of paintings over buildings, in varying degrees of transparancy, to give the user an impression of how the differences between the scene depicted in the painting and the actual scene in (the virtual) reality.

We have chosen for the phrase *virtual context* to characterize this prototype, since it does express how virtual reality technology enables us to relate an information space to its original context.

From the perspective of virtual reality, however, we could also have characterized our prototype as an application of *augmented virtual reality*, since what we have is a virtual reality model of a reallife location that is augmented with information that is related to it, (almost) without disrupting the virtual reality experience. In summary, we may characterize our approach as follows.

augmented virtual reality

- give user sense of geographic placement of buildings

- show how multiple objects in a museum relate to eachother
- show what paintings convey about their subject, and how

Considering the fact that many city-scape paintings of Amsterdam have been made, many of which are in the Royal Museum, and that paintings may say many things about their subject, we believe that our approach is viable for this particular instance. The augmented virtual reality approach would also qualify as a possible approach to cultural heritage projects, provided that sufficient pictorial material is available or can be reconstructed.

Although we were quite satisfied with what we accomplished, there are still many things that can be done and also a number of open problems. Guided tours are a wellknown phenomenon. But how to place them in our virtual context is not entirely clear. As another problem, our approach does not seem suited to account for buildings that do no longer exist. Another thing we have to study is how to change the temporal context, that is for example change from a model of the dam in 2001 to a model of the Dam in 1850. We would then also like to have 'viewpoint transitions' over space and time!

Finally, to give better access to the underlying information space we must also provide for textual user queries, and find an adequate response to those queries.

¡name a=ex-t-7-1¿

VRML To realize our prototype we used VRML, which limits us to medium quality desktop VR. At this stage, VRML is a good option, since it is a relatively stable format with a reasonable programmatic model. In short, what VRML offers is

VRML

- declarative means for defining geometry and appearance
- prototype abstraction mechanism
- powerful event model
- relatively strong programmatic capabilities

Although VRML allows for writing models (including geometry and appearance) using a plain text editor, many tools support export to VRML. As a consequence, often tools are used to create more complex models.

In addition, VRML allows for defining prototype abstractions, so reuse of models and behavior can be easily realized.

Defining dynamic behavior involves the routing of events that may come from a variety of built-in sensors (for example a TimeSensor for animations) to scripts or so-called interpolators, that allow for the manipulation of geometry and appearance parameters of the model.

In particular, the use of scripts or the *External Authoring Interface* (EAI), that allows for defining behavior in Java, is essential for realizing complex behavior.

Summarizing, VRML is a sufficiently rich declarative language for defining 3D scenes, with a relatively powerful programming model for realizing complex behavior. Some may think that VRML is dead. It isn't. The underlying model is endorsed in both the X3D and RM3D standards, simply since it has proven its worth.

research directions– *augmented virtuality*

Given an information space, there is a duality between information and presentation. For an audience or user to be able to digest a presentation, the amount of information must be limited. Effective presentation, moreover, requires the use of proper rethorics (which may be transcoded as *ways of presenting*) that belong to the medium. Using VR, which is (even in its desktop format) a powerful presentation vehicle, one should always beware of the question *what is it good for?* Generally one may ask, what is the added value of using VR? In an abstract fashion the answer should be, to bridge the gap between information content and presentation. Or, in other words, to resolve the duality between information and presentation!

Let's look at an example, a site about archeology, announced as a site offering *Virtual Archeology*. Perhaps it is good to bring to your attention that the *virtual*, in Computer Science, means nothing but another level of indirection to allow for a (more) flexible usage of entities or objects. See [OO], section 1.2.

virtual archeology

- variety of archeological sites
- various paths through individual site
- reconstruction of 'lost' elements
- 'discovery' of new material
- glossary – general background knowledge

For a site about archeology, *virtual* means the ability to present the information in a number of ways, for example as paths through a particular site, with the possibility to explore the reconstruction of lost or perished material, and (for students) to discover new perspectives on the material. In addition, for didactic reasons there may also be a glossary to explain concepts from archeology.

Now, how would you construct such a site about virtual archeology? As a collection of HTML pages and links? It seems that we can do better, using VR and rich interaction mechanisms!

So, what is meant by *augmented virtuality*? Nothing that hasn't been expressed by the notion of *augmented virtual reality*, of which an example has been given in this section. The phrase *augmented virtuality* itself is just one of those potentially meaningless fancy phrases. It was introduced simply to draw your attention to the duality between information and presentation, and to invite you to think about possible ways to resolve this duality.

7.2 navigation by query

Virtual worlds form (in itself) a rich repository of multimedia information. So, when working on the musical feature detector, sketched in sectin 6.3, the thought occurred to ask funding for a research project on information retrieval in virtual worlds. This project is called RIF, which stands for

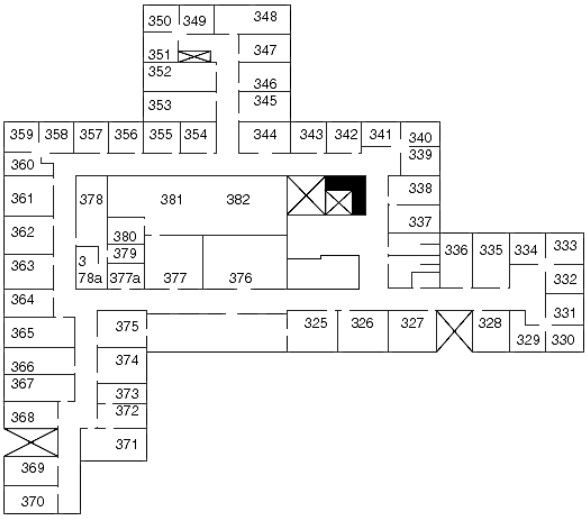
RIF

For the RIF project, we decided to develop a small multi-user community of our own, using the *blaxxun* Community Server. Then, during the development of our own virtual environment, the question came upof how to present the results of a query to the user. The concept we came up with was *navigation by query*, and in this section we will look at the prototype we developed to explore this concept.

case study – CWI

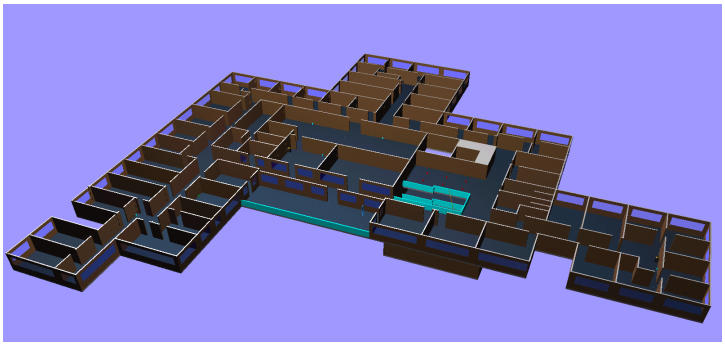
For our prototype, we took one of the worlds of our virtual environment, the third floor of the CWI. The reason for this is that we were (at the time) doing our research there, and so there were plenty locations of interest, such as the rooms of our collegues, the printer room, and not to forget, the coffee corner.

the map



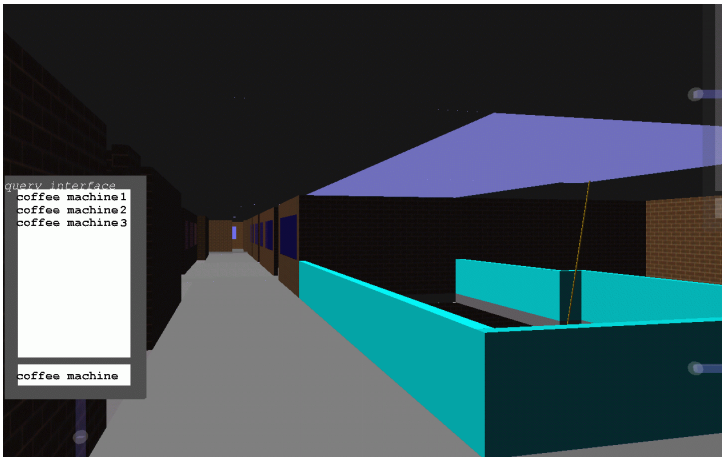
We started out by taking a map of the third floor, and developed a model of it, using a tool developed by a student, who needed such a tool for realizing his game *Out of the Dark*.

the model



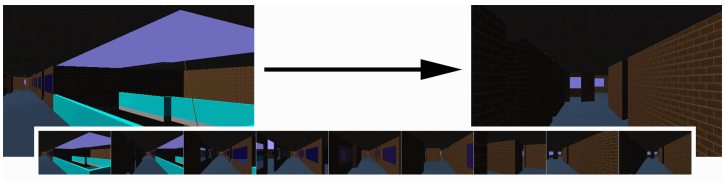
When dwelling around in (this part of) our virtual environment, the user may pose (arbitrary) queries, for example *where is the coffee machine*.

the query



Remind, that after a few hours of research, coffee might be needed to get fresh ideas!

navigation



As a result, the user is then so to speak taken by the hand and led to one of the coffee machines that can be found on the third floor. In effect, with knowledge of the layout of the building a viewpoint transformation is executed, in a tempo that allows the user to

explore and discover

the (model of the) third floor of the CWI.

The idea is rather straightforward. Some have asked us why *navigation by query* might be useful. Well, simply, it seems to offer an interesting alternative to navigation by explicit interaction and navigation in the formof a guided tour. Our primary goal in developing the prototype, however, was to see whether navigation by query is feasible, and under what conditions.

information in virtual worlds

Developing the prototype has forced us to think more explicitly about what information is available in virtual worlds, and (perhaps more importantly) how to gain access to it. So the question we asked ourselves was

what are we searching for?

Now, in a virtual world, such as the ones built with VRML, we can distinguish between the following types of information: viewpoints, that is positions in the world from where interesting things can be looked at or accessed in any other way; areas of interest, where those intersting things are located; objects, that may provide information or offer particular kinds of functionality; persons, that is other users thatare visiting the world; and even text, which might be on billboards or slides.

Some of this information is, so to speak, hard-wired in the model and may be accessed anytime, in some cases even by scanning the VRML file. Other information, however, is of a more dynamic nature, since it might be due to the presence of multiple users, the execution of scripts, or events that happen in response to user interaction. Some information may even be explicitly hidden, such as for example the actions one should take in solving a puzzle or playing a game.

When the virtual world is loaded, all the information (or at least most of it) is present in the so-called scenegraph, the structure thatis built to render the world. Using the software interface to access the scengraph (which is usually browser-specific), we can look for annotations, node types and textual content to extract information from the world. This information may then be stored in a database, and be reused later for other users and queries. In principle, more advanced techniques could be used to extract information from the materials used, and even from textures and geometry.

presentation issues

In our prototype, we aimed at solving the question how to present the results of a query, using navigation. First of all, we had to

choose a metaphor

for navigation. Dependent on the object of interest a viewpoint can be selected. For a viewpoint, it is just that viewpoint. For an area of interest, the viewpoint

selected must enable the user to view the area, and when objects or persons are chosen, care must be taken not to block the users' view by some obstacle.

Now answering a query then comes down to planning a suitable route and apply a series of viewpoint transformations along that route.
Not surprisingly, the navigation metaphor we chose was

walking

as the preferred mode of viewpoint transformtions.

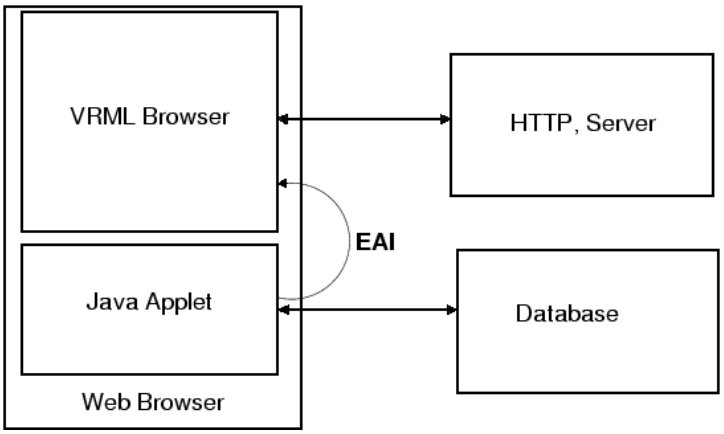
the prototype

The structure of the prototype is depicted in the figure below.

In realizing the prototype,we made the following (simplifying) assumptions.

We avoided a number of difficulties by choosing for explicit annotations (which indicate locations and areas of interest), and by avoiding the intricacies of route planning and advanced text processing.

The requirements laid down before hand just stated that we would have a database and that we would avoid superfluous user interface elements. Instead, we used control and input panels written in VRML, in order to provide a 3D(pseudo-immersive) interface.



Now, our assumptions may in principle be relaxed. For example, annotation might be done incrementally by users that visit the world or to some extent even automatically, by using feature extractors. Instead if explicit maps, we may dynamically create maps based on users' navigation patterns. And, instead of simple keyword matching, we may apply more advanced text retrieval techniques. But this is left as future work. Anyway, we were satisfied that we could state the following conclusions:

conclusions

- navigation by query is feasible and may help users to find locations and objects
- determining suitable navigation routes without an explicitly defined map is hard

As is often the result with good research, you solve one problem and a number of other problems come up. So, one of the questions that remains was: how can we improve on navigation? What additional navigation support can we provide?

research directions– *extended user interfaces*

Is desktop VR a suitable candidate as an interface technology for multimedia information systems? And if so, what needs to be done to apply this technology effectively?

At first sight, our vision of applying VR as an interface to multimedia systems seems to be doomed to fail. As Ben Schneiderman, in a keynote for the Web3D Symposium 2002, observes:

3D GUI

> *Wishful thinking about the widespread adoption of three-dimensional interfaces has not helped spawn winning applications. Success stories with three-dimensional games do not translate into broad acceptance of head-tracking immersive virtual reality. To accelerate adoption of advanced interfaces, designers must understand their appeal and performance benefits as well as honestly identify their deficits. We need to separate out the features that make 3D useful and understand how they help overcome the challenges of dis-orientation during navigation and distraction from occlusion.*
>
> Ben Shneiderman

So, even if advanced (3D) user interfaces might be useful, there are a number of questions to raise. Again, following Ben Schneiderman:

> *Does spatial memory improve with 3D layouts? Is it true that 3D is more natural and easier to learn? Careful empirical studies clarify why modest aspects of 3D, such as shading for buttons and overlapping of windows are helpful, but 3D bar charts and directory structures are not. 3D sometimes pays off for medical imagery, chemical molecules, and architecture, but has yet to prove beneficial for performance measures in shopping or operating systems.* Ben Shneiderman

In particular, according to Schneiderman, we must beware of *tacky 3D*, gadgets in 3D space that are superfluous and only hindering the user to perform a task. Well-spoken and based on adequate observations! Nevertheless, at this stage, we should (in my opinion) adopt a slightly more liberal attitude and explore in what ways the presentation of (multimedia) information could be augmented by using (desktop) VR. But enough about *augmentation*. Let's discuss technology, and investigate what is required for the effective deployment of VR from the point of view of intelligent agents!

7.3 intelligent agents

Visitors in virtual environments are often represented by so-called avatars. Wouldn't it be nice to have intelligent avatars that can show you around, and tell you more about the (virtual) world you're in.

Now, this is how the idea came up to merge the RIF project, which was about information retrieval, with the WASP project, another acronym, which stands for:

WASP

Web Agent Support Program

The WASP project aims at realizing intelligent services using both client-side and server-side agents, and possibly multiple agents. The technical vehicle for realizing agents is the language DLP, which stands for

DLP

Distributed Logic Programming

Merging the two projects required providing the full VRML EAI API in DLP, so that DLP could be used for programming the dynamic aspects of VRML worlds.

background Historically, the WASP project precedes the RIF project, but we started working on it after the RIF project had already started. Merging these two projects had more consequences than we could predict at the time. The major consequence is that we shifted focus with respect to programming the dynamcis of virtual environments. Instead of scripts (in Javascript), Java (through the EAI), and even C++ (to program *blaxxun* Community Server extensions), we introduced the distributed logic programming language DLP as a uniform computational platform. In particular, for programming inteligent agents a logic programming language is much more suited than any other language. All we had to do was merge DLP with VRML, which we did by lifting the Java EAI to DLP, so that function calls are available as built-ins in the logic programming language.

When experimenting with agents, and in particular communication between agents, we found that communication between agents may be used to maintain a shared state between multiple users. The idea is simple, for each user there is an agent that observes the world using its 'sensors' and that may change the world using its 'effectors'. When it is notified by some other agent (that is co-located with some other user) it can update its world, according to the notification. Enough background and ideas. Let's look at the prototypes that we developed.

multi-user soccer game

To demonstrate the viability of our approach we developed a multi-user soccer game, using the DLP+VRML platform.

We chose for this particular application because it offers us a range of challenges.

multi-user soccer game

- *multiple (human) users* – may join during the game
- *multiple agents* – to participate in the game (e.g. as goalkeeper)
- *reactivity* – players (users and agents) have to react quickly
- *cooperation/competition* – requires 'intelligent' communication
- *dynamic behavior* – sufficiently complex 3D scenes, including the dynamic behavior of the ball

Without going into detail, just imagine that you and some others wish to participate in a game, but there are no other players thatwant to join. No problem, we just add some intelligent agent football players. And they might as well be taken out when other (human) players announce themselves.

For each agent player, dependent on its role (which might be *goal-keeper*, *defender*, *mid-fielder* and *forward*), a simple coginitive loop is defined: sensing, thinking, acting. Based on the information the agent gets, which includes the agent's position, the location of the ball, and the location of the goal, the agents decides which action to take. This can be expressed rather succinctly as rules in the logic programming formalism, and also the actions can be effected using the built-in VRML functionality of DLP.

Basically, the VRML-related built-ins allow for obtaining and modifying the values of *control points* in the VRML world.

control points

- get/set – position, rotation, viewpoint

These control points are in fact the identifiable nodes in the scenegraph (that is, technically, the nodes that have been given a name using the DEF construct).

This approach allows us to take an arbitrarily complex VRML world and manipulate it using the control points. On the other hand, there are also built-ins that allow for the creation of objects in VRML. In that case, we have much finer control from the logic programming language.

All in all we estimate that, in comparison with other approaches, programming such a game in DLP takes far less time than it would have taken using the basic programming capabilities of VRML.

agents in virtual environments

Let us analyse in somewhat more detail why agents in virtual environments may be useful. First of all, observe that the phrase *agents in virtual environments* has two shades of meaning:

agents in virtual environments

- virtual environments with embedded autonomous agents
- virtual environments supported by ACL communication

where ACL stands for *Agent Communication Language*. Our idea, basically is to use an ACL for realizing shared objects, such as for example the ball in the soccer game.

The general concept op multi-user virtual environments (in VRML) has been studied by the *Living Worlds Working Group*. Let's look at some definitions provided by this working group first. A *scene* is defined as a geometrically bounded, continuously navigable part of the world. Then, more specifically a *world* is defined as a collection of (linked) scenes.

Now, multi-user virtual environments distinguish themselves from single-user virtual environments by allowing for so-called *Shared Objects* in scenes, that is objects that can be seen and interacted with by multiple independent users, simultaneously. This requires synchronization among multiple clients, which may either be realized through a server or through client-to-client communication.

Commonly, a distinction is made between a *pilot* object and a *drone* object.

Shared Object

- *pilot* – instance that will be replicated
- *drone* – instance that replicates pilot

So, generally speaking, pilot objects control drone objects. There are many ways to realize a pilot-drone replication scheme. We have chosen to use agent technology, and correspondingly we make a distinction between *pilot agents*, that control the state of a shared object, and *drone agents*, that merely replicate the state of a shared object.

Since we have (for example in the soccer game) different types of shared objects, we make a further distinction between agents (for each of which there is

a pilot and a drone version). So, we have *object agents*, which control a single shared object (like the soccerball). For these agents the pilot is at the server, and the drone is at the client. We further have agents that control the users' avatars, for which the pilot at user/client side, and the drone either at the server or the client. Finally, we have autonomous agents, like football players, with their own avatar. For those agents, the pilot is at the server, and the drones at the clients.

Now, this classification of agents gives us a setup that allows for the realization of shared objects in virtual environments in an efficient manner. See [Community] for details.

The programming platform needed to implement our proposal must satisfy the following requirements.

programming platform

- VRML EAI support
- distributed communication capabilities (TCP/IP)
- multiple threads of control – for multiple shared objects
- declarative language – for agent support

So, we adapted the distributed logic programming language DLP (which in its own right may be called an agent-oriented language *avant la lettre*), to include VRML capabilities. See the online reference to the AVID project for a further elaboration of these concepts.

PAMELA

The WASP project's chief focus is to develop architectural support for web-aware (multi) agent systems. So, when we (finally) got started with the project we developed a taxonomy along the following dimensions:

taxonomy of agents

- 2D/3D – to distinguish between text-based and avatar embodied agents
- client/server – to indicate where agents reside
- single/multi – as a measure of complexity

A classification along these dimensions results in a lattice, with as the most complex category a *3D-server-multi-agent system*, of which the distributed soccer game is an example. See [Taxonomy].

When we restrict ourselves to *3D-client-single-agent systems*, we may think of,for example, navigation or presentation agents, that may help the user to roam around in the world, or that provide support for presenting the results of a query as objects in a 3D scene.

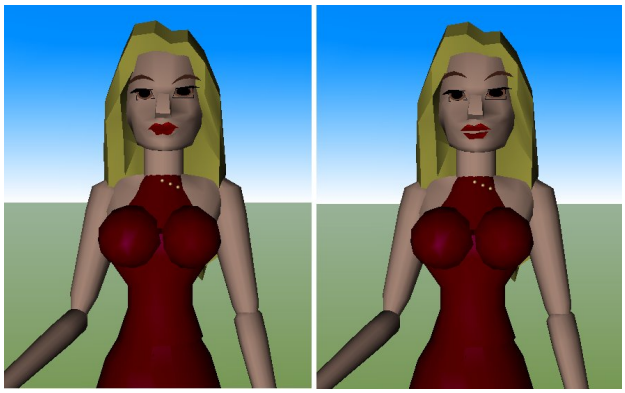
Our original demonstrator for the WASP project was an agent of the latter kind, with the nickname *PAMELA*, which is an acronym for:

PAMELA

Personal Assistent for Multimedia Electronic Archives

The PAMELA functional requirements included: autonomous and on-demand search capabilities, (user and system) modifiablepreferences, and multimedia presentation facilities. It was, however, only later that we added the requirement that PAMELA should be able to live in 3D space.

In a similar way as the soccer players, PAMELA has control over objects in 3D space. PAMELA now also provides animation facilities for its avatar embodiment.



To realize the PAMELA representative, we studied how to effect facial animations and body movements following the *Humanoid Animation Working Group* proposal.

H-Anim

- control points – joints, limbs and facial features

The H-Anim proposal lists a number of control points for (the representation of the) human body and face, that may be manipulated upto six degrees of freedom. Six degrees of freedom allows for movement and rotation along any of the X,Y,Z axes. In practice, movement and rotation for body and face control points will be constrained though.

presentation agent Now, just imagine how such an assistant could be of help in multimedia information retrieval.

presentation agent

> Given any collection of results, PAMELA could design some spatial layout and select suitableobject types, including for example color-based relevance cues, to present the results in a scene. PAMELA could then navigate you through the scene, indicating the possible relevance of particular results.

persuasion games But we could go one step further than this and, taking inspiration fromthe research field of *persuasive technology*, think about possible

persuasion games we could play, using the (facial and body) animation facilities of PAMELA:

persuasion games

- single avatar persuasive argumentation
- multiple avatar dialog games

Just think of a news readerpresenting a hot news item. or a news reader trying to provoke a comment on some hot issue. Playing another trick on the PAMELA acronym, we could think of

PAMELA

Persuasive Agent with Multimedia Enlightened Arguments

I agree, this sounds too flashy for my taste as well. But, what this finale is meant to express is, simply, that I see it as a challenge to create such synthetic actors using the DLP+VRML platform.

research directions– *embodied conversational agents*

A variety of applications may benefit from deploying embodied conversational agents, either in the form of animated humanoid avatars or, more simply, as a 'talking head'. An interesting example is provided by *Signing Avatar*, a system that allows for translating arbitrary text in both spoken language and sign language for the deaf, presented by animated humanoid avatars. Here the use of animated avatars is essential to communicate with a particular group of users, using the sign language for the deaf.

Other applications of embodied conversational agents include e-commerce and social marketing, although in these cases it may not always be evident that animated avatars or faces actually do provide added value.

Another usage of embodied conversational agents may be observed in virtual environments such as Active Worlds, *blaxxun* Community and Adobe Atmosphere. Despite the rich literary background of such environments, including Neil Stephenson's Snow Crash, the functionality of such agents is usually rather shallow, due to the poor repertoire of gestures and movements on the one hand and the restricted computational model underlying these agents on the other hand. In effect, the definition of agent avatars in virtual environments generally relies on a proprietary scripting language which, as in the case of *blaxxun* Agents, offers only limited pattern matching and a fixed repertoire of built-in actions.

In contrast, the scripting language for *Signing Avatar* is based on the H-Anim standard and allows for a precise definition of a complex repertoire of gestures, as exemplified by the sign language for the deaf. Nevertheless, also this scripting language is of a proprietary nature and does not allow for higher-order abstractions of semantically meaningful behavior.

scripting behavior In this section we introduced a software platform for agants. This platform not only offers powerful computational capabilities but also an expressive scripting language (STEP) for defining gestures and driving the behavior of our humanoid agent avatars.

The design of the scripting language was motivated by the requirements listed below.

STEP

- *convenience* – for non-professional authors
- *compositional semantics* – combining operations
- *re-definability* – for high-level specification of actions
- *parametrization* – for the adaptation of actions
- *interaction* – with a (virtual) environment

Our scripting language STEP meets these requirements. STEP is based on dynamic logic [DL] and allows for arbitrary abstractions using the primitives and composition operators provided by our logic. STEP is implemented on top of DLP,

As a last bit of propaganda:

DLP+X3D

> The DLP+X3D platform provides together with the STEP scripting language the computational facilities for defining semantically meaningful behaviors and allows for a rich presentational environment, in particular 3D virtual environments that may include streaming video, text and speech.

See appendix **??** for more details.

evaluation criteria The primary criterium against which to evaluate applications that involve embodied conversational agents is whether the application becomes more effective by using such agents. Effective, in terms of communication with the user. Evidently, for the *Signing Avatar* application this seems to be quite obvious. For other applications, for example negotiation in e-commerce, this question might be more difficult to answer.

As concerns the embedding of conversationl agents in VR, we might make a distinction between *presentational VR*, *instructional VR* and *educational VR*. An example of educational VR is described in [EducationalVR]. No mention of agents was made in the latter reference though. In instructional VR, explaining for example the use of a machine, the appearance of a conversational agent seems to be quite natural. In presentational VR, however, the appearance of such agents might be considered as no more than a gimmick.

Considering the use of agents in applications in general, we must make a distinction between *information agents*, *presentation agents* and *conversational agents*. Although the boundaries between these categories are not clearcut, there seems to be an increasing degree of interactivity with the user.

From a system perspective, we might be interested in what range of agent categories the system covers. Does it provide support for managing information

and possibly information retrieval? Another issue in this regard could be whether the system is built around open standards, such as XML and X3D, to allow for the incorporation of a variety of content.

Last but not least, from a user perspective, what seems to matter most is the naturalness of the (conversational) agents. This is determined by the graphical quality, as well as contextual parameters, that is how well the agent is embedded in its environment. More important even are emotive parameters, that is the mood and style (in gestures and possibly speech) with which the agents manifest themselves. In other words, the properties that determine whether an agent is (really) convincing.

questions

virtual environments

1. (*) Discuss how *virtual environments* may be used for giving access to *(multimedia) information*. Give a brief characterization of *virtual environments*, and indicate how *information (hyper) spaces* may be projected in a virtual environment.

concepts

2. What is meant by *virtual context*?
3. Give an example of *navigation by query*, and indicate its possible advantages.
4. Discuss the deployment of *(intelligente) navigation agents*.

technology

5. Give a brief characterization of: VRML.
6. What is a *viewpoint transformation*?
7. What kinds of navigation can you think of?
8. How may intelligent avatars be realized? Give an example.