Supporting Virtual Reality in an Adaptive Web-Based Learning Environment

Olga De Troyer, Frederic Kleinermann, Bram Pellens, and Ahmed Ewais

Vrije Universiteit Brussel, WISE Research Group, Pleinlaan 2,
1050 Brussel, Belgium
{Olga.DeTroyer,Frederic.Kleinermann,Bram.Pellens,
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 world. Therefore, being able to dynamically adapt the virtual world to the personal preferences, knowledge, skills and competences, learning goals and the personal or social context of the learning becomes important. In this paper, we describe how an adaptive Web-based learning environment can be extended from a technological point of view to support VR.

Keywords: Virtual Reality, E-Learning, Adaptive Learning Environment.

1 Introduction

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 teaches how to fly.

Most of the time, when VR is considered for learning, it is offer as a stand-alone application (e.g., [1], [2], [3]) and there is usually no way to adapt it to personal preferences, prior knowledge, skills and competences, learning goals and the personal or social context of the learner. Augmenting a virtual world with adaptive capabilities could have many advantages [4]. It may be more effective to guide learners through the world according to their background and learning goals, or only show them the objects and behaviors that are relevant for their current knowledge.

In this paper, we explain how VR can be supported in the context of an adaptive Web-based learning environment developed in the context of GRAPPLE, an EU FP7 project. GRAPPLE is mainly oriented towards classical learning resources, but the use of other types of learning materials (VR and simulations) is also investigated. Here, we concentrate on how the learning environment is extended to support VR.

2 The GRAPPLE Architecture

GRAPPLE aims at providing a Web-based adaptive learning environment. The two main components are the *Authoring Tool* and the *Adaptive Engine* (see figure 1). The

U. Cress, V. Dimitrova, and M. Specht (Eds.): EC-TEL 2009, LNCS 5794, pp. 627–632, 2009.

Authoring Tool (Web-based) allows a course author to define a course at a conceptual level. This is done by means of a (graphical) *Domain Model* (DM) and *Conceptual Adaptation Model* (CAM*)* [5]. The DM describes the concepts that should be considered in the course. The CAM expresses at a high-level and by using pre-defined pedagogical relations (such as the *prerequisite* relation) how the content and structure needs to be adapted at runtime. The authoring tool can also be used (by a more experienced person) to define new pedagogical relations, called CRTs. Defining a CRT also implies defining the adaptive behavior associated with the relation. Different adaptive behaviors may be possible for the same pedagogical relation, e.g., the prerequisite relation ("A is prerequisite for B") can be associated with an adaptive behavior that hides the dependent concept B as long as A has not been studied, or with an adaptive behavior that forces the learning sequence A than B. Next, the graphical CAM is translated into a format (called GAL – Generic Adaptation Language) that the adaptive engine can handled. During this translation, the adaptive behaviors, associated with the definition of the pedagogical relations, are used to express the desired adaptive behavior for the course.

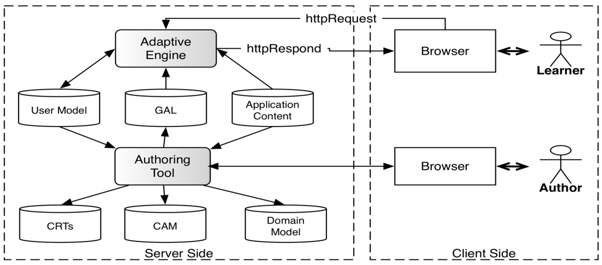

Fig. 1. GRAPPLE Architecture

The Adaptive Engine does the actual adaptive delivery of the content. Based on the state of the learner's profile (captured in the *User Model*) and the GAL specifications, the Adaptive Engine will select the proper learning resources and deliver the required navigation structure and content to a Web browser. The Adaptive Engine also keeps track of the progress of the learner. Internal variables are maintained in order to be able to instruct the User Model service what to update in the learner's User Model. This allows runtime adaptation. The adaptive engine of GRAPPLE is implemented as a client-server application. All its functionality is located at the server-side.

3 Adaptive VR in the Context of GRAPPLE

To support VR, it was necessary to extent the architecture of GRAPPLE. The VR material considered may range from some simple 3D objects to complete 3D virtual environments (VE) in which the user can navigate and interact with the 3D (or 2D) objects in it. Because GRAPPLE is a web-based learning environment, it is using the XML format for the learning resources. For displaying 3D objects inside a browser, X3D [6] can be used, which is XML-based. Therefore, individual 3D resources can be included or excluded in the same way as the regular XML resources. However, to support adaptive VE's, extensions are necessary for the authoring tool and for the adaptive engine. These extensions are necessary because the regular GRAPPLE tools consider a resource as a black box. To be able to adapt the VE itself, i.e. adapting the presentation of the objects in the world, enabling and disabling behaviors and interaction, including objects conditionally, and/or providing dedicated navigation possibilities in the virtual world, this approach is no longer suitable. Although some of the adaptations can be seen as extensions of adaptations for text resources, they are very specific for 3D material and the possibilities are much richer. For instance, to visually indicate that an object has not yet been studied, we may want to give it a different color or make it smaller; when a learner is studying a complex object (like a planet of the solar system), the visual appearance of the object could change according to the aspects being studied (size, temperature, geography, …) or become more detailed while more and more knowledge is acquired. It may also be necessary to disable or enable behavior and the interaction for an object according to the progress of the learner. To allow specifying this, a dedicated VR authoring tool is necessary. Furthermore, it is necessary to provide a preview of the VE to the author while he is specifying the adaptations because otherwise he needs to specify adaptations blindly, which may be very difficult. For instance, it is not possible to replace one 3D object in a VE by any other 3D object, as this object may not fit into the VE. Also behaviors are usually strongly connected to the actual 3D object, and it is not always possible to replace a behavior by any other behavior. The VR authoring tool is also implemented as a Web application (see figure 2). It allows specifying VR-specific CRTs and has a component to define CAMs. The VR specific authoring tool does not need a specific DM component; it uses the DM tool of the general authoring tool. Note the availability of a *Previewer*. The *Loading component* is responsible for retrieving the necessarily information from the different repositories (using available web services) and for retrieving the VR resources that needs to be previewed. The *Saving component* is responsible for storing all the information defined by the author, i.e. newly defined CRTs and CAM specifications. The output format is GAL but we also have an independent XML format to be able to connect to other systems.

The extension of the adaptive engine towards VR is realized by means of a browser plug-in (see figure 3) responsible for (1) updating the VE if the adaptive engine instructs to do so, (2) monitoring the learner's behaviour, and (3) sending information about the learner's behavior to the Adaptive Engine. For the retrieval of VR content by the VR browser plug-in, a server-side plug-in is added, the *VR-Manager*.

The VR browser plug-in has three components namely the *Monitor component*, the *Update component,* and the *VR player*. An existing VR player is used to visualize the VR content. We require that the VR player supports Document Object Model (DOM) [7], JavaScript [8], Scene Authoring Interface (SAI) [9], and X3D [6]. The scene Authoring Interface is used to communicate to the VR player. In this way, it is possible to update at runtime the scene graph without the need to reload it completely. The Update Component is responsible for interpreting the adaptation requests received from the Adaptive Engine and for translating it into a form that can be understood by the VR player; it instructs the VR player to update the scene. The Monitor Component is responsible for keeping track of what happens in the VE and translating this in a form understandable by the Adaptive Engine, which on its turn will inform the User Model service about the progress made by the learner.

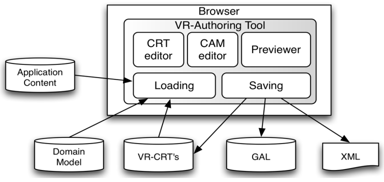

Fig. 2. VR Authoring Tool - Components

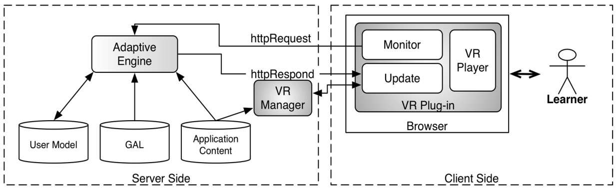

Fig. 3. Adaptive VR Delivery

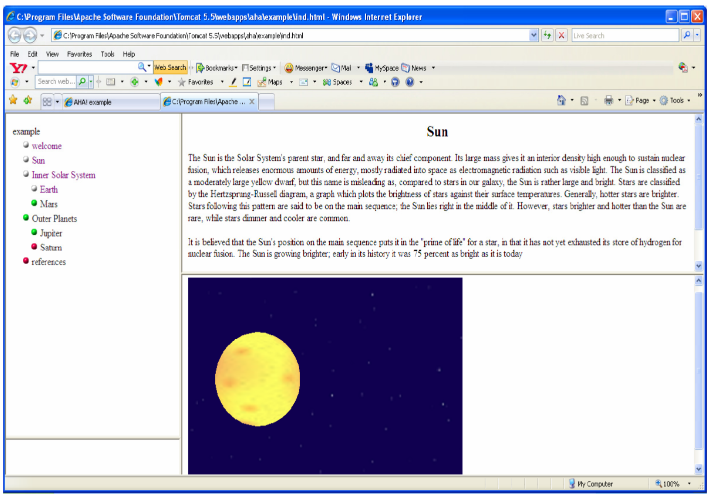

Fig. 4. Studying the Sun

To validate parts the adaptive delivery of the VR material, we have created a prototype and elaborated an example course with it. As the adaptive engine of GRAPPLE was not yet available at that time, the prototype is based on AHA! 3.0 [10], an adaptive learning engine on which the GRAPPLE adaptive engine will be based. As VR player we have used Ajax3D [11] that uses the Vivaty player [12]. It can use Ajax and can be embedded inside Firefox and Internet Explorer. Both the Monitor Component and the Update Component have been prototyped.

To test the prototype, an example adaptive course has been developed. The adaptive course is about the solar system. To investigate the issues related to the combination of different types of content, this course contains plain text explaining the solar system, as well as a VE of the solar system were the sun and different planets are displayed in 3D (see figure 4). The text as well as the VE will adapt according to the learner's knowledge and progress. E.g., a planet appears when the learner starts to study it; planets that have been studied will stop rotating; and when all planets are studied the whole solar system is available in the VE.

4 Related Work

Brusilovsky et al. [13] have integrated some adaptive hypermedia methods (mainly for navigation) into Virtual Environments. The approach of Santos and Osorio [14] is based on agents that help the user by providing him more information about interesting products, and by guiding him to their preferred area. The work of Moreno-Ger et al. [15] on 2D games is not on VR but is interesting as it provide an authoring tool allowing authors to create adaptive courses. Celentano and Pittarello [16] monitor a user's behavior and compare it with previous patterns of interaction. Whenever the system detects that the user is entering a recurrent pattern, it may perform some activities on behalf of the user. Chittaro and Ranon did quite some related work, first in the context of e-commerce, later on also for e-learning. Some of their work can be found in [17], [18] and [19]. The work in [19] is close to ours, and especially to the prototype that we have developed and which is also using AHA! (see section 3). However, they don't provide an authoring tool for specifying the adaptive story lines like we do. Furthermore, the main file containing the VE is reloaded at fixed time intervals to keep the adaptation inline with the student's user model. This is a serious drawback, especially for large VEs, because it will make the system very slow. In our approach, runtime adaptations don't require reloading the complete VE.

5 Conclusions and Future Work

This paper describes an approach to support the adaptive delivery of Virtual Reality learning material inside GRAPPLE, a Web-based adaptive learning environment. The approach is innovative from different aspects. Firstly, it contains a visual authoring tool for specifying the adaptive strategy for a VE. Next, the adaptation of the VE is done at run-time without the need to reload the VE each time, which will provide the necessarily performance required for large VE's. In addition, VR material and classical (textual and 2D multimedia) can be integrated in a single course and the adaptation can be performed for whatever type of content. The activities performed by the learner in the VE can be monitored and the effect can be directly reflected in the VE. First a prototype has been developed for the adaptive delivery. Currently, the implementation of the VR-authoring tool has been started as well as the implementation of the final VR-plug-in. Several experiments are planned to validate the approach as well as it usability and effectiveness.

Acknowledgments. This work is realized in the context of the EU FP7 project GRAPPLE (215434). The design of the overall GRAPPLE architecture has been a collaborative effort of the different partners.

References

1. Alexiou, A., Bouras, C., Giannaka, E., Kapoulas, V., Nani, M., Tsiatsos, T.: Using VR technology to support e-learning: the 3D virtual radiopharmacy laboratory. In: Distributed Computing Systems Workshops, 2004. Proceedings 24th International Conference, pp. 268–273 (2004)
2. KM Quest, `http://www.kmquest.net`
3. De Byl, P.: Designing Games-Based Embedded Authentic Learning Experiences. In: Ferdig, R.E. (ed.) Handbook of Research Effective Electronic Gaming in Education. Information Science Reference (2009)
4. Chittaro, L., Ranon, R.: Adaptive Hypermedia Techniques for 3D Educational Virtual Environments. IEEE Intelligent Systems 22(4), 31–37 (2007)
5. 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: Dillenbourg, P., Specht, M. (eds.) EC-TEL 2008. LNCS, vol. 5192, pp. 132–143. Springer, Heidelberg (2008)
6. Brutzman, D., Daly, L.: X3D: Extensible 3D graphics for Web Authors. The Morgan Kaufmann Series in Interactive 3D technology (2008)
7. W3C Document Object Model, `http://www.w3.org/DOM/`
8. JavaScript, `http://www.javascript.com`
9. Scene Authoring Interface Tutorial, `http://www.xj3d.org/tutorials/general_sai.html`
10. AHA! 3.0, `http://aha.win.tue.nl/`
11. Ajax3D, `http://www.ajax3d.org/`
12. Vivaty, `http://www.vivaty.com/`
13. 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)
14. dos Santos, C.T., Osorio, F.S.: AdapTIVE: An Intelligent Virtual Environment and Its Application in E-Commerce. In: Proceedings of 28th Annual International Computer Software and Applications Conference (COMPSAC 2004), pp. 468-473 (2004)
15. Moreno-Ger, P., Sierra-Rodriguez, J.L., Ferandez-Manjon, B.: Games-based learning in e-learning Environments. UPGRADE 12(3), 15–20 (2008)
16. Celentano, A., Pittarello, F.: Observing and Adapting User Behaviour in Navigational 3D interface. In: Proceedings of 7th International Conference on Advanced Visual Interfaces (AVI 2004), pp. 275–282. ACM Press, New York (2004)
17. Chittaro, L., Ranon, R.: Adaptive 3D Web Sites. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) Adaptive Web 2007. LNCS, vol. 4321, pp. 433–462. Springer, Heidelberg (2007)
18. Chittaro, L., Ranon, R.: Adaptive Hypermedia Techniques for 3D Educational Virtual Environments. IEEE Intelligent Systems 22(4), 31–37 (2007)
19. 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)