

# Enhancing Virtual Reality Learning Environments with Adaptivity: Lessons Learned

Olga De Troyer, Frederic Kleinermann, Ahmed Ewais

Vrije Universiteit Brussel, WISE Research Group, Pleinlaan 2,  
1050 Brussel, Belgium

{Olga.DeTroyer, Frederic.Kleinermann, [Ahmed.Ewais@vub.ac.be](mailto:Ahmed.Ewais@vub.ac.be)}

**Abstract.** Virtual Reality (VR) is gaining in popularity and its added value for learning is being recognized. However, its richness in representation and manipulation possibilities may also become one of its weaknesses, as some learners may be overwhelmed and be easily lost in a virtual environment. Others may spend all their time on exploring features not relevant for their learning tasks. Therefore, being able to dynamically adapt the virtual environment to the personal preferences, prior knowledge, skills and competences, learning goals and the personal or social context in which the learning takes place becomes important. In this paper, we discuss possible adaptations and adaptation strategies for virtual learning environments. We also report on a prototype implementation of an adaptive Web-based virtual learning environment and the lessons learned from this.

**Keywords:** E-Learning, Virtual Reality, Adaptivity, Virtual World Learning Environment

## 1 Introduction

Some material is easier to learn when it is visualized and when the learner can interact with it. Virtual Reality (VR) provides ways to use 3D visualizations with which the user can interact. For some learning situations and topics, VR may be of great value because the physical counterpart may not be available, too dangerous or too expensive. The most famous example is the flight simulator that pilots safely teach how to fly a plane in various circumstances. Another example where VR has been used successfully is the domain of medicines, e.g., to simulate operations or to study the human body. Virtual worlds (such as Second Life [1] and Active Worlds [2]) are complete 3D computer environments containing 3D (and possibly also 2D) objects in which the user can navigate and interact with the objects. Usually, the objects may have different behaviors and users are presented by means of avatars. These virtual worlds are also called Virtual Environments (VE). In collaborative environments, such as Second Life, users can also meet each other and socialize. For certain subjects and for certain types of learners using such a VE may be much more appealing and motivating than the use of classical learning material, e.g., to simulate the effect of physical laws (e.g. [4]); to simulate social environments and allow people to practice social skills; or to learn about history (e.g. [5], [6]).

However, the richness of such a Virtual Reality Learning Environment (VLE) can also become its weakness. The learner may be overwhelmed or get lost in the VLE [7], not knowing what to do first or next, or may be distracted too much and not be able to focus on the actual learning task. For people not familiar with VEs (novice users), the time required to get acquainted with such a VLE (i.e. learnability) may be long and therefore their short-term satisfaction may be low. On the other hand, youngsters used to play video games, may spend their time in activities not very much related to the learning activities, especially if they have low motivation for learning. This then results in a low effectiveness. These concerns are confirmed in [8]. The authors reported “novice student-players made quite a lot of navigational effort. This means that on average they had wasted quite a lot of their time trying to find their way in the virtual reality worlds and thus they had been left less time for reading the theory and answering questions that would help them extend and consolidate their knowledge”, “virtual reality distractions were observed in many students’ protocols but not to a great extent. The distractions occurred when users behaved as if they had forgotten what their ultimate goal was... Instead, they repeated actions without any particular meaning” and “one important finding is that the first two kinds of usability problem (user interface acquaintance and navigational effort) affect mostly the less experienced whereas the third kind of usability problem (virtual reality distractions) affects mostly the more experienced users”. The authors concluded: “these findings show that all categories of user may benefit less than they could from the educational content of a VR-educational game due to usability problems. Thus the design of VR-game interfaces has to attract a lot of attention for the elimination or improvement of these three kinds of problem”. Although, these results were obtained in the context of a VR-educational game (which is a special type of VLE) Dede et al. [6] made similar statements. They noted that students exhibit noticeable individual differences in their interaction styles, and abilities to interact with the 3-D environment. Furthermore, they observed that usability and learning are two goals that may conflict. Optimizing for usability may impede learning if it requires changes to the interface that rely on interactions or representations that are inappropriate for the learning task.

One way to solve these problems is by providing the VLE in an adaptive way, e.g., adapted to the individual learner and to the progress that he makes during the learning. Augmenting a VLE with adaptive capabilities has many advantages [9]. It could solve the difference between novice and more experienced users, but also adapt the VLE to other individual differences. For instance, it may be more effective to guide a learner through the VE according to his/her background and learning goals, or only show him/her the objects that are relevant for his/her current knowledge level, or adapt the environment to his/her learning style. It is also well known that cultural aspects influence the learning process [10], so adapting the VLE to the culture of the learner can also be important.

In this paper, we explain how a VLE can be dynamically adapted to an individual learner to better support the actual learning process and increase the usability. The actual creation of the VR learning material is not in the scope of the research. With the success of application such as Second Life, Active Worlds, more and more VR material becomes available for free on the Web (e.g., in Google’s 3D Warehouse [11]) and easy to use tools are available to create such material (e.g., Google SketchUp [12], 3D Studio Max [13]).

The rest of the paper is structured as follows. In section 2, we consider related work. Section 3 deals with possible adaptations in a VLE. In section 4, we give some adaptation strategies that can be used in an adaptive VLE to enhance usability. Section 5 presents the different approaches that can be used for an adaptive virtual reality learning system and the different components needed. In section 6, we report on a prototype adaptive virtual reality learning system implemented and on the lessons learned from this. Finally, section 7 presents conclusions and future work.

## 2 Related Work

In general, little research has been done in the context of usability of VLEs. Some work deals with the development of methods to investigate the usability of VEs (e.g., [14], [15]), which is not directly relevant for this paper. Other work investigated the usability of some individual VE, such as [6], [4], [7], [8]. No work has been found on investigating the usability of adaptive VR environments, neither on guidelines for using adaptive techniques in the context of VR learning environments. Some work deals with adaptation in 3D environments. We will review this work without being limited to the domain of E-learning.

Brusilovsky et al. [16] have integrated some adaptive hypermedia methods into virtual environments by developing an approach that supports different navigation techniques in the context of 3D E-Commerce. It matches the user needs for shopping by supporting different navigation techniques. This work is interesting because it has extended some of the adaptive hypermedia methods (such as direct guidance, hiding, sorting) to 3D environments.

Chittaro and Ranon have done quite some work in the context of adaptive VR. In 2000 [17], they have described how to introduce adaptation inside e-commerce. Their approach is called ADVIRT. A set of personalization rules exploits a model of the customer to adapt features of the VR store, such as the display of products through the concept of shelf, display spots and banners. They have also customized and personalized the navigation and different layouts of the store. In 2002 [18], they have introduced a software architecture for adaptive 3D web sites called Awe3D (Adaptive Web 3D) which can generate and deliver adaptive Virtual Reality Modeling language (VRML). This work targeted E-commerce and not E-learning. However it shows how 3D content can be personalized according to the user. In 2007 [19], the same authors has explained that adaptation can happen for navigation and interaction in order to help the users in finding and using information more efficiently. For navigation and interaction, they proposed direct guidance, hiding, sorting and annotation based on the work of Brusilovsky et al. [16], The adaptations they propose, are derived by making an analogy with adaptive web-based hypermedia. We take a different approach. The types of adaptation what we propose are based on the different components that make up a VR environment. This results in a more elaborated set of adaptation types. Furthermore, Chittaro and Ranon suggested an alternative approach for sorting and annotation by using virtual characters that act as navigation guides to show users the path to an object of interest, and to provide annotations in the form of additional information on navigation and interaction possibilities. Finally, and based on their

previous work [20], Chittaro and Ranon have extended the E-learning platform EVE [21]. They introduced Adaptive EVE that is tailored to the knowledge level of a student and to their preferred style of learning. To achieve adaptivity in the context of EVE, they have used the AHA! engine which was originally developed for adaptive hypermedia applications [22].

Santos and Osorio [23] have introduced another approach for adaptation in VR. Their approach is called AdapTIVE (Adaptive Three-dimensional Intelligent and Virtual Environment) and is based on agents, called Interactive and Virtual Agents that assist the users and help them to interact with the environment. They have applied their approach to E-commerce and Distance Learning systems.

Celentano and Pittarello [24] have developed an approach for adaptive navigation and interaction where a user's behavior is monitored in order to exploit the acquired knowledge for anticipating user's needs in forthcoming interactions. The approach uses "sensors" that tells when an object has been interacted with. These software sensors collect usage data and compare them with previous patterns of interaction. These patterns represent sequences of activities that users perform in some specific situation. Whenever the system detects that the user is entering a recurrent pattern of interaction, it may perform some activities of that pattern on behalf of the user.

Daschelt et al. [25] have developed an approach that provides adaptation to the user's device. Their approach suggests different alternatives with respect to the screen space usage for the same 3D interface element and information presented. Furthermore, 3D content is also considered in media adaptation. For instance, they describe a showcase where the seat capacity of a conference room can be adapted. This work is more on adapting the content for large audience.

In the ELEKTRA project [26], an EU-project aiming to bridge the gap between cognitive theory, pedagogy and gaming practices, a framework for adaptive interventions in educational games was introduced. They introduce a difference between macro- and microadaptivity. Macroadaptivity refers to traditional techniques of adaptation such as adaptive presentation and adaptive navigation. Microadaptivity is adaptation within learning tasks. Microadaptivity affects only the presentation of a learning object or a learning situation. It is achieved without compromising the learner's gaming experience. For this they use an adaptation system that provides recommendations to the game engine, but it is the ultimate decision of the game engine to whether or not to enact a recommendation. This approach was also followed in the 80Days project, the successor of ELEKTRA. Pedagogically, this project is grounded in the framework of self-regulated personalized learning which propagates the importance of self-regulation [27].

### **3 Adaptation in Virtual Reality Environments**

There are many definitions of Virtual Environments (VE) or Virtual Reality (VR) [28]. In the context of this research, we focus on desktop VR, which can be defined as a three-dimensional computer representation of a space, displayed on screen, in which users can move their viewpoints freely in real time and perform several actions.

Before we discuss how a VE (and more in particular a Virtual Learning Environment VLE) can be adapted, we first discuss the different components of a VE:

### **1. The scene and the objects**

The scene corresponds to the 3D space in which the objects are located. It contains lights, viewpoints and cameras. Furthermore, it has also some properties that apply to all the objects being located inside the 3D space. For instance, gravity can be a property that applies to all its objects. The objects are usually 3D, but there can also be 2D objects in the 3D space. They have a visual representation with color and material properties, a size, a position in the world, and an orientation. Special objects, so-called avatars, are used to represent the user(s) in the 3D space. The user's avatar can be represented explicitly (by an object) or implicitly in which case the viewpoint of the camera is used to show the user's position.

### **2. Behaviors**

The objects may have behaviors. Behaviors may reflect real life behaviors. For instance, objects may be able to move, rotate, change size, transform, and so on. An avatar is an object with behavior. Usually, avatars can walk, run, and sometimes even fly through the 3D space. Navigation through the 3D space is achieved by the behavior of the user's avatar. In general, the 3D space also contains objects without behavior.

### **3. User Interaction**

In a VE, the user is able to interact with the objects. For example, a user may pick up an object and drag it to some other place in the space (if the object is moveable). User interaction can also trigger behavior, e.g., clicking on an object may start a behavior. User interaction may be achieved by means of a regular mouse and keyboard or through special hardware such as a 3D mouse or data gloves [30].

### **4. Communication**

Nowadays, more and more VEs are also collaborative environments in which remote users can interact with each other, e.g., talk or chat to each other or perform activities together. For some learning situations, e.g., practicing social skills, this can be an important requirement.

### **5. Sound**

A VE usually also involves sound. Sound can be important in simulations, to enhance the feeling of reality or simply to simulate some sound. Sound/speech can also be used as an instruction and feedback mechanism during the learning process. A VE can be adapted in many different ways. In principle, adaptation can happen for each of the components of a VE. An adaptation can be limited to a single component of the VE, but it can also involve many different components. We will first describe adaptations that apply on single component, i.e. on objects, behaviors, interaction, and for avatars. We call these *adaptation types*. For the moment, we didn't consider any adaptation types yet for the scene or for sound. Communication is also not considered because we focus on single-user VLEs. Next, in section 4, we will consider more high-level adaptations that involve more than one component. These we called *adaptation strategies*. In this paper, we particular focus on adaptation strategies for VLEs. To express the adaptation types and adaptation strategies, formal instructions have been developed but they are omitted here; we only describe them informally.

### 3.1 Adaptation Types for Objects

Objects populating the VLE have a visual appearance in terms of a geometry (shape) and material properties (colors and textures). To enhance the usability of a VLE for a learner, it may be advisable to change the visual appearance of an object during the lifetime of the VLE. We illustrate this with some examples. To visually indicate that an object has not yet been studied we may want to highlight it, make it smaller or even hide it. When a student is learning about an object being represented in the VLE, the visual appearance of the object could change according to the aspects being studied, for instance when studying a planet, it would be interesting to adapt the representation of the planet to the aspect considered, like its internal composition, or its atmosphere. It may also be useful that the visual representation of an object becomes more detailed while more and more knowledge is acquired. For instance, the planet Saturn can be shown first with its rings. Once the user has read enough on Saturn and its satellites, the satellites inside the rings of Saturn can be shown as well.

Therefore, a first set of adaptation types for objects is concerned with the adaptation of the visualization of an object, i.e. how to display it and how to hide it:

- *semiDisplay* is used to display the object in a semi-manner, by having a semi-transparent bounding box around it. See figure 1(a) for an example: the Sun has been semi-hidden.
- *changeSize* is used to change the size of an object.
- *changeMaterialProperties* is used to change the material properties (color or texture) of an object. Figure 1(b) shows the sun with a different texture than in figure 1(a).
- *changeVRRepresentation* is used to change the visual representation of an object completely; its current visualization will be replaced by a different one.
- *semiHide* allows to make the visual appearance of objects semi-hidden. This is similar as the *semiDisplay* but the purpose is slightly different, i.e. hiding instead of displaying.
- *hide* allows hiding an object visually.
- *display* allows displaying an object that has been hidden before.

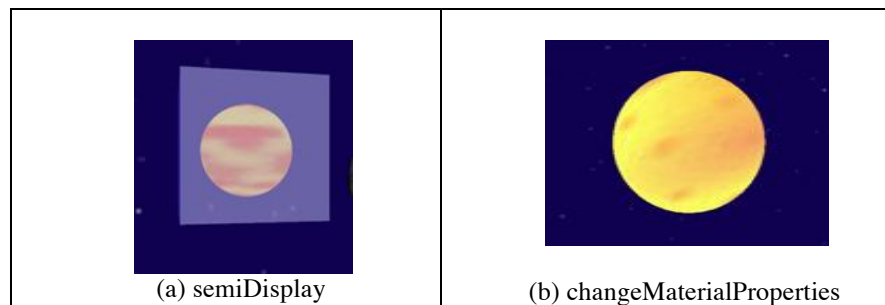
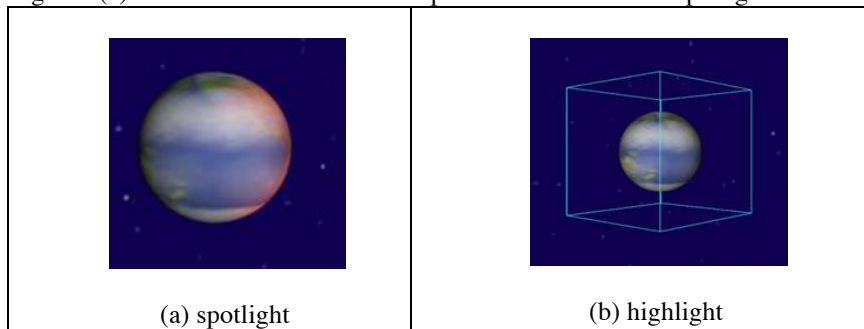


Figure 1

Furthermore, like in classical textual learning material, it may be useful to mark objects. A reason for marking an object is for instance to draw the attention of the

learner that the material associated with the object has (or has not yet) been studied or to indicate the importance of the object. In a VLE, marking an object can be done in different ways. We distinguish two different adaptation types for marking because they are essentially different:

- *spotlight* allows to mark an object by putting a spotlight on it; in this way the object becomes more visible and can be used to draw the attention to this object. Figure 2(a) shows an illustration of the planet earth with a red spotlight.



**Figure 2**

- *highlight* allows to mark an object by drawing a box around the object where only the edges of the box are displayed. Figure 2(b) shows the planet earth being highlighted. The size of the box and the color of the edges can be specified.

Note that marking is considered different from changing the material properties (like the color) of an object, as by marking we don't change any property of the object. Also note that adding a title or a name to a VR object is not considered as marking, but as annotation. In general, annotations can be used to attach explanations, comments or names to objects. Annotations should only be shown when appropriate for the learning process as otherwise it might clutter the VLE too much. In addition, it may be necessary to adapt the annotations to the profile of the learner. Therefore, adaptation types are defined for adding and hiding annotations:

- *displayAnnotation* allows displaying an annotation with an object. An example is shown in figure 3.
- *hideAnnotation* allows hiding an annotation associated with an object.



**Figure 3: displayAnnotation**

### 3.2 Adaptation Types for Behaviors

Behaviors are used to make the VLE dynamic, i.e. to create environments where objects are active by performing some behaviors. For instance, in a VLE for the solar system we can have planets that are rotating and comets that move through the universe. However, to guide the learning process it may be useful to disable and enable behaviors when appropriate, e.g., a solar system where all planets are rotating at the same time may be confusing for a beginner. It may also be useful to adapt the parameters of a behavior, for instance to show the behavior of the sun with a different value for its temperature. Possible adaptation types for behaviors are:

- *enableBehavior* allows enabling a behavior associated with an object.
- *disableBehavior* allows disabling a behavior associated with an object.
- *changeBehavior* allows changing a behavior by modifying the values of its parameters.

### 3.3 Adaptation Types for User Interactions

In a VE, there are different ways to interact with an object, e.g., by clicking on it, by touching it, by passing closed. Furthermore, interaction (e.g., clicking) can trigger the start (or end) of a behavior. In the context of a VLE, it may be useful to control the user interaction. For example, to not overload the learner, we may want to prohibit interaction with an object as long as the learner has not obtained a certain level of knowledge. The enabling of an interaction possibility could also be used as a kind of reward after some successful study, and disabling interaction after some time could avoid spending too much time with some appealing feature.

Adapting interaction comes down to *enabling* or *disabling* some type(s) of interaction provided for an object. Possible interaction types considered for objects are: touching, clicking, and passing by.

### 3.4 Adaptation Types for Avatars

Avatars play a special role in a VLE. The avatar of the user is used to provide the view of the user inside the VLE. The avatar can have the look of a person, but other types of representation are also possible. As already explained, sometimes the avatar does not have a representation. Furthermore, an avatar has behaviors that allow the avatar to progress inside the VLE (i.e. navigate), for instance, jumping and walking. *Jumping* means going from one point to the next by making jumps. *Walking* means going from one point to the next smoothly and by following a certain path. Note that other behaviors are also possible like *flying*. An adaptation type *adaptAvatar* is defined to change the representation and/or the behavior for an avatar.



## 4 Adaptation Strategies for VR Learning Environments

In the previous section, we have discussed possible adaptations for individual components (object, behavior, interaction, ...) of a VE. In this section, we deal with adaptations that go beyond the adaptation of a single component. These adaptations can have an impact on several components of the VE or on a part of the VE. We call them *adaptation strategies*, as they can be used as pre-defined strategies to adapt a VE. We focus in particular on adaptation strategies for learning purposes.

A first group of adaptation strategies defined are those that will have an impact on how the learner can navigate through the VLE:

- *restrictedNavigation* allows restricting the navigation of the learner to some of the objects in the scene. In other words, the learner will only be able to navigate from one object to the next object in a given list of objects. Furthermore, the navigation can be restricted to a particular navigation behavior, e.g., jumping or waking. To specify the objects allowed to navigate to, different selection criteria can be used. For instance, it could be useful to select objects based on the learner's knowledge, or based on pre-requisite relations between objects. In general, this adaptation strategy can be used to force a learner to visit only a pre-defined number of objects.
- *navigationWithRestrictedBehavior* allows restricting the possible behaviors of objects while navigating. The restricted behavior can apply on all objects (in the VLE) or on a specified list of objects. This strategy is for instance useful to allow a learner to first explore a VLE (or a part of it) without being annoyed by objects showing all kinds of behavior; afterwards when he is more familiar with the VLE behavior can be enabled (using another adaptation strategy).
- *navigationWithRestrictedInteraction* allows restricting the possible interactions with some objects while navigating. Similar as for *navigationWithRestrictedBehavior*, this adaptation strategy can be used to allow a learner to first explore the VLE (or a part of it) without being able to fully interact with the objects in the scene.
- *TourGuide* provides a tour guide to the learner. A tour guide takes the learner through a tour in a VLE. Like in real life, a tour guide can provide an easy and efficient way to learn quickly some essential facts about objects in a large and unknown VLE.
- The following adaptation strategies allow specifying that the learner can navigate freely in the VLE, with or without suggestions (*freeWithSuggestions* and *completelyFree*). Suggesting is done by using marking (i.e. spotlight or highlight). This strategy can be used to give the learner a lot of freedom but still provide some guidance.

A second group of adaptation strategies allow adapting a group of objects:

- *filterObjects* allows to filter the objects that should be available (visible) in the VLE. This strategy can be used to avoid that the learner doesn't know on which objects to focus first. It also allows gradually building the VLE; the more knowledge the learner obtains the more objects become visible.
- *markObjects* allows to mark (i.e. by highlight or spotlight) a number of objects in the VLE. This adaptation strategy can be used as an alternative to the *filterObjects*. In some situations it may not be possible (or not desirable) to hide objects (e.g., if

we want to show the connections and dependencies of a complex system). Also, some learners may find it annoying that not all objects are visible or may perceive the VLE not attractive anymore. The strategy can also be used to mark the objects already studied.

A next group of adaptation strategies are strategies to specify some conditions for displaying objects:

- *displayAtMost* allows to specify when some objects should not be displayed anymore. The condition can be some pedagogical criteria like the knowledge level the learner currently has for the object or a limit on the number of times that the object should be displayed. This strategy can for instance be useful if the purpose is to perform some tests without having the subject(s) of the study visible.
- *displayAfter* allows to specify the condition(s) that need to be satisfied for objects to be displayed. This adaptation strategy can for instance be used to keep the VLE appealing by dynamically changing the objects in the scene. This can avoid that the learner gets bored.

The next group of adaptation strategies consists of strategies for adapting the behavior of some objects conditionally:

- *behaviorAtMost* allows to specify when a behavior should be disabled. This adaptation strategy can for instance be used to avoid that the learner keeps “playing” with an object or a number of objects having the same behavior.
- *behaviorAfter* allows to indicate when a certain behavior of an object (or some objects) should be executed. This adaptation strategy can for instance be used to state that the behavior for some specific objects should only start when the knowledge level of the learner for these objects is above a certain threshold.
- *behaviorSpeed* allows to specify the speed of a behavior. This is in particular useful when the behavior simulates a real world behavior. Being able to slow down the behavior will allow the learner to better observe what is happening, especially for behaviors that are happening very fast in real time (e.g., an explosion, or the trajectory followed by a bullet).

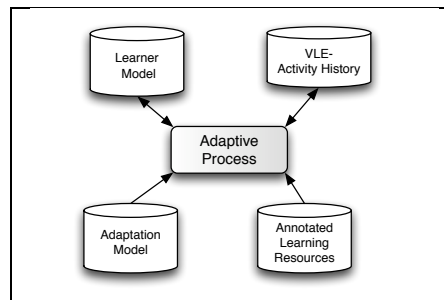
The last group of adaptation strategies contains strategies for limiting the interaction with some objects by means of a condition: *interactionAtMost* and *interactionAfter*.

The adaptation strategies presented here are only a subset of possible adaptation strategies. It is not our aim (and it would also not be possible) to define all possible adaptation strategies, but to provide a set of adaptation strategies that is useful in adaptive VLEs. It is of course possible to define a new adaptation strategy if there is a need for.

## 5 Driving the Adaptation in a VR Learning Environment

In the previous two sections we have provide the ingredients for making VLEs adaptive. In this section, we discuss how to actually drive the adaptation process and how this differs from the adaptation process of a regular learning environment. Similar as for an adaptive learning system, there are three different approaches: an author-driven approach, a teacher-driven approach and a model-driven approach.

In the *author-driven approach*, the author of the course is given full control over the adaptation process. During the design of the course, the author needs to specify the adaptations explicitly, e.g., through rules. This gives the control to the author but also requires that the author keeps track, at design time, of all possible adaptation scenarios, which may be hard. The alternative is to have some kind of automatic adaptation. This is what we call a *model-driven approach*, as the adaptation process is then driven at run-time by means of adaptation models and/or intelligent algorithms. The models and algorithms can be engineered in advance or be based on advanced AI techniques [29, 30, 31]. This approach is used in so-called intelligent tutoring systems [32]. Although, it relieves the work of the author, it inevitably also introduces in-transparency. The author does not know in advance how the VLE will be adapted. The last approach, the *teacher-driven approach*, is similar to the author-driven approach in the sense that it is human-driven but the specification of the required adaptation is done at run-time, i.e. while the learner is in action. This has the advantage that the teacher can respond to the particular situation of the learner and don't need to preview all possible adaptation scenario's, but it has the big disadvantage that the learning need to be supervised by a teacher which is in general too time consuming. This approach is therefore not used a lot.



**Figure 4: Adaptation Models**

What every approach is used, it requires a set of models to drive the adaptation process [33]. The models needed to drive the adaptive process in case of VR are illustrated in figure 4. First of all, it is necessary to assign attributes to the learning content, such as the level of difficulty, pre-requisites, learning time required (see e.g., the IEEE 1484.12.1 – 2002 Standard for Learning Object Metadata (LOM) for an elaborated set of such meta data). This information is needed to be able to adapt the content to the learning goals and to the learner. To adapt content to the learner, it is also necessary to have information about the learner (personal preferences, background knowledge, etc) and to keep track of his/her learning progress at runtime. This is usually done through the use of a *user-profile*, a *user model*, or *learner model*. In general, meta data for learning objects and user data are the two main models required for driving the adaptation process. However, in case of a VLE, it is also necessary to keep track at runtime of what is happening in the virtual world. For instance, to be able to use some of the adaptation strategies mentioned in section 4, it is necessary to keep track of the objects with which the learner has interacted, or which behaviors have been performed, or the time spent with a certain object or in a

certain part of the VLE. For example, to be able to perform the adaptations associated with the adaptation strategy *displayAtMost*, the system needs to keep track of how often the object has already been displayed to the learner. We call this information *VLE activity history*. The more information is kept about the activities of the learner in the VLE, the more it can be taken into account to adapt the VLE to the individual learner. Note that some data from this VLE activity history will also impact the state of the learner model, for instance activities performed in the VLE by the learner may raise his knowledge level of a certain topic.

## 6 Prototype and Lessons Learned

To be able to experiment with the adaptive VLEs, we developed a prototype implementation for the approach. This prototype implementation is based on an existing Web-based adaptive learning environment [34], developed in the context of an EU FP7 STREP called GRAPPLE. GRAPPLE is using an author-driven approach for specifying the adaptations. Therefore, we have also followed this approach in our prototype. As the learning environment is Web-based, the VR format supported is X3D [36]. The actual delivery of the adaptive VE is done using an existing VR player Vivaty [37] inside a Web browser.

We will now briefly elaborate on the different components of this prototype and on lessons learned in developing and using this prototype.

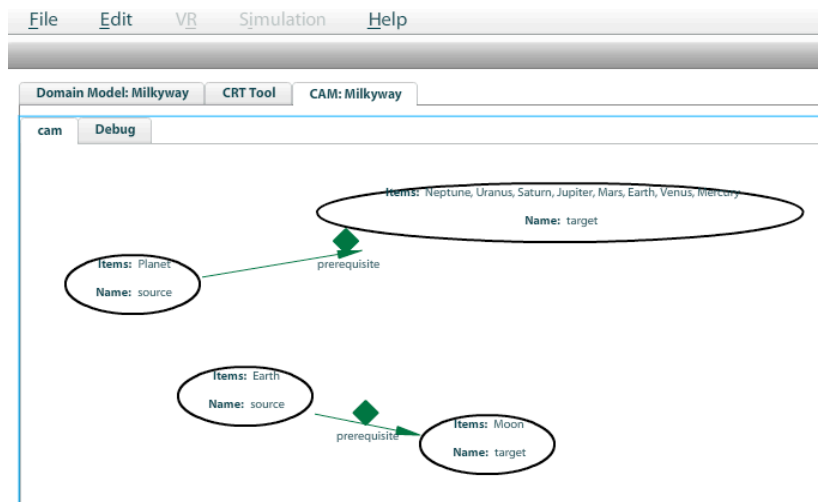
The two main components of GRAPPLE are the *Authoring Tool* and the *Adaptive Delivery Environment*. The Authoring Tool allows a course author to define a course. Therefore, the author needs to define a *Domain Model* and a *Conceptual Adaptation Model* [35]. The Domain Model describes the concepts that should be considered in the course. Learning resources are associated with these learning concepts. The Conceptual Adaptation Model expresses at a high-level how the content and structure needs to be adapted at runtime. The kernel of the Adaptive Delivery Environment is the *Adaptive Engine*. Based on the state of the learner's profile (captured in the *User Model*) and the specifications given in the Authoring Tool, the Adaptive Engine will select the proper learning resources and deliver the required navigation structure and content to a Web browser. Also note that the Adaptive Engine can keep track of the progress of the learner and will inform the User Model of this. Updates in the User Model may trigger new adaptation rules specified in the Conceptual Adaptation Model and in this way the course will be adapted at run-time. In GRAPPLE, the control over the User Model is left to the author, i.e. the author can decide which information to maintain in the User Model and how and which learning activities should update that information. GRAPPLE also allows importing information about the learner from external learning management systems (see [34]).

In order to support adaptive VR, it was necessary to extend GRAPPLE. GRAPPLE is a web-based learning environment using XML for the learning resources. Therefore, for displaying VR inside a browser, X3D [36] (which is XML-based) can be used. However, to support adaptive VR, two extensions were necessary. To be able to allow authors to specify how adaptation should happen inside a VLE an adapted authoring tool was necessary. Next, an extension of the adaptive delivery environment

was needed to allow for the actual adaptive delivery of the VE, i.e. adapting the presentation of the objects, enabling and disabling behaviors and interaction, including objects conditionally, and/or providing dedicated navigation possibilities in the virtual world. We will not discuss the extension of the adaptive delivery, as this was rather a technical issue, but we do mention difficulties encountered when extending the authoring tool, as they are more at the level of usability. We also discuss an example adaptive VLE elaborated with the prototype. We end this section with some conclusions.

## 6.1 Authoring the Adaptation

For the authoring tool, we started by following the authoring approach of GRAPPLE. Let us start by briefly explaining this approach (more information can be found in [35]). GRAPPLE allows an author to specify the required adaptation using *pedagogical based adaptation rules*. We illustrate this concept with an example, a “prerequisite” adaptation rule. This adaptation rule is based on the pedagogical relationship “prerequisite” that may exist between topics (concepts) in a course. This adaptation rule specifies that if concept A is a prerequisite for concept B, the course material for concept B should be hidden as long as the knowledge level for concept A is not above a certain threshold (details are omitted).



**Figure 5: GRAPPLE Adaptation Rules Examples**

Important to know is that those rules are predefined. To be reusable in different courses, they are defined in a generic way, i.e. using placeholders for the different concepts. In the Conceptual Adaptation Model, the author can then use (i.e. instantiate) such an adaptation rule by filling in the placeholders with concrete concepts (from the Domain Model). For instance, the author can specify that the learning material for concept “Earth” should be hidden as long as the learner doesn’t have enough knowledge about the concept “Planet” by using this “prerequisite”

adaptation rule. In this case, the author instantiate this adaptation rule by replacing the placeholders by the concepts “Planet” and “Earth”. Figure 5 shows an example. It is a screenshot of the GRAPPLE authoring tool. In this tool, a graphical notation is used for the adaptation rules. On the canvas, we see two applications of the “prerequisite” rule. The upper one states that the concept Planet is prerequisite for the concepts Neptune, Uranus, Saturn, Jupiter, Mars, Earth, Venus, and Mercury. The lower one states that the concept Earth is a prerequisite for the concept Moon.

In the Conceptual Adaptation Model, the adaptation rules are given in an order-independent way. It is actually GRAPPLE’s adaptive engine that will figure out when to use which adaptation rule (based on the conditions in the rules). In addition, it is also the adaptation engine that selects the most appropriate learning resource for a concept. For instance, a learner with no or low knowledge of the concept “Planet” should be given an “introductory” learning resource for the concept “Planet”. The selection is done by matching the meta data of the learning objects with the ones required by the adaptation rules. Actually the exact content of the course is composed on the fly by the adaptation engine. So, GRAPPLE is using a declarative approach for specifying an adaptive course. This is a powerful approach, as the author of a course only needs to specify the desired results and not how this needs to be achieved. Although, at first sight this declarative approach looked applicable for VLEs, we encountered several problems when applying it:

1. The first problem that we encounter is that adaptation in VLEs cannot be specified at the level of concepts (as done in GRAPPLE) but needs to be specified at the level of individual concepts, behaviors, and so on. While in a classical text-oriented learning environment it is easy to compose a page by combining different pieces of text or adapt a page by replacing one piece of text by another, this is not straightforward in a VLE. Indeed, let’s go back to the example of the planets. There may be different 3D representations of Earth (being possible learning resources for the concept “Earth”), however they may not all fit in a given VLE. They may be too large or too small or the texture used may not be appropriate. The same applies for behaviors. It is not always possible to replace one behavior by another. Therefore it is not possible to let the adaptive engine autonomously replaces VR resources, and it is certainly not possible to let the adaptive engine compose the actual VLE as all 3D objects should fit together and need to be positioned and oriented in the 3D space with care. It is even not possible to leave the selection of the appropriate learning resources to the adaptive engine as currently the meta data used for the learning objects is not capable to specify 3D specific issues (such as size, texture, etc.) needed to be able to do an appropriate selection. This forced us to adapt the format of the adaptations rules to make it possible to use them at the level of the individual resources. More in particular, instead of having placeholders for concepts, we need placeholders for the combination concept, resource.
2. Next, there are much more possibilities for adapting a VLE (see section 3) than for adapting a text-based course. To be able to allow authors to fully exploit all these possibilities, it would be necessary to predefine a large set of pedagogical based adaptation rules. Actually, a rule needs to be predefined for each possible combination of pedagogical situation and adaptation technique. While for text or figures, there are only a few adaptation techniques for indicating a pre-requisite relation (i.e. hiding or make it inaccessible), there are plenty of possibilities in a

VLE (hiding, semi-hiding, changing size, disabling behaviors, disabling interaction, using marking, etc.). This means that while for text-based adaption, there will only be two adaptation rule associated with the pre-requisite relation, there can be many more adaptations rules associate with this pre-requisite relation in case of VLEs. It is easy to see, that already for only a few pedagogical situations, providing a generic adaptation rule for each possible combination would result in a large and unwieldy set of predefined adaptation rules.

3. Another important obstacle was the fact that in the Conceptual Adaptation Model, the adaptation rules are specified in an order independent way. This was confusing for authors of a VLE as they usually have a certain storyline in mind that they want the learner to follow. For instance, in case of a course about the solar system (see section 6.3 for this example), an author wanted to start by taking the learner on a tour though the VLE representing the solar system, then he wanted to allow the learner to study the generic concepts Sun, Planet and Satellite. During that phase, all behavior and interaction would be disabled. While studying such a concept (e.g., Planet), examples of the concept (Earth, Mars, Venus, ...) would be marked. Next, he wanted the student to study about the inner solar system and then he would remove all objects not belonging to this inner solar system. In a similar way, he would allow to learner to study the outer solar system, and so on. Unfortunately, it was rather impossible for the author to specify this scenario using the approach used in GRAPPLE because it doesn't allow to specify adaptations in an order dependent way. Complex and artificial conditions were needed to realize this.

## 6.2 Example VLE

To test the prototype, an example adaptive VLE has been developed. The VLE is part of a course on the solar system. The course is a combination of textual learning material about the solar system and a VLE of the solar system where the sun and different planets are displayed in 3D. The textual material is displayed in one frame of the browser where the learner can navigate through the textual material using standard navigation, i.e. a menu and hyperlinks. He is also able to navigate (using VR navigation techniques) through the VLE displayed in another frame (see figure 6). Using the Conceptual Adaptation Model, the author has specified how the VLE should adapt according to the learner's knowledge level.

As already explained in section 6.1, the authoring of the adaptation was not easy. However, we were able to show that the VLE could be adapted dynamically based on the User Model, the VLE-activity history and the Conceptual Adaptation Model. For instance, we specified that at the start the names of the planets familiar to the learner should be in green, and the names of the planets that the learner still needs to study should be in red. Then the learner can start the course and he will be presented with a classical menu in the left panel of the browser window and an empty VLE (complete black). If he clicks on the Sun-link, a textual explanation of the sun will be displayed in the text frame. It was specified in the Conceptual Adaptation Model that the text describing the sun should be extended each time the learner visit the Sun-page, and once the learner has seen the complete textual explanation, a 3D model of the sun will be displayed inside the VLE (see figure 6). The other planets will appear in the same

way. Furthermore, planets that are studied completely will stop rotating. When all planets have been studied, the learner will see the complete solar system in the VLE and is able to explore it freely.

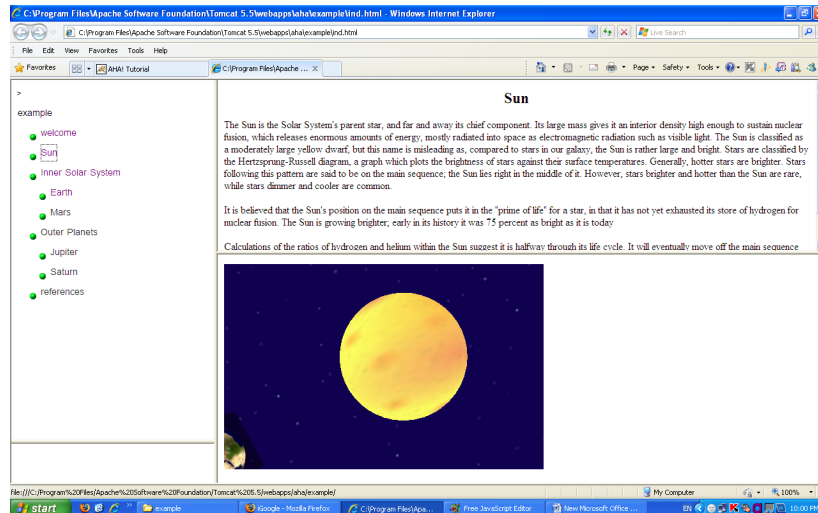


Figure 6: Example Course

Below are some examples of adaptation rules used in the prototype. To make it easier for the reader, we are not using the format of GRAPPLE, but we have expressed the rules as if-then rules. The first example is a rule that specifies that if the user's knowledge for the concept Earth is above the minimal required level (which is expressed by means of an integer value), then the VR object Earth should be highlighted in the VLE. The goal of this adaptation is to draw the attention of the learner to the fact that he reached the required minimal level of knowledge for Earth.

**Example adaptation rules:**

*if (user\_knowledge (Earth) > min\_required)*  
*then { highlight(Earth)}*

*if ((user\_knowledge(Earth) > min\_required)*  
*AND (hasInteractedWith(Earth) > 3)*  
*then { spotlight(Mars), Annotate(Mars), behaviour(AroundOrbit, Mars)}*

The second rule is to state that if the user's knowledge for the concept Earth is above the minimal required level and he has interacted with the planet Earth more than 3 times, then a spotlight should be set on the VR-object Mars, a certain annotation should be displayed with it, and it should start to move around the sun. The goal of this adaptation is to draw the attention to the next subject to study.



### 6.3 Conclusions for the Prototype

Because the adaptations that we could specify with the current authoring approach are limited and also cumbersome to achieve (see section 6.1), we decide to postpone the planned evaluation of the adaptive VLE itself. We found it not useful to set up an elaborated experiment with end-users to validate the usability of an adaptive VLE when we are not satisfied with the type of adaptations that can be specified (and therefore achieved). Therefore we decided to first work on an improve approach for authoring the adaptation. In that approach, it will be possible for the author to compose a time-based storyline. Furthermore, we will opt for adaptation rules that are easier to understand by an author by untangle the pedagogical aspects (the conditions) from the adaptation effects, and which the authors themselves can compose (hence removing the need for predefined adaptation rules). Figure 7 shows a screenshot of this new authoring tool. The scenario modeled in this figure is as follows. The course starts by marking the VR object Earth, and then it will start rotating around its axis. Once this VR object is rotating and the user has interacted with it by clicking on it, the VR object Earth will start to move around its orbit. If now, the user interacts again with Earth it will be unmarked and instead the VR object Mars will be marked and the viewpoint of the user will be moved towards Mars. Note that this scenario doesn't have any pedagogical meaning but was only created to verify the feasibility of the authoring approach. The squares with the letters AB denote Adaptations Blocks. They are used to specify the adaptations required for the different VR objects.

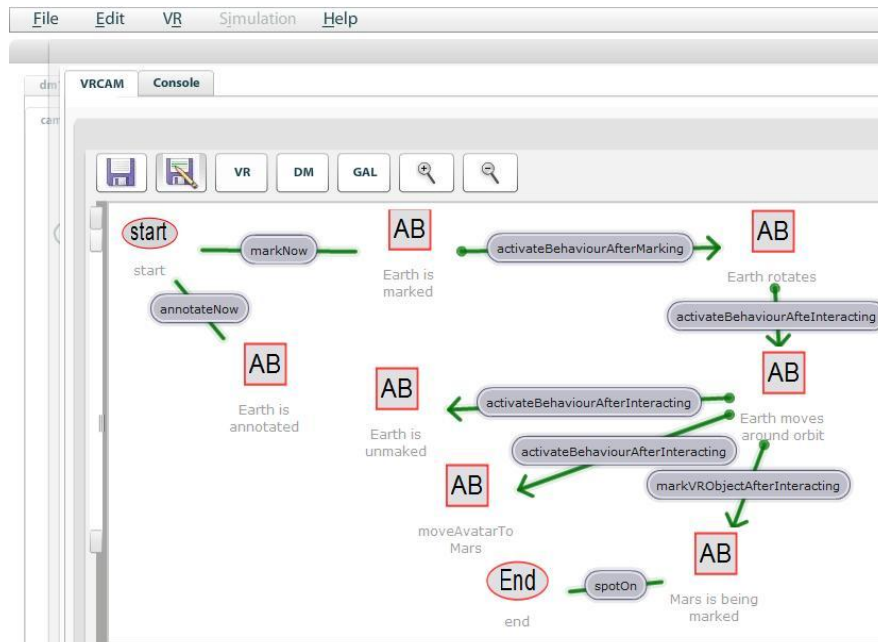


Figure 7: Time-based Authoring Approach

Once we have finished the prototype for this authoring approach, we will perform an empirical evaluation to study the usability both of the author tools and an actual virtual learning environment. In a first phase, the evaluation will be rather exploratory with the aim of identifying weaknesses and gathering ideas for recommendations for improvement. Later, an explicit, experimental comparison will be performed to measure user acceptance and the impact of adaptivity on the actual learning process.

## 7 Conclusions and Future Work

We have discussed the potential of adaptivity for VR learning environments. As in classical, text-based learning environments, adaptivity could be used to adapt a VR learning environment to the individual characteristics and preferences of the learners, their background, learning goals etc. However, the richness of Virtual Reality allows much more possibilities for adaptations than possible in a classical text-based learning environment. Therefore, we introduced a set of adaptation types for VR, as well as a set of adaptation strategies specific for VR learning environments. Next, we presented the different models needed for the adaptive process. We also discussed a prototype implementation of an adaptive VR learning system. The prototype is based on an existing Web-based adaptive learning environment that is author-driven (meaning that the author of a course specifies at design time the required adaptations) and its authoring tool uses a declarative approach, based on pedagogical-based adaptation rules. This turned out to be quite difficult for authors of virtual learning environments. First of all, it was difficult to realize a storyline (often used in the virtual environments). Next, the richness of the adaptations possibilities for VR resulted in a large and unmanageable set of pedagogical-based adaptation rules. This was due to the fact that in the approach used, the pedagogical issues and adaptation types are entangled in a single adaptation rule. Therefore, we are currently working on a different authoring approach using a time-based storyline and adaptation rules that are easier to express. Next, an empirical evaluation will be performed to study the usability both of the author tools and an actual virtual learning environment.

**Acknowledgments.** The work described in this paper is realized in the context of the EU FP7 project GRAPPLE (215434).

## References

1. Second Life, <http://secondlife.com>
2. Active Worlds, <http://www.activeworlds.com/>
3. Dede, C., Salzman, M., and Loftin, B.: The Development of a Virtual World for Learning Newtonian Mechanics. In P. Brusilovsky, P. Kommers, and N Streitz (Eds.), *Multimedia, Hypermedia, and Virtual Reality*. LNCS Volume 1077, Berlin: Springer (1996)

4. Cobb S.V.G., Neale, H.R., and Reynolds, H.: Evaluation of virtual learning environments. Proceedings of the 2nd European Conference on Disability, Virtual Reality & Assoc. Tech., Skövde, Sweden, ECDVRAT and University of Reading, UK, pp. 17-23 (1998)
5. Di Blas, N., Paolini, P. and Poggi, C.: SEE (Shrine Educational Experience): an Online Cooperative 3D Environment Supporting Innovative Educational Activities. In D. Lassner & C. McNaught (Eds.), Proceedings of Educational Multimedia, Hypermedia and Telecommunications 2003, pp. 1527-1534, Chesapeake, VA: AACE. (2003).
6. Dede, C., Ketelhut, D. J., and Ruess, K.: Motivation, usability, and learning outcomes in a prototype museum-based multi-user virtual environment. In P. Bell, R. Stevens & T. Satwicz (Eds.), *Keeping learning complex: The proceedings of the Fifth International Conference of the Learning Sciences*. Mahwah, NJ: Erlbaum (2002)
7. Bricken, M. and Byrne, C. M.: Summer students in virtual reality: A Pilot Study on Educational Applications of Virtual Reality Technology. In Wexelblat, A. (ed.), *Virtual Reality: Applications and Exploration*. New York: Academic Press, Inc., pp.199-218 (1993)
8. Virvou, M. and Katsionis, G: On the usability and likeability of virtual reality games for education: The case of VR-ENGAGE. In: Computers & Education, Volume 50, Issue 1, January 2008, pp. 154-178 (2008)
9. Chittaro, L., Ranon, R.: Adaptive Hypermedia Techniques for 3D Educational Virtual Environments. IEEE Intelligent Systems, vol. 22, issue 4, pp. 31-37 (2007)
10. Mushtaha, A., De Troyer, O.: Cross-cultural understanding of content and interface in the context of e-learning systems. In: Nuray M. Aykin (eds.) Usability and Internationalization. HCI and Culture, LNCS, vol. 4559/2007, pp. 164-173. Springer Berlin / Heidelberg (2007)
11. Google 3D Warehouse, <http://sketchup.google.com/3dwarehouse>
12. Google SketchUp, <http://sketchup.google.com>
13. Murdock, K.L.: 3ds max 5 bible. Wiley Publishing (2003)
14. Bowman, D., Gabbard, J., and Hix, D.: A Survey of Usability Evaluation in Virtual Environments: Classification and Comparison of Methods. In: *Presence: Teleoperators and Virtual Environments*, vol. 11, no. 4, pp. 404-424, (2002)
15. Tromp, J.G., Steed, A., Wilson, J.R.: Systematic Usability Evaluation and Design Issues for Collaborative Virtual Environments. In: Presence, Vol. 12, No. 3, pp. 241-267, (2003)
16. Brusilovsky, P., Hughes, S., Lewis, M.: Adaptive Navigation Support in 3-D E-Commerce Activities. In: Proceedings of Workshop on Recommendation and Personalization in eCommerce at the 2nd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH'2002), Malaga, Spain, pp. 132-139. (2002)
17. Chittaro, L., Ranon, R.: Adding Adaptive Features to Virtual Reality Interfaces for E-Commerce. In: Proceedings of International Conference on Adaptive Hypermedia and Adaptive Web-based Systems. LNCS, vol. 1892, pp. 86-97. Springer-Verlag, Berlin (2000)
18. Chittaro, L., Ranon, R.: Dynamic Generation of Personalized VRML Content: a General Approach and its Application to 3D E-Commerce. In: Proceedings of 7th International Conference on 3D Web Technology, Web3D 2002, pp. 145-154. ACM Press (2002)
19. Chittaro, L., Ranon, R.: Adaptive 3D Web Sites. In: Brusilovsky, P., Kobsa, A., Nejdil, W. (eds.) *The Adaptive Web - Methods and Strategies of Web Personalization*, LNCS, vol. 4321 pp. 433-464. Springer-Verlag, Berlin (2007)
20. Chittaro, L., Ranon R.: Using the X3D Language for Adaptive Manipulation of 3D Web Content. In: Proceedings of the 3rd International Conference on Adaptive Hypermedia and Adaptive Web-based Systems, pp. 287-290. Springer-Verlag, (2004)
21. Chittaro, L., Ranon, R.: Adaptive Hypermedia Techniques for 3D Educational Virtual Environments. IEEE Intelligent Systems, vol. 22, issue 4, pp. 31-37 (2007)
22. Chittaro, L., Ranon R.: An Adaptive 3D Virtual Environment for Learning the X3D Language. In: Proceedings of the 2008 International Conference on Intelligent User Interfaces (IUI 2008), pp. 419-420. ACM Press, New York (2008)

23. Santos, D., C. T., Osorio, F. S.: AdapTIVE: An Intelligent Virtual Environment and Its Application in E-Commerce. In: Proceedings of 28<sup>th</sup> Annual International Computer Software and Applications Conference (COMPSAC'04), pp. 468-473. (2004)
24. Celentano, A., Pittarello, F.: Observing and Adapting User Behaviour in Navigational 3D interface, In: Proceedings of 7<sup>th</sup> International Conference on Advanced Visual Interfaces (AVI 2004), pp. 275-282. ACM Press (2004)
25. Dachsel, R., Hanz, M., Meissner, K.: Contigra: an XML-based architecture for component-oriented 3D application. In: Proceedings of the 7<sup>th</sup> international conference on 3D web technology, pp.155-163. ACM Press (2002)
26. Kickmeier-Rust, M.D., Peirce, N., Conclan, O., Schwarz, D., Verpoorten D., and Albert, D.: Immersive Digital Games: The Interfaces for Next-Generation E-Learning? Universal Access in Human-Computer Interaction. Applications and Services, LNCS 4556, pp. 647-656, Springer (2007)
27. Law, E. L.-C., and Kickmeier-Rust, M. D.: 80Days: Immersive digital educational games with adaptive storytelling, Proceedings of the First International Workshop on Story-Telling and Educational Games (STEG'08) - The power of narration and imagination in technology enhanced learning, R. Klamma, R, et al. (Eds.) (2008)
28. Gutierrez, M.A., Vexo, F., Thalmann, D.: Stepping into Virtual Reality: A Practical Approach, Springer (2008)
29. Manslow, J.: Using Reinforcement Learning to solve AI Control Problems, In AI Game Programming Wisdom 2, ed. S. Rabin, Charles River Media, Inc, pp. 591-603, (2004)
30. Manslow, J.: Learning and Adaptation, In AI Game Programming Wisdom, ed. S. Rabin, Charles River Media, Inc, pp. 557-566. (2002)
31. Funge, J.D.: AI for Games and Animation: A cognitive Modeling Approach, A K Peters, Ltd (1999)
32. Sleeman, D. and Brown J.S. (Eds.): Intelligent Tutoring Systems. Academic Press. New York (1982)
33. Paramythis, A. and Loidl-Reisinger, S.: Adaptive Learning Environments and e-Learning Standards. Electronic Journal of e-Learning 2(1), pp. 181-194, (2003)
34. De Bra, P., Smits, D., van der Sluijs, K., Christea, A., Hendrix, M.: GRAPPLE, Personalization and Adaptation in Learning Management Systems. In: ED-MEDIA, World Conference on Educational Multimedia, Hypermedia & Telecommunication (2010)
35. Hendrix, M., De Bra, P., Pechenizkiy, M., Smits, D., Cristea, A.: Defining adaptation in a generic multi layer model: CAM: The GRAPPLE Conceptual Adaptation Model. In: 3rd European Conference on Technology-Enhanced Learning EC-TEL 2008. Springer LNCS, vol. 5192/2008, pp. 132 – 143. Springer-Verlag, Berlin / Heidelberg (2008)
36. Brutzman, D., Daly L.: X3D: Extensible 3D graphics for Web Authors. The Morgan Kaufmann Series in Interactive 3D technology (2008)
37. Vivaty, <http://www.vivaty.com/>