Observing and Adapting User Behavior in Navigational 3D Interfaces

Augusto Celentano and Fabio Pittarello
Dipartimento di Informatica, Università Ca' Foscari di Venezia
Via Torino 155, I-30172 Mestre (VE), Italia
{auce,pitt}@dsi.unive.it

ABSTRACT

In a navigation-oriented interaction paradigm, such as desktop, mixed and augmented virtual reality, recognizing the user needs is a valuable improvement, provided that the system is able to correctly anticipate the user actions. Methodologies for adapting both navigation and content allow the user to interact with a customized version of the 3D world, lessening the cognitive load needed for accomplishing tasks such as finding places and objects, and acting on virtual devices.

This work discusses adaptivity of interaction in 3D environments, obtained through the coordinated use of three approaches: structured design of the interaction space, distinction between a base world layer and an interactive experience layer, and user monitoring in order to infer interaction patterns. Identification of such recurring patterns is used for anticipating users actions in approaching places and objects of each experience class. An agent based architecture is proposed, and a simple application related to consumer e-business is analyzed.

Categories and Subject Descriptors

H.5.1 [**Information Interfaces and Presentation**]: Multimedia Information Systems—*Artificial, augmented, and virtual realities*; H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*Evaluation/methodology*; D.2.2 [**Software Engineering**]: Design Tools and Techniques—*User interfaces*

General Terms

Design, Experimentation, Human Factors.

1. INTRODUCTION

Three-dimensional representations of realistic or abstract worlds and objects are often used as metaphors for representing information access and interaction paradigms. Scientific visualization, desktop virtual reality, mixed and augmented reality are a sample of fields in which 3D representation has been extensively experimented. Applications related to entertainment, cultural heritage, tourism and education have been developed according to a mixture of three-dimensional representation of information environment, navigation oriented interaction and information discovering paradigms. The computational load needed to generate and support navigation and interaction in 3D worlds can be afforded by current hardware and software technology, and is well balanced by the effectiveness of such a representation in the aforementioned applications.

Interaction in 3D-based interfaces is more complex than in WIMP bi-dimensional interfaces. WIMP interfaces offer a synthetic vision of interaction possibilities, progressively drawn to the user attention through a planned sequence of windows and panels in a clearly understandable sequence. The user knows which sequence(s) of actions perform an expected operation. Equivalent results can be obtained in different ways involving different interaction styles, but the number of alternative behaviors is usually small, e.g., menus selection vs. keyboard shortcuts.

Interaction in three-dimensional environments involves an explorative approach and requires typical real-world operations such as moving, turning around objects, changing the point of view without changing the position, and so on. At each step there is a large number of possible actions, and different ways to alternate movement and action upon interaction devices.

Adaptivity has been studied as a possible solution for overcoming usability problems, and customization of different features of the 3D worlds such as geometry and navigation has been proposed. This work tries to lessen the user cognitive effort and to augment the interaction speed by monitoring the user behavior, and exploiting the acquired knowledge for anticipating the user needs in forthcoming interaction. We propose a design methodology based on three elements: a way to organize the three-dimensional space based on the concept of *interaction locus*; a selection mechanisms able to configure the interaction space with one among alternative *classes of experience*; an *agent-based* approach where two classes of agents, bound respectively to the user and to the interaction locus, analyze the user behavior in order to discover recurring interaction patterns, and maintain user- and locus-related contexts. The methodology can be applied to different domains, such as e-commerce, information presentation, virtual communities, cultural heritage, and so on.

The paper is organized as follows: Section 2 reviews the

AVI '04, May 25-28, 2004, Gallipoli (LE), Italy

literature related to interaction issues in 3D worlds. Section 3 presents the interaction locus concept for interactive space structuring. Section 4 discusses the design of classes of experience as a way to give users different interaction opportunities in the same world. Section 5 introduces an agent-based approach for tracing and adapting the interaction to recurring user behavior. Section 6 illustrates the architecture of an adaptive system and a prototype implementation. Section 7 illustrates a case study, and Section 8 summarizes the results achieved.

2. RELATED WORK

In recent times computer-human interaction research has analyzed the issues related to interaction in 3D environments. In particular navigation, that is often the primary interaction in 3D worlds and the prerequisite for more sophisticated activities, has been considered by a remarkable number of studies. The navigation problem has been considered in its three fundamental issues: identification of the scene structure, orientation and navigation itself, which often depends on the knowledge acquired in the previous activities. Different approaches have been proposed for these issues: enforcement of the scene structure [19, 21], insertion of landmarks [24] and different navigation paradigms. A recent work emphasized that users can benefit from tools that guide the user in the scene [14]; constrained navigation, combining guidance with freedom of movement can be in many cases an helpful compromise that avoid user disorientation and enhances usability of 3D worlds.

Customization of the interaction often diminishes the cognitive load for the user; we may distinguish between *adaptive* and *adaptable* techniques that are characterized by different levels of automatic customization of the user experience. Adaptive techniques customize the user's experience according to static user data gathered with preliminary questionnaires and to usage data resulting from the analysis of the user's activity. The granularity of the approach can vary from the identification of classes of users to personalization based on the individual features of the single user. The adaptable approach avoids automatic customization of the user experience; the user must explicitly choose, among different opportunities, the interaction paradigm that best fits his/her needs.

Adaptive approaches [4, 18, 22, 23] are widely used in designing hypermedia and web portals where users are recognized through login and password or cookies, and receive customized versions of the web site. Also adaptable approaches are considered in the web context, such as in the *Avanti* project [11].

Adaptation and personalization of 3D environments is a less explored issue, even though there are several remarkable studies. Recent adaptive techniques propose content customization and allow to generate dynamically personalized 3D worlds [8, 13, 25]. Many proposals offer application examples related to 3D virtual stores, whose content is generated on the basis of user profiles or monitoring their activities.

Other methodologies focus on adapting user interaction; in particular an interesting work focuses on adapting the navigation activity [7]. This work considers a specific usability problem that affects many Web3D worlds, i.e. insufficient user assistance during world exploration for finding objects/places of interests. The study proposes automatically generated tours that take advantage of humanoid characters for guiding the visitors through the 3D scene; these tours can be personalized on the basis of the user interests. Our approach shares with [7] the interest for adapting user interaction. However, our methodology considers specifically also other interactive activities that are adapted on the basis of initial user profile or usage data.

3. THE INTERACTION LOCUS

The *interaction locus* (*IL*) was conceived as a means for giving a structure to the 3D space [21]. The basic idea was to mark in an initially unstructured 3D scene the portions of space that have distinguished morphologic features and that are characterized by specific interaction modalities. It is both a help to recognize a part of the scene that has a visual unity, such as a museum room or a city square, or that is characterized by homogeneous behaviors. For example, in a museum interaction consists mostly in moving from room to room and examining the artworks, while in an industrial plant there are places where some actions are forbidden for security reasons. In general, each environment is characterized by a set of *interaction loci* that in most cases can be organized in classes, corresponding to specific categories of interaction opportunities.

With the interaction locus concept the designer of an interactive experience is enabled to superimpose to a *raw* scene a set of virtual entities whose task is to inform the user about the nature of the part of the scene he/she's entered and to present and to mediate the possible interactions inside the area controlled by it. Each locus contains a number of *interactive objects* that can be manipulated by the users. Also the interactive objects can be categorized in different classes, whose instances can appear in different interaction loci of the same world. Several alternative sets of *IL*s can be associated to a specific scene, according to different parameters such as the user profile or other context variables, in order to offer to different users interaction experiences tailored to their needs.

The interaction locus is denoted by a number of properties which are summarized in the following list.

- It is a connected portion of space where the user can interact with suitable objects and devices. An underlying base world is present, real or virtual. In a virtual environment it is a *base geometry* describing a realistic, simulated or imaginary world. In a mixed reality environment it is the part of the world to which experiences are attached. In a real world with embedded computing devices it is the part of the world in which the user can interact with the devices. The base world is independent from the *IL*, and several *IL*s can be superimposed on it, each corresponding to a different kind of experience the user can live.
- An interaction locus can be hierarchically decomposed in a set of internal sub-loci; conversely, many interaction loci can be aggregated into super-loci. Thus, interaction loci are hierarchically structured, e.g., a museum room, a wall of the room, a set of paintings of a same author, one specific painting. Each level of the hierarchy refines the interaction which can take place at the outer hierarchy level, and defines further specialized interactive behavior. It is therefore a specialization hierarchy in object oriented terms.

- Except for the hierarchical inclusion, interaction loci do not overlap. This requirement is necessary in order to unambiguously define the interaction opportunities at each location of a space. However, several *IL* hierarchies can be defined on a same base world, and alternatively activated for different users or for the same user at different times, giving different interaction experiences.

- Each *IL* has a defined (but possibly invisible) boundary, therefore the interaction management system can perceive when the user enters and exits it.

Identifiable interaction devices populate the *IL*, which support the exchange of information between the user and the world. They are of three basic types:

- *interactive objects*, i.e., objects on which the user operates directly, which change their state or their appearance as a consequence of the user interaction. An example of an interactive object in a cultural heritage context is an artwork revealing information when the user approaches or touches it, e.g., by animating itself;

- *artifacts*, i.e., mediators of the interaction between the user and the world, which make evident to the user interaction opportunities which would otherwise be unknown. An example of an artifact is a metaphoric object acting as a button for activating a task;

- *dynamic information objects*, which modify their state or appearance, e.g., show up or hide, as a consequence of the user interaction on other devices in the environment. An example of information object is a text panel showing information about the place user is visiting.

4. CLASSES OF EXPERIENCE

Interaction that goes beyond the limits of a bi-dimensional screen is denoted as a *post-wimp human computer interaction* [9]. Several paradigms have been proposed, among which interaction in 3D worlds is distinguished by being a *Positional Interaction Paradigm* (*PIP*). In a *PIP* humans participate to the interaction with their real or virtual body; the human position inside the environment is tracked by some input device and considered as one of the main data for interaction. The subjective presence of the humans and their interaction with objects inside the three-dimensional world (that have a physical/virtual location too) is a key feature that distinguishes them from other paradigms where interaction doesn't take into account the location of the user, or where the interactive entities and objects are placed in an abstract workspace [1].

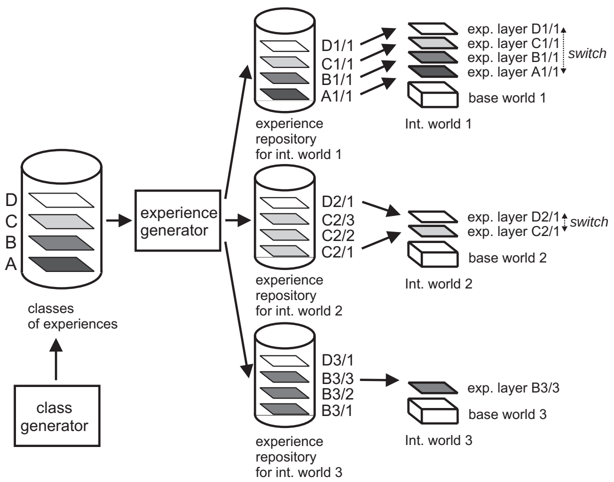


Figure 1: Classes of experience in interactive worlds

If we compare *PIP*s to the common-sense notion of experience, we can notice a number of similarities: experiences are characterized by the direct involvement of humans in the world, and are often the cumulative results of different active phases performed in different places and at different times. But human experiences are richer due to the knowledge enhancement as a result of the experience, and to the possibility of using the new knowledge acquired during the experience as a guide for future experiences.

We have proposed in [6] to enhance interactions allowed by *PIP*s, trying to emulate the positive features of the human experience; therefore we refer to such enhanced interaction activities, in terms of quality and quantity of interaction, as *Experiential Interaction Paradigm* (*EIP*). In *EIP*s the experience of a user interacting with a system becomes an important source of knowledge for the designer.

A fundamental element of the methodology is the concept of *class of experience*. It denotes a typology of interactive experiences general enough to be used as a reference template which can be adapted to specific 3D worlds and users. For example, the class *shopping in a mall* may be specialized to different malls with particular paths among the shops of interest, showing different goods, still with the same basic mechanisms and interaction opportunities.

Each class of experience defines a general scenario for structuring a space in different interaction loci; the scenario is not related to a specific 3D world, but defines a set of rules and behaviors that characterize the same experience typology in different environments. As shown in Figure 1, a *class generator* is used for defining the features of the specific class of experience and for storing the results in a class repository. An *experience generator* shifts from the abstraction level that characterizes the experience class to the specialized definition for a given 3D world. In Figure 1 it is used for defining experiences of three different worlds. Referring to the *mall* example, three different sets of locations for the interaction loci and the associated interactive objects are to be defined for three different malls; the activities that can be performed in the interaction loci of the different environments have a common definition in the abstract classes that are accessed by the experience generator for specializing the experience. Additional references about the authoring methodology may be found in [6, 20].

The result of the authoring process is a set of experiences that are used by the run-time system (described in Section 6) for generating 3D interactive worlds. The interaction spaces are made of two components:

- a *base world*, the underlying real world in an augmented reality environment and the 3D static part of the simulation and all the dynamic parts that do not participate to the interactive experience in a virtual environment;

- an *experience layer*, composed by the definition of the interaction loci that populate the environment and the interactive objects contained in them.

The experience layers are superimposed to the base worlds, and are available at different times; i.e., the interaction locus component, which defines the interactive experience, can be changed without affecting the base world. The different experience layers are available in a repository associated to each different 3D environment that can be accessed by the run-time system. Users approach a specific experience among the ones that are available for their profile as the result of an adaptation process performed by the interaction management system. Shifts between different experiences are performed by the interaction adaptive system according to the user profile and to the current context.

In Figure 1 the naming schema Xm/n is adopted for denoting classes of experience (X), interactive worlds (m) and user classes, or user profiles (n). In the interactive world 1 repository four different experiences, each generated from a different class, are available for users having profile 1. In world 2 three different experiences (C2/1, C2/2 and C2/3) are generated from the same class of experience for three different user profiles. For example, they could be referred to the same 3D environment, but emphasizing the needs of three different customer ages, such as teenagers, adults and seniors. An additional experience (D2/1) generated from a different class is available for profile 1 users who have therefore two different experience opportunities.

5. OBSERVING PATTERNS OF INTERACTION

In order to observe and adapt user interaction, the interaction process itself must be described formally. 2D interfaces have been extensively investigated; we assume as a starting point of our methodology the *PCL model* of human-computer interaction [3], targeted to solve usability issues in interactive systems design. According to this approach, interaction is a cyclic process where users interpret images on the screen (called *characteristic structures*) and recognize in them perceptive units with meanings and functional roles. The result of such interpretation consists in issuing commands that the machine interprets and executes, materializing the result as a modification of the screen image, i.e., of the characteristic structures on it. Therefore, users and computers communicate by interpreting the other's messages and materializing their own. Each step transforms the visible interface, which takes the correct state to respond to user's interaction.

The PCL model formalizes the interaction process and describes how to map this interactive process on a control finite-state machine (CFSM). A complete description would be beyond the scope and size of this work, therefore the reader is referred to [3] for details. In summary, the interactive process is described as a set of pairs $\langle s(t), a(t)\rangle$ where s is the state of the CFSM and a is the activity the user is performing at time t. A sequence of such pairs for consecutive times $t_k, \ldots, t_m$ is called *Pattern of Interaction* (*POI*) and describes the activities the user has done in that specific interaction period. Informally, *POI*s are "*sequences of activities which the user performs in some specific situation during the interactive execution of a task*" [2].

The patterns and the reasons of their repetition are interesting since they can reveal preferential uses of the system, and can give designers the opportunity to modify it, for example by introducing of new functionalities. Patterns of interaction can be observed and expressed in a form suitable for subsequent automatic analysis. Assuming the knowledge of the Control Finite State Machine of a class of equivalent *IL*s, during the interaction it is easy to observe the activity $a(t)$ performed by the user at time step t and relate it to the current state $s(t)$ of the CFSM, and then derive the state $s(t+1)$ which is reached as a consequence of activity $a(t)$.

Following the *POI* approach, Arondi et al. [2] have used a set of specialized agents [15, 12] for analyzing the user activity. The approach was then extended to 3D worlds [5]. The methodology discussed here uses two types of agent for monitoring the user activity and for adapting interaction on the basis of recurring patterns. Extending the definition of pattern of interaction to the 3D world is not trivial, due to the almost unlimited possibilities the user has to move around without really interacting with the world. While each input event can be traced, it is almost useless to consider user movement unless it produces an effect on the world. Therefore we consider only interactions captured by explicit sensors such as proximity sensors, triggers, and so on, and of course all the interaction with the objects and artifacts of the world.

The agents are named according to a metaphor inspired by the ancient Roman religion: a *numen*, a kind of *guardian angel*, is associated to the user, knows his/her profile and maintains knowledge about his/her previous interactions. A *genius loci* is associated to each interaction locus, knows its topology and the interaction opportunities in it, including those of the interactive objects associated to it. In terms of the PCL model, it knows the CFSM of the locus, and is therefore able to identify paths among its states.

The communication protocol between a user *numen* and a *genius loci* is activated when the user enters an interaction locus, and continues according to the following steps:

1. the *genius loci* gets from the *numen* the user profile, which is a static collection of properties and data about the user, and a user history, which is a set of data collected during navigation and interaction in the world;

2. the *numen* gives the *genius loci* the information requested, and the *genius* possibly modifies the interaction properties of that locus;

3. the user then interacts in a more effective way, while the *genius* keeps on in analyzing the user behavior, possibly discovering repeating patterns of interaction according to the information received from the *numen*. If such patterns are discovered, the *genius loci* modifies the interaction behavior;

4. on exiting the interaction locus, the *genius loci* returns to the *numen* the result of its analysis. The *numen* looks for repeated patterns of interaction compatible with the profile, updating the dynamic user history.

When entering a new interaction locus, the dialog is repeated. The modification of user interaction can take place in two ways: (1) the *numen* drives the user in the interaction locus corresponding to the selected experience layer among the ones provided for that world, and (2) the *genius*

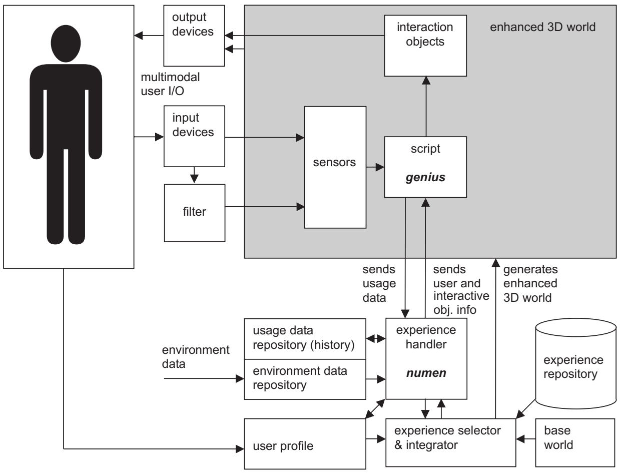


Figure 2: System architecture

loci changes the behavior of the CSFM of the locus. These two ways are detailed in the next two sections.

6. SYSTEM ARCHITECTURE

Figure 2 describes the implementation architecture for an adaptive interaction system designed according to the concepts described so far. The architecture refers to the mixed reality domain [16] and therefore can be used both for virtual worlds where the user *avatar* wanders through a modelled 3D scene, and for augmented reality situations where the user moves with his/her own body in a real scene. A specific component (*filter*) may be needed in the latter situation for managing the input of the user location, that may be affected by errors related to the specific device used for monitoring it (e.g., GPS, IR sensors, etc.).

Users interact with a 3D scene composed by the base world and the experience layer. The second part is the active part of the 3D scene, containing the definition of the interaction loci, a variety of sensors that capture the user input, the interactive objects, artifacts and dynamic information objects associated to the *IL*, and the *genius loci*.

As discussed in Section 4, there may be one or more alternative experience layers, therefore one of more sets of *IL*s superimposed to the same 3D base worlds. As already shown in Figure 1, the experience layers are kept in a separate database (*experience repository*) that can be accessed by the *experience selector and integrator* (Figure 2); this component adaptively builds the enhanced 3D world associating the experience layer to the 3D base world on the basis of the user profile and the available experience layers, according to the mapping defined in the authoring phase [20].

Data coming from the user can be separated into two different categories: the ones collected before interaction and stored in the user profile and the ones generated during interaction. Concerning the latter category, only a part of them (possibly filtered by the filter component) are caught by the sensors associated to the *ILs* and embedded in the 3D scene. The sensors pass such data (i.e., the patterns of interaction) to the *script component* that implements the *genius loci* entity. There is a separate script component for each *IL*, running a separate computation process.

The script component has three main functionalities: it logs data received by the sensors, compares them with recurring interaction patterns received by the *experience handler* for possibly adapting user interaction with the interactive objects, and eventually sends the usage data to the experience handler for further processing.

The last component, external to the 3D world, implements the *numen* entity. The experience handler receives the usage data from the script components and identifies the recurring patterns. The experience handler accesses also the user profile, that may influence adaptivity related to the different classes of objects and *IL*s. A third category of data, the *environment data*, can be meaningful for the interaction and therefore may be monitored by the system and accessed by the experience handler. The accumulated knowledge is passed to the script components that use it for adapting interaction.

In order to test the system functionalities we have built a demonstration prototype in VRML. Cortona SDK[1] and Microsoft Visual Studio were used for implementing the components external to the 3D world: the experience handler and the different accessory components for storing user, usage and environmental data. Communication and control of the VRML scene are also handled by components external to the 3D world.

7. CASE STUDY: A VIRTUAL FAIR

The case study concerns a 3D simulation representing an interactive experience in a virtual fair. Users can wander through the stands, ask information about the products, consult brochure and have live demonstrations of objects on exhibition. This experience can be formulated in the following terms: "*A shopping experience in a virtual fair must allow a user to explore the fair in an organized way according to his/her needs, expressed by setting a user profile or by recalling data collected in a previous analysis phase. The exploration has the goal of obtaining information related to the products available and easily finding the products of interest*".

The pavillion contains 18 stands belong to 4 different classes: *hardware products*, *software*, *entertainment* and *hi-fi*, visually identified by a different color for the floor and a specific sound track the user can hear upon entering a stand. The interactive objects are categorized in 6 classes: *hostess*, *showcase*, *brochure*, *product*, *TV screen* and *test bench*. Each stand of the fair is associated with an *interaction locus*. Also interactive objects whose behavior is not trivial have their own interaction locus.

The sensors associated to the *IL*s and the interactive objects embedded in the 3D scene are mapped to a CFSM, according to the PCL model, in order to log the user interaction and adapt the 3D environment to the user's needs. The mapping, made using the statechart standard, represents three different levels of detail: (i) the pavillion level, (ii) the stand level and (iii) the interactive object level.

The user interacting in the 3D world moves through the CFSM states. The arcs starting from his/her current state define the relevant interaction possibilities, and when followed specific task are accomplished. The monitoring activity allows the *genius loci* to store the interaction patterns and to compare them with the information received from

[1] www.parallelgraphics.com/products/cortona

the *numen*. Such information is used for possibly driving the user directly to a state representing the ultimate interaction action of the pattern, even if the state is not directly connected to the current state. In other words, the system automatically performs a set of actions on behalf of the user.

At system start-up, the 3D world is loaded with the experience layer associated to the user profile. The experience handler parses the world description, identifies the interaction loci, the associated interactive objects and creates a *genius loci* for each *IL*. The experience handler creates also the CFSM of the entire world and initializes it.

A threshold mechanism is used to monitor the user activity. The system initializes the number of interactions to zero for each type of interaction locus, i.e., for each class of interactive objects and stands, and increments it whenever the user interacts with the objects located in the various stands. The thresholds which trigger the proactive behavior are defined in the user profiles. When user interaction with a class of objects reaches the threshold value, the system modifies the behavior of all the objects of the same class. Recurring interaction with the same type of stand marks that stand category as preferred for personalized guided tours. Updating the user profile with the cumulative interaction count enables the user to take advantage of previous experience in future visits of the world.

7.1 The interface

The implementation belongs to the category of the so-called desktop virtual reality applications, that renounces to VR immersion and are characterized by the use of standard monitors for visual output and keyboard and/or mouse for input activities. In particular the implementation adopts a hybrid interface that organizes the visual layout in a set of functional areas.

While the artifacts that are specific to certain scene location are contextualized in the 3D scene, those ones that need to be always present are visualized outside the 3D areas as bi-dimensional artifacts. All the textual and iconic messages for the user are displayed over the top of neutral backgrounds in areas external to the 3D scene. This contributes to lower the system load for the scene rendering, reserving more resources for augmenting the visualization frame rate, and to augment the readability of the messages.

Figure 3 shows the prototype interface, populated by the 3D geometry and related data of the virtual fair. The 3D

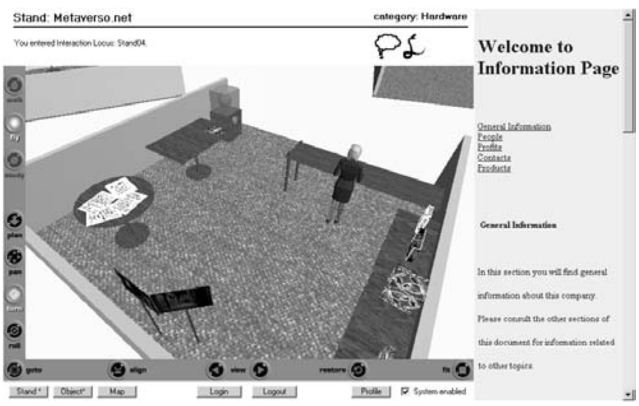
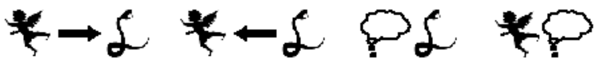


Figure 3: The prototype interface



(a) (b) (c) (d)

Figure 4: The icons showing agents activity

visualization area fills most of the screen. Panels and menu bars show information related to the current state of interaction, hypertext related to the interactive objects and activate functions for managing the user profile and other tasks.

Two messages above the 3D visualization identify the *IL* the user is currently navigating and its class. Additional messages dynamically describe the user activity, signaling the last action monitored by the system.

Icons (an angel for the *numen* and a snake for the *genius loci*) show the agents activity for that particular phase of interaction (Figure 4):

(a) the user enters the *IL*; the *numen* sends information (if available) to the *genius loci* associated to the *IL* for adapting interaction;

(b) the user exits the *IL*; the *genius loci* communicates to the *numen* the patterns of interaction;

(c) the user is within the stand; the *genius loci* tracks the user and may adapt interaction in case of recurring patterns of interaction;

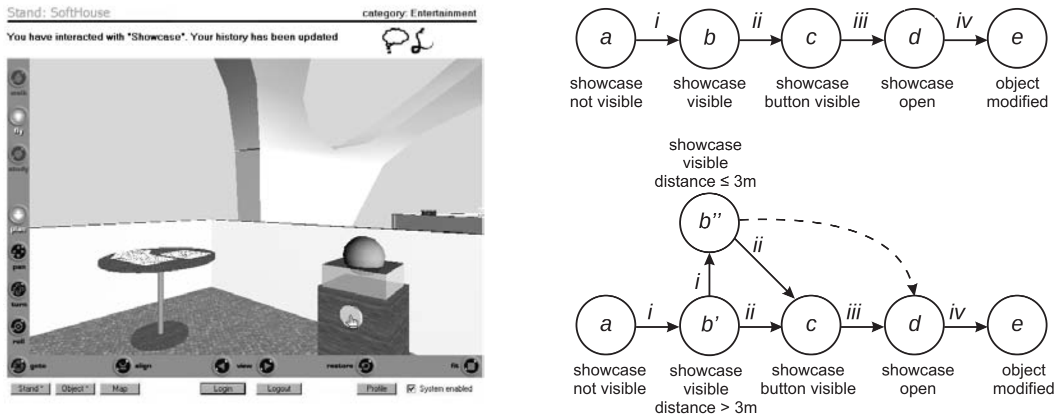
(d) the user is out of the stand; the *numen* elaborates the information received from the *genius loci*, identifying recurring patterns of interaction.

According to well-known usability heuristics [17], informing the user about the state of the system is an important means for improving its usability. In interactive systems, informing the users about the state of interaction helps them to feel in control of it, and contributes to the acceptance of adaptive interaction. That is the reason why messages shown by the system in the interface upper area describe also the actions performed on behalf of the users, such as the automatic personalization performed by the adaptive engine.

For testing the system and experimenting several configurations, the interface is provided with a button bar activating functions for managing and tracking the adaptiveness. In a real application they should be protected from user activation and managed by a system administrator.

The functions allow the user to log into the system, to set up his/her profile, and to enable and disable the adapting system. As we shall discuss in Section 7.2, recovery from wrong system assumptions about the user needs is an important issue; the user must be at any time in control of the interaction, and if fears to loose control he/she should be able to switch off the personalization engine for some time, in order to gradually take confidence with the system.

The prototype doesn't implement sophisticated navigation techniques. Users move through the world using the standard controls of the VR interface. A limited form of adaptable navigation is supported: among the available navigation paradigms a *guided tour* through favorite stands allows the user to visit in sequence only the stands of the most frequently accessed category in previous interaction phases.



(a) (b)

Figure 5: User interaction with a showcase: (a) interface; (b) CFSM

7.2 Adaptive behavior examples

Figure 5a illustrates user interaction with a showcase. The user interface shows the showcase with other interactive objects in the stand. Interaction with the showcase is characterized by the following sequence of actions: (i) move near to the showcase; (ii) find the button; (iii) click on the button for opening the showcase; (iv) interact with the object that was concealed by the glass. Showcases can be found in several stands of the fair, so tuning their response to user interest can be worth.

Figure 5b shows the CFSM mapping user interaction with the showcase. The nodes represent the different states of interaction, while the arcs represent the user actions. Only the actions relevant for opening the showcase are drawn.

Showcases may have different orientation in the virtual fair, therefore the visitor may need to turn around the showcase to find the opening button, and to go forth and back for taking it on the screen.

A set of proximity sensors surrounding the interactive objects of a certain class are created by the *genius loci* whenever the threshold associated to that class is exceeded. The sensors react to the user presence within their spatial limit, activating the interaction sequence guessed by the personalization system. Adaptive interaction automatically opens the showcase when the user approaches it without additional interaction, but only if the user has shown interest for such objects, effectively diminishing the user cognitive load.

From the implementation side, the adaptive system partially modifies the CFSM, splitting state (b) into two different states: state (b'), where the showcase is visible but far from the user location, and state (b") where the showcase is visible and close to the user. The distance at which transition between b' and b" occurs is defined by the bounding box of the proximity sensor surrounding the showcase. Going to state (b"), i.e., triggering the proximity sensor, makes the system to automatically execute steps (ii) and (iii), as shown by the dashed arrow of Figure 5b.

In other cases the adaptive system does not modify the CFSM of the interaction locus. Instead, when recognizing a recurring behavior, a sequence of steps on the CFSM are executed on behalf of the user, leading him/her directly to the final state of the interaction sequence.

Other approaches for supporting users during interaction are based on suggestions [10] that can be accepted or not by the user.

While adaptive interaction shares with this category of approaches the aim to be useful and not to cause problems when the guess about the user intentions is wrong, it differs from them because it promotes automatic execution of certain activities on behalf of the user.

Recovery from unwanted situations is an important heuristic [17] that anyway should be considered in the design of interactive systems. That is the reason why in the prototype we avoided strong adaptations that would have prevented the user from easy recover, such as teleporting to a distant location. All the personalization proposed allow the user to recover using standard and fast interaction techniques.

Besides de-activating the adaptation system, *undo* operations should also be provided together with messages describing the performed adaptation. In terms of implementation such a widget is not difficult to add, since it has to bring the CFSM back to the previous state, recover the previous state of the user interface, and temporarily disabling the adaptive system in order to avoid interaction loops.

8. SUMMARY AND FUTURE WORK

We have discussed a methodology for adapting a navigation oriented interactive environment to the user needs based on three base concepts: the structuring of the interaction space in a set of *interaction loci* which help the user to identify the interaction opportunities; a specialization of generic *classes of interactive experiences* according to the user profile; a dynamic modification of the interaction behavior according to the analysis of *user patterns*. An *agent-based* architecture for augmented/virtual reality environment has been discussed. The proposed system is also a valuable tool for designers. Information about recurring patterns of interaction shared by several user could help the designer to effectively improve the design.

Informal tests of the demonstration application with a few users evidenced a good degree of satisfaction both about system behavior and about the interactive experience, but only a thorough test, planned as a next step, could evaluate the usability according to standard procedures.

More complex mechanisms beyond the simple threshold based level implemented in the prototype can be designed. Our goal was not to define sophisticated technical solutions, but rather to provide a methodological framework in which several technical solutions can be experienced.

The communication system of agents is based on a shared knowledge about the interaction loci: both the *numena* and the *genii loci* know about fairs, stands, hostesses and displays. Switching to a different 3D world, e.g. a museum, generally requires references to a different knowledge base. The difference, hence the need to re-define the knowledge base for different interactive worlds, is bound to the classes of experience approach, where different classes require to set up different knowledge schemas. An important issue to investigate is the possibility to define a base ontology for describing common interaction behaviors that can be shared among different classes of worlds.

9. ACKNOWLEDGEMENT

The authors gratefully acknowledge the contribution of Michele Nodari to the development of the prototype system.

10. REFERENCES

[1] G. D. Abowd, C. G. Atkeson, J. Hong, S. Long, R. Kooper, and M. Pinkerton. Cyberguide: A mobile context-aware tour guide. *ACM Wireless Networks*, 3:421–433, 1997.

[2] S. Arondi, P. Baroni., D. Fogli, and P. Mussio. Supporting co-evolution of users and systems by the recognition of interaction patterns. In *Proc. of the International Conference on Advanced Visual Interfaces (AVI 2002)*, pages 177–189, Trento, Italy, 2002.

[3] P. Bottoni, M. F. Costabile, S. Levialdi, and P. Mussio. Defining visual languages for interactive computing. *IEEE Transactions on Systems, Man and Cybernetics*, 27(6):773–783, 1997.

[4] P. Brusilovsky. Adaptive hypermedia. *User Modeling and User Adapted Interaction*, 11(1–2):87–110, 2001.

[5] A. Celentano, D. Fogli, P. Mussio, and F. Pittarello. Mobile context-aware interaction: an agent based approach. In *Workshop on Artificial Intelligence in Mobile Systems (AIMS), ECAI Conference*, pages 29–36, Lyon, France, 2002.

[6] A. Celentano and F. Pittarello. Classes of experiences: a high level approach to support content experts for the authoring of 3d environments. In *Structured Design of Virtual Environments and 3D-Components.* Shaker Verlag, 2001.

[7] L. Chittaro and R. L. Ieronutti. Guiding visitors of web3d worlds through automatically generated tours. In *Proceedings of the 8th International Conference on 3D Web Technology*, Sain Malo, France, 2003.

[8] L. Chittaro and R. Ranon. Dynamic generation of personalized vrml content: a general approach and its application to 3d e-commerce. In *Proceedings of the 7th International Conference on 3D Web Technology*, pages 145–154, Tempe, Arizona, 2002.

[9] A. V. Dam. Post-WIMP user interfaces. *Comm. of the ACM*, 40(2):63–67, 1997.

[10] A. Dix, R. Beale, and A. Wood. Architectures to make simple visualisations using simple systems. In *Proceedings of the working conference on Advanced visual interfaces (UIST'98)*, pages 51–60, Palermo, Italy, 2000.

[11] J. Fink, A. Kobsa, and A. Nill. Adaptable and adaptive information provision for all users, including disabled and elderly people. *The New Review of Hypermedia e Multimedia*, 4:163–188, 1998.

[12] S. Franklin and A. Graesser. Is it an agent or just a program? a taxonomy for autonomous agents. In J. P. Miller, M. J. Wooldridge, and N. R. Jennings, editors, *Intelligent Agents III – Proc. of the Third Intern. Workshop on Agent Theories, Architectures, and Languages*, volume 1193 of *LNAI*, pages 21–35. Springer Verlag, 1996.

[13] T. Guinan, G. M. P. O'Hare, and N. Doikov. Enter: The personalisation and contextualisation of 3-dimensional worlds. In *Proceedings of Eight Euromicro Workshop on Parallel and Distributed Processing (EURO-PDP 2000)*, pages 142–149, Rhodes, Greece, 2000.

[14] E. Haik, T. Barker, J. Sapsford, and S. Trainis. Investigation into effective navigation in desktop virtual interfaces. In *Proceedings of the 7th International Conference on 3D Web Technology*, pages 59–66, Tempe, Arizona, 2002.

[15] N. R. Jennings and M. Wooldridge. Intelligent agents: theory and practice. *The Knowledge Engineering Review*, 10(2), 1995.

[16] P. Milgram and F. Kishino. A taxonomy of mixed reality visual displays. *IEICE Trans. Inf. Syst.*, E77-D(11), 1994.

[17] J. Nielsen. *Usabilty Engineering.* Academic Press, Bosto, Massachussets, 1993.

[18] M. Perkowitz and O. Etzioni. Adaptive web sites. *Communications of the ACM*, 43(8):152–158, 2000.

[19] F. Pittarello. Multi sensory 3d tours for cultural heritage: the palazzo grassi experience. In *Proceedings of ICHIM2001, Cultural heritage and technologies in the 3rd millennium*, Milano, Italy, 2001.

[20] F. Pittarello. Accessing information through multimodal 3d environments: Towards universal access. *Universal Access In the Information Society Journal*, 2(2):1–16, 2003.

[21] F. Pittarello and A. Celentano. Interaction locus – a multimodal approach for the structuring of virtual spaces. In *Proceedings of HCItaly 2001 Symposium*, Florence, Italy, 2001.

[22] M. Schneider-Hufschmidt, T. Kühme, and U. M. U. (eds.). *Adaptive user interfaces: Principles and practice.* Elsevier Science, Amsterdam, 1993.

[23] C. Stephanidis. Adaptive techniques for universal access. *User Modeling and User Adapted Interaction*, 11(1–2):159–179, 2001.

[24] N. Vinson. Design guidelines for landmarks to support navigation in virtual environments. In *Proceedings of CHI '99*, pages 278–285. ACM Press, 1999.

[25] K. Walczak and W. Cellary. Building database applications of virtual reality with x-vrml. In *Proceedings of the 7th Int. Conference on 3D Web Technology*, pages 111–120, Tempe, Arizona, 2002.