Vrije Universiteit Brussel

FACULTY OF SCIENCES & BIO-ENGINEERING SCIENCE
DEPARTMENT OF COMPUTER SCIENCE
RESEARCH GROUP WEB & INFORMATION SYSTEM ENGINEERING (WISE)

Authoring Adaptive 3D Virtual Learning Environments

Thesis submitted in fulfilment of the requirements for the award of the degree of Doctor in Sciences by

Ahmed Ewais

Brussels, 2013

Promoter: Prof. Dr. Olga De Troyer

Abstract

The use of 3D (3 Dimensional) Virtual Environments is gaining interest in the context of academic discussions on E-learning technologies, as it provides several advantages over classical learning material. However, the use of 3D for learning environments also has drawbacks, especially usability and the effectiveness may cause problems. One way to overcome these drawbacks is by providing an *adaptive* 3D Virtual Learning Environment (3D VLE), i.e. an environment that dynamically adapts to the learner and the activities that he performs in the environment. However, developing adaptive 3D Virtual Learning Environments is not simple and should be supported.

In this thesis, we discuss adaptive 3D VLEs and explain how a course author can specify (i.e. author) such an environment. The authoring approach that we present allows authors to create adaptive 3D VLEs *without the need to be an expert in 3D and without using programming or scripting languages*. In particular, the thesis elaborates on *the principles used for the authoring approach*, as well as on *the different aspects that need to be supported*, i.e. the pedagogical aspects, the adaptation aspects, and *the requirement to support the specification of an adaptive storyline* which should be followed by learners.

Consequently, the main objectives of the research work are: (1) to investigate the possible adaptations that can be applied to 3D VLEs, in order to allow adapting a 3D VLE to the learner's knowledge, skills, and behaviour; and (2) to study how to facilitate the authoring process of adaptive 3D VLE for 3D novice educators.

To achieve these objectives, we first present *a generic conceptual modelling framework* for this type of systems. In addition to that, *a number of adaptation types and strategies* are proposed that authors can use for specifying adaptivity in 3D VLEs. Furthermore, *the concept of adaptation theme* is proposed to ease the specification of consistent adaptations for an adaptive 3D VLE. The research work also proposes *visual languages* specifically conceived *to allow authors to specify adaptivity in 3D VLEs*. Namely, the proposed visual languages are the Pedagogical Model Language, the Adaptive Storyline Language, and the Adaptive Topic Language. Each language is used to specify a different aspect of the conceptual modelling framework. Within this research work, we also conducted *an evaluation to the proposed visual languages*. In particular, the conducted evaluations aimed at validating both usability and acceptability aspects related to the proposed visual languages. Finally, we also propose *a prototype of authoring tool* for supporting the proposed authoring approach and the visual languages.

The findings presented in the thesis can be applied by software engineers to construct authoring tools for designing adaptive 3D VLEs. In addition, the findings can also be used for further research.

Samenvatting

Het gebruik van 3D (drie-dimensionele) virtuele omgevingen kan op behoorlijk wat belangstelling rekenen in de context van E-learning, omdat het een aantal voordelen biedt in vergelijking met traditionele leermiddelen. Deze digitale omgevingen laten toe om het leermateriaal te visualiseren in 3D, simulaties te tonen en de leerling kan op allerlei manieren met het leermateriaal interageren. Het gebruik van 3D voor leeromgevingen kan echter ook nadelen hebben, vooral wat betreft gebruiksgemak en doelmatigheid. Voor leerlingen die niet bekend zijn met 3D kan een 3D omgeving overweldigend zijn en navigeren in de 3D ruimte en interageren met de 3D objecten kan moeilijk zijn. Daar en tegen kan voor gevorderde “gamers” de verleiding groot zijn om zich in de 3D omgeving bezig te houden met activiteiten die niet direct met leren te maken hebben. Een mogelijkheid om deze problemen op te lossen is het gebruik van adaptieve 3D virtuele leeromgeving (“3D *Virtual Learning Environment” of kortweg* 3D-VLE), d.w.z. een leeromgeving die zich dynamisch aanpast aan de behoeften van de leerling en diens activiteiten in de 3D-VLE. Het is echter niet makkelijk zulke adaptieve 3D-VLE’s te ontwikkelen, zeker niet voor niet specialisten. Nochtans is de betrokkenheid van pedagogisch geschoolden (bijv. leerkrachten in het vakgebied) belangrijk om tot goede en effectieve leeromgevingen te komen. Daarom is hiervoor ondersteuning nodig.

In deze thesis gaat het over hoe we niet-informatici (bijv. leerkrachten) kunnen betrekken bij het ontwikkelen van adaptieve 3D-VLE’s en meer bepaald hoe ze zelf een adaptieve 3D VLE kunnen ontwerpen. We leggen uit hoe een dergelijke leeromgeving kan worden gespecificeerd zonder dat men een expert in 3D moet zijn en bekend moet zijn met programmeertalen. Uit onderzoek is gebleken dat het nuttig is om het leerproces door middel van een 3D-VLE degelijk te structureren. Daarom laat de benadering die we voorstellen toe om een soort van verhaallijn te definiëren voor de te ontwikkelen 3D-VLE. Zowel de verhaallijn als de 3D-VLE zelf kan afgestemd worden op de behoeften en het gedrag van de leerling.

Om dit te realiseren, werden de volgende onderzoeksvragen opgesteld:

1) Welke mogelijkheden bestaan er om een 3D-VLE aan te passen aan de kennis van de leerling, diens vaardigheden en gedrag;
2) Hoe kan het ontwerpproces van een adaptieve 3D-VLE worden vergemakkelijkt zodat het bruikbaar is voor leken op het gebied van 3D.

Om deze onderzoeksvragen te beantwoorden hebben we eerst een *generisch conceptueel modelleerraamwerk* voor adaptieve 3D-VLE’s opgesteld. Daarnaast hebben we een aantal adaptatietechnieken en strategieën voorgesteld, die de

auteurs van 3D-VLE's kunnen gebruiken om de adaptatie van 3D-VLE's te specificeren. Verder wordt het begrip *adaptatiethema* voorgesteld om de consistentie van de adaptaties te vergemakkelijken. We stellen ook drie visuele modelleertalen voor, die speciaal zijn ontworpen om het specificeren van adaptieve 3D-VLE's te vergemakkelijken. De voorgestelde visuele talen zijn de *Pedagogische Modelleertaal*, de *Adaptieve Verhaallijntaal* en de *Adaptieve Topictaal*. Elke taal laat toe om een ander aspect van de adaptieve 3D-VLE te specificeren. Binnen het kader van dit onderzoek hebben we de voorgestelde visuele talen ook geëvalueerd. Deze evaluaties hadden tot doel zowel de toepasbarheid als de bruikbaarheid en aanvaardbaarheid van de voorgestelde modelleertalen te beoordelen. Bovendien stellen we een prototype van een "*authoring tool"* voor om de benadering softwarematig te ondersteunen.

De onderzoeksresultaten kunnen worden gebruikt om *authoring tools* voor het ontwerpen van adaptieve 3D-VLE's te implementeren en natuurlijk ook voor verder onderzoek.

Declaration

Parts of this thesis have been published:

- Ewais, A. & De Troyer, O. (Submitted on 2014). Usability Evaluation of Visual Modelling Languages for Authoring Adaptive 3D Virtual Learning Environment. International Conference on Computer Supported Education (CSEDU 2014).
- Ewais, A. & De Troyer, O. (2014). Authoring Adaptive 3D Virtual Learning Environments. International Journal of Virtual and Personal Learning Environments (IJVPLE). 5 (1).
- Ewais, A. & De Troyer, O. (2013). Authoring Storyline-based Adaptive 3D Virtual Learning Environments. In: Proceedings of the 5th International Conference on Computer Supported Education (CSEDU 2013). Aachen, Germany.
- Ewais, A. & De Troyer, O. (2013). Usability Evaluation of an Adaptive 3D Virtual Learning Environment. International Journal of Virtual and Personal Learning Environments (IJVPLE). 4 (1).
- De Troyer, O., Kleinermann, F. & Ewais, A. (2010). Enhancing Virtual Reality Learning Environments with Adaptivity : Lessons Learned. In: The 6th international conference on HCI in work and learning, life and leisure: workgroup human-computer interaction and usability engineering. 2010, Klagenfurt, Austria: Springer-Verlag, pp. 244–265.
- De Troyer, O., Kleinermann, F., Pellens, B. & Ewais, A. (2009). Supporting virtual reality in an adaptive web-based learning environment. In: Proceedings of the 4th European Conference on Technology Enhanced Learning Learning in the Synergy of Multiple Disciplines. 2009, Nice, France: Springer-Verlag, pp. 627–632.

Acknowledgments

Without the encouragement, support and help from wonderful persons, it would never turn into such enjoyable experience. Several persons made it possible to end up with this research work. Therefore, this is a great opportunity to express my endless gratitude to all those who stood by me during my research work.

I would like to express my sincere appreciation to my promoter, Prof. Dr. Olga De Troyer, the director of the Head of the Web & Information Systems Engineering (WISE) Laboratory of the Vrije Universiteit Brussel . I would like to thank her first for giving me the opportunity to work under her supervision. Her invaluable insights, suggestions and wise guidance have provided solid bases for this dissertation. She always gave me the motivation that I needed in order to end up with this research work. I remain amazed that she never hesitated to give me part of her time when I needed it. You have been great and wonderful promoter. Great thanks for you Olga.

Next, I would like to thank the jury members, Prof. Dr. Bernard Manderick, Prof. Dr. Karin Coninx, Prof. Dr. Beat Signer, Prof. Dr. Chang Zhu and Prof. Dr. Alexandra I. Cristea for providing me with good pointers to improve the quality of my work.

I owe my thanks to my parents. I thank them for their unlimited love and support throughout the past thirty-two years. They taught me a lot and without you I would have never achieved what I have become. My gratitude is also extended towards my father in law and my mother in law for their support.

I would like to offer my very grateful thanks for my beloved wife Bayan through her unwavering support, encouragement and patience during my pursuit of PhD study. I thank you for being there for giving me the power to finish this research. I would like also to thank my lovely two kids Omar and Karam who fill my days with happiness and joy. Three of you missed out on a lot of husband and father time while I was doing my research.

Gratitude is extended to my closest friend Dr.Abed Alghani Mushtaha. It was always great to chat with smart futurist and wonderful friend like you Abed. Those nice memorises will never be forgotten. You are my brother forever. I am also in debt to Dr.Marwan Alsafadi for his support and care through difficult times. I also extend special thanks to my friends Karam Damidi, Malek Obaid and Ahmed Tayeh.

I would like to acknowledge my former and current colleagues in WISE research group. I thank you for exchanging research opinion and discussions. I am also deeply indebted to my ex-colleague at WISE, Dr. Frederic Kleinermann.

I will not forget to thank European Union, Erasmus Mundus External Cooperation Window (EMECW) program for offering me the funding during this research work. I would like also to extend my great thanks to Prof. Dr. Kim Van Der Borght for his cooperation and support.

Dedication

To

The memory of my grandfather and grandmother

My parents and my brothers & sisters

My wife Bayan

My two kids Omar and Karam

Seek knowledge from the cradle to the grave

-Prophet Muhammed

Table of Contents

List of Figures

List of Tables

List of Acronyms

3D VE	3 Dimensional Virtual Environment
3D VLE	3 Dimensional Virtual Learning Environment
AEHS	Adaptive Educational Hypermedia Systems
AH	Adaptive Hypermedia
AHA!	Adaptive Hypermedia for All!
AM	Adaptation Model
API	Application Programming Interface
ASLL	Adaptive StoryLine Language
ATL	Adaptive Topic Model
DM	Domain Model
GRAPPLE	Generic Responsive Adaptive Personalized Learning Environment
IEEE	Institute of Electrical and Electronics Engineers
LAMS	Learning Activity Management System
PM	Pedagogical Model
PML	Pedagogical Model Language
PRT	Pedagogical Relationship Types
UM	User Model
UML	Unified Modelling Language
VR	Virtual Reality
VW	Virtual World
W3C	World Wide Web Consortium
WYSIWYG	What You See Is What You Get
XML	EXtensible Mark-up Language
XSL	EXtensible Style-sheet Language

Chapter 1: Introduction

1.1 Introduction

Evolutions in technology in the last two decades have helped researchers to explore the use of three-dimensional Virtual Environments (3D VE). As a result, more and more, 3D VEs are exploited in different domains such as training and simulation (Yusoff et al., 2011; Schneider et al., 2011; Moreno-Ger et al., 2010; Hubal, 2008), e-commerce (Hughes et al., 2002; Chittaro & Coppola, 2000; Chittaro & Ranon, 2002b; Lepouras & Vassilakis, 2006), e-learning (Chittaro & Ranon, 2007a; Wiecha et al., 2010; Georgouli, 2011; Chen et al., 2007), educational games (Göbel et al., 2009; Bourgonjon et al., 2010; Gee, 2007), and medical and health care (Kato, 2010; Schaefer et al., 2011; Wiecha et al., 2010; Hatem Abdul kader, 2011).

In general, 3D Virtual Environments or Virtual Reality (VR) are defined as three-dimensional (3D), multisensory, immersive, real time, and interactive simulations of a space that can be experienced by users via three dimensional input and output devices (Burdea, 2006). Another definition is pointed out by Gaddis who defines it as: "*a computer-generated simulation of the real or imagined environment or world.*" (Gaddis, 1998). While Vince described VR as "*Virtual Reality is about creating acceptable substitutes for real objects or environments, and it is not really about constructing imaginary worlds that are indistinguishable from the real world.*" (Vince, 2004).

Basically, 3D VEs can be divided into two categories: *advanced* 3D VE applications and *standard* 3D VE applications. The first category uses 3D VE with high-end technology in order to provide a feeling of presence and to enable interaction with the 3D world. The sort of technologies used for this group of applications are immersive displays (CAVE, VDesk) and interacting devices such as data-gloves, phantom, tracking systems. However, these technologies are still quite expensive and are often only used in specific situations, e.g., virtual prototyping, military training, surgical simulators, and scientific visualization. The second category consists of 3D VEs for low-end computer (such as PCs) and the Web. Thanks to recent advances in graphic cards, network communication and CPUs, 3D VE applications for PC are becoming more and more realistic in terms of graphics and can also provide good immersive feeling. Examples of such kind of 3D VE applications are Second Life[1], Active World[2], OpenSpace3D[3], OnLive! Traveler[4], but can also be found in the domain of serious games[5].

It is important to mention that the work presented in this thesis is dealing with the second category of 3D VEs, and more particularly focuses on browser-based 3D virtual environments, i.e. environments that are accessed through a web-browser interface (Chittaro & Ranon, 2007c).

Increasing attention is being paid to web-based 3D VE for educational purposes. We will refer to such environments as *3D Virtual Learning Environments* (3D VLEs) and these are the ones that we will consider in this dissertation. 3D VLEs are considered as promising and with high potential for enhancing the learning experience. This is due to the following important benefits:

- 3D VLEs can provide better insight in the learning material, or simply be more motivating, but in addition it also gives an enhanced feeling of presence (Slater, 1999; Dalgarno & Lee, 2010; Warburton, 2009). Since both learning and presence are strongly related to each other, a strong presence sensation, which can be experienced using 3D VLE, means increasing the learning benefits (Witmer & Singer, 1998).
- 3D VLE helps learners to acquire knowledge and skills through practicing specific tasks in a realistic environment and to achieve a more realistic understanding and representation of different learning topics (Chittaro & Ranon, 2007a).
- 3D VLE allows learners to train in a safe environment and perform experiments rather than running the risk of crucial damages or destructions, as would be the case in physical environment. The most famous example in this context is the flight simulator that teaches how to fly a plane safely under various circumstances (Dörr et al., 2000). Another important example is from the domain of medical training where learners can simulate surgical operations and study the human body without risk for the patient (Brodlie, 2000; Kruger et al., 1995).
- In collaborative 3D VLEs, users are able to communicate with each other using avatars or different chat techniques; they are also able to know what activities are performed by other learners inside the 3D virtual space

[1] http://secondlife.com
[2] http://www.activeworlds.com/
[3] http://www.openspace3d.com/
[4] http://www.digitalspace.com/
[5] http://www.seriousgames.org/index2.html

(Bronack et al., 2008; Molka-Danielsen, 2011). And as such they provide a good environment for collaborative learning or for practicing social skills.

Nevertheless the fact that 3D VEs represents a promising area for educational contexts, researchers exposed some obstacles and drawbacks related to the learning process, usability, efficiency, and accessibility aspects in 3D VLE (Di Blas et al., 2003; Dede et al., 2003; Bricken & Byrne, 1992; Chittaro & Ranon, 2007a). Others claimed that the richness of such a 3D VLE could also be its weakness (De Troyer et al., 2010). The drawbacks are discussed in the next section.

One of the possible solutions for diminishing different drawbacks and obstacles is to exploit adaptation techniques for 3D VLEs. Some approaches and frameworks have already been proposed in this context. For instance, researchers in (Dong et al., 2008; Dos Santos & Osório, 2004b) proposed approaches to adapt user interaction according to the user's different activities inside the 3D VLE. Others (Chittaro & Ranon, 2007a) suggest to adapt the 3D content based on the user profile. More reviewed literature is discussed in chapter two.

Furthermore, a large-scale use of adaptive 3D VLEs is hindered by a lack of easy to use development tools, especially tools that can be used by educators, i.e. so-called authoring tools. In addition, there is also a lack of experience on how to author such environments. Both problems will be tackled in this dissertation. We elaborate on the problem statement in the next section.

1.2 Problem Statement and Research Motivation

Although *adaptive* 3D VLE represents a promising area with high potential for improving web based educational applications, a number of studies have been done in this area outlining disadvantages and limitations. As stated in (Di Blas et al., 2003; Dede et al., 2003; Chittaro & Ranon, 2007a; De Troyer et al., 2010), adaptive 3D VLE is a new emerging trend and still under development. As a result, there are still open issues related to *providing better support for both the actual learning process and the usability aspects when 3D VLEs are dynamically adapted to an individual learner*. In addition, involving *educators in the development process of adaptive 3D VLE is hindered by some barriers*.

These issues were the motivation for our research. We discuss them in more detail.

Firstly, usability and learning are two goals that may conflict as observed by some researchers (Virvou & Katsionis, 2008; Burigat & Chittaro, 2007; Dede et al., 2003). 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. Most of the usability issues are related to 3D content presentations, user navigation, and user interaction.

With regards to 3D contents, 3D virtual environments tend to have magnificent views for 3D objects by using texture images with high bitmap resolution and different source lights. Furthermore, 3D contents may be displayed with a lot of detail. Such “fashion accessories”, which are needed to have appealing 3D VE, could also be considered as an obstacle that distracts the learner from the learning process.

Concerning user navigation and interaction inside 3D VLEs, the learner may be overwhelmed or get lost during his navigation (Bricken & Byrne, 1992), 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 (Burigat & Chittaro, 2007). On the one hand, for learners who are not familiar with 3D (novice users), the time required to get acquaint with such a 3D VLE may be long and therefore their short-term satisfaction may be low (Virvou & Katsionis, 2008). On the other hand, youngsters who are used to play video games and navigate in 3D worlds 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 (Virvou & Katsionis, 2008). Similarly, It was noted that students exhibit noticeable individual differences in their interaction styles, and abilities to interact with the 3D VE (Dede et al., 2003).

As mentioned earlier, one way to deal with these problems is by providing the virtual world in an adaptive way, i.e. adapted to personal preferences, prior knowledge and skills. To achieve this, we first need to investigate in what ways 3D VLEs can be adapted to provide this personalised experience.

Secondly, supporting or involving educators in the development of adaptive 3D VLE is still in its infancy. The difficulty of engaging instructors in designing adaptive 3D VLE can be considered as one of the major barriers to adopt adaptive 3D VLE widespread. Therefore, we consider this as a priority for our research.

Mostly, adaptive 3D VLEs are developed by well skilled programmers or researchers, and most of the adaptation mechanisms are realised by an ad-hoc implementation. In addition, nearly every adaptive 3D VLE has its own unique way to deliver 3D contents and materials adaptively. Educator who wants to develop an adaptive 3D VLE needs to resort to a programming language. However, in general, educators are not computer scientists or skilled programmers.

One way to allow authors (educators) to create educational and adaptive 3D VEs is by using authoring tools. In (Overmars, 2004), it was noted that involving educators in the development of educational 3D games can be achieved by providing user friendly, effective, and efficient authoring tools. In this manner, educators can guarantee that the developed game will provide the required level of usefulness and educational purposes (Bourgonjon et al., 2010). To the same end, supporting educators with user friendly and effective authoring tools for the adaptive 3D VLE can also better guarantee the targeted educational purposes and goals.

Although, there are already some authoring tools proposed to support educators in designing educational 3D VE, their support is limited in one way or another. For instance, the authoring tools Smiley (Jourdan et al., 1999) and GRiNS (Bulterman et al., 1999) allow an author to integrate educational materials inside a 3D virtual world by means of WYSIWYG paradigm. However, the tools are not suitable to provide adaptation for 3D VEs. Likewise, recent authoring tools like *PIVOTE*[6], *SimBionic*[7] are proposed to enable non-programmers to design adaptive 3D VEs with a specific storyline. However, the educator is not able to define an adaptive flow inside the 3D VE. More on the state of the art with respect to the authoring tools will be given in section 3.2.2.

1.3 Research Objectives and Scope

Based on the research problems discussed in the previous section, our research objectives are as follow: (1) studying *what can be adapted in a 3D VLE and how* in order to allow adapting a 3D VLE to the learner's knowledge, skills, and behaviour taking into account different usability issues mentioned in previous section; and (2) investigating *how we can support 3D novice educators with an authoring approach for creating adaptive 3D VLEs*.

Firstly, our work will investigate what can be adapted in general, in a 3D VLE and how. We will focus on adaptations that have the potential of improving the learning process and the personal experience. To achieve this, we will first investigate the different components of 3D VLE, i.e. the anatomy of a 3D VLE. Based on this anatomy, we will provide possible adaptation techniques that can be applied to 3D VLE components.

Secondly, our work aims at developing a modelling approach for adaptive 3D VLE authoring that allows 3D novice educators (i.e. not being experts in VR) to design adaptive 3D courses. The authoring process should hide low level technical details to reduce the effort and technical skills required for the authoring of adaptive 3D VLEs.

Concerning the scope of this PhD thesis, it is important to clarify the fact that the authoring approach that we target, does not consider creating the actual 3D contents and designing the 3D VE itself, as there are already many authoring tools to create 3D models. Furthermore, more and more 3D materials are becoming available for free on the Web (e.g., in Google's 3D Warehouse (Google3DWarehouse, 2006)) and easy to use tools are available to create such materials (e.g., Google SketchUp (GoogleSketchUp, 2000), 3D

[6] www.pivote.info/
[7] http://www.simbionic.com/products.htm

Studio Max (Murdock, 2003)). We will focus on authoring the adaptivity, as well as pedagogical aspects for existing 3D VEs.

Additionally, our work does not consider issues related to learning environments and adaptation that are already well studied, such as the concept of a domain model for a learning environment and the concept of a user model for adaptation. Work like (Aroyo et al., 2006a; Octavia et al., 2009; Dos Santos & Osório, 2004b; Celentano & Nodari, 2004; Kolb & Kolb, 2005; Kang & Tan, 2010) discuss different issues related to user modelling at length. Some interesting work related to domain modelling in the context of 3D VE can be found in (Bonis et al., 2007; Dachselt et al., 2006; Bilasco et al., 2007; Bille et al., 2004; Buche & Querrec, 2011; Hubal, 2008).

With regards to adaptation mechanisms, our work is mainly limited to adaptation towards user preferences, skills and knowledge. Other types of adaptation, for instance providing adaptive 3D VLE according to the user's device or location, are not explored in this work.

Furthermore, this research is targeting single-user 3D VLEs. Supporting adaptivity in the context of collaborative 3D VLEs is beyond the scope of this PhD dissertation.

1.4 Research Questions

Based on the observations made related to the identified research problems, this research work will provide answers to the following research questions:

RQ 1: What are possible adaptations that can be applied to 3D VLEs in order to allow adapting a 3D VLE to the learner's knowledge, skills, and behaviour?

RQ 2: How to facilitate the authoring process of adaptive 3D VLE for 3D novice educators?

RQ 2.1: What needs to be specified during authoring to allow for an adaptive 3D VLE?

RQ 2.2: What is an approach for authoring adaptive 3D VLE usable for 3D novice educators ?

RQ 2.3: How to support 3D novice educators in specifying adaptive 3D VLE?

1.5 Research Approach

To achieve the aims and objectives that are discussed earlier and answer the formulated research questions, we followed the following research approach.

Basically, our research approach was based on the Design Science Research Methodology (DSRM) (Peffers et al., 2007). Following DSRM, our approach includes six steps: 1) problem identification and motivation, 2) delineate the objectives of a solution, 3) design and development of the solution, 4) demonstration, 5) evaluation, and 6) communication. Furthermore, an iterative method to satisfy the ultimate objectives and goals was used.

The *problem identification and motivation* was realized after an explorative study. In other words, after reviewing related literature work, some important aspects in the context of adaptive 3D VLEs that still need more investigation were revealed. These issues are already discussed in section 1.2 which presents the problem statement and the motivation of our research.

After that, *the objectives of a solution* were established by first understanding the current attempts to provide adaptation for 3D VLEs and determining how adaptation mechanisms are incorporated in such environments. This step was taken to identify the current limitations that exist in the proposed approaches and frameworks in order to identify the possible improvements and objectives for a solution.

This step involved a literature review, but the objectives of a solution are also based on our work done in the context of an EU-project GRAPPLE[8], which aimed at providing a technological enhanced learning environment (TEL) that adapts automatically to the learner's knowledge, skills, preferences, etc. The GRAPPLE project was mainly oriented towards classical learning material; however our role in the project was to extend the approaches and tools developed for the classical learning material towards the use of 3D content. For this purpose, we developed an authoring tool and a delivery environment for adaptive 3D VLEs. Both tools were an extension of the tools developed for the classical learning materials. In the context of GRAPPLE, we also performed evaluations of the tools developed. Two studies were conducted in order to determine if the proposed solutions fulfilled the requirements. The first study was related to the authoring tool. The evaluation results were quite positive. However, participants revealed some usability problems such as the need for more user feedback during the authoring process and difficult terminology. More details about the evaluation methodology and results are presented in (Ewais & De Troyer, 2014). The second study was related to the delivery environment. The usability and acceptability results were largely inclined to 'good to perfect'. However, participants also reported some drawbacks like the frustration or hardships in navigation; the study also revealed the need for more research on how to combine 3D and text-based adaptive learning. More details about the evaluation methodology and results are published in (Ewais & De

[8] http://grapple.win.tue.nl/home.html

Troyer, 2013b). Based on the results from this evaluation, we decided that different aspects, especially for the authoring, had to be reconsidered resulting in the work presented in this research work.

For the research step *design and development of a solution* five sub-research steps were conducted as follow:

Firstly, a research step was conducted to re-examine and compare the different adaptation techniques that are proposed in different contexts. Our objective was to explore a wide range of adaptation mechanisms that are applicable for 3D VEs without considering a specific context. Furthermore, this provides us with insight knowledge about 3D VLE anatomy and components that are applicable for adaptation. The result of this research step is a list of adaptation techniques for 3D VLE components with a high level of abstraction. This research step answers research question RQ1.

Secondly, a number of research domains such as adaptive hypermedia and adaptive 3D games and 3D VE domains have been investigated thoroughly. As a result, a useful set of conceptual models have been extracted to allow describing adaptation aspects along with pedagogical bases and shielding the authors from the complexities of the authoring process and the technologies.

Thirdly, we investigated work related to design requirements that are proposed in educational video games and educational 3D VE. As a result of this research step, functional and usability requirements were derived for enabling 3D novice educators to design adaptive 3D VLEs.

Fourthly, we constructed a conceptual framework for adaptive 3D VLEs. This step builds on the results of the previous three sub-research steps: defining a list of adaptation techniques, identifying conceptual models, and determining functional requirements and usability requirements. This answers research question RQ 2.1.

Fifthly, after consulting different work proposing visual languages to design adaptive 3D educational games (presented in section 3.2.2.3), we opt to have visual modelling languages to support authors in designing adaptive 3D VLE. In particular, the modelling languages are aimed to support authors in defining pedagogical bases, specifying adaptive storylines, and applying adaptation inside 3D VLE. This answers research question RQ 2.3.

Finally, an authoring approach has been proposed as well as a prototype for this authoring approach. This step answers research question RQ2.2.

As far as a *demonstration* step is concerned, we have used the proposed approach to author a simple adaptive 3D course on the solar system.

For the *evaluation* step, a quantitative as well as a qualitative evaluation method was chosen. This step focussed on the evaluation of the visual languages. Good feedback related to both usability and acceptability aspects was received in the quantitative user evaluation. To seek for explanations about the less positive aspects, we also performed a number of user interviews.

The *communication* step was realized by being involved in GRAPPLE project in which our initial research steps has been carried out. Additionally, we have also communicated and published the results in international journals and conferences (Ewais & De Troyer, 2013a, 2013b, 2014; De Troyer et al., 2009, 2010). By that, we had the opportunity to meet peers and share and discuss the current hot topics in this research context.

1.6 Thesis Outline

This dissertation is organized as follows:

Chapter 1: gives an overview of the general context of our work, the research problems that we want to tackle, as well as our motivation and ultimate objectives, and the research approach taken.

Chapter 2: provides general background. We discuss different applications using 3D Virtual Environments, and particularly web applications and game-based applications. We also discuss 3D Virtual Environments that are built in the context of e-learning. As our approach can be considered as an extension of work done in the context of adaptive hypermedia, this chapter also presents an overview of different approaches in the context of adaptive hypertext and hypermedia.

Chapter 3: reviews related work dealing with authoring adaptivity in 3D Virtual Learning Environments, as well as in video games. It starts with an overview of proposed adaptation techniques used in educational video games and 3D Virtual Learning Environments. Then, an overview of existing authoring tools, which are used to create adaptive hypertext and hypermedia applications, is presented. It continues with authoring approaches that are developed to create adaptation in video games and 3D Virtual Learning Environments.

Chapter 4: starts with discussing the different parts that can be adapted inside the 3D VLE. Next, a systematic overview of possible adaptation techniques that can be applied to the 3D VLE components is proposed. A list of adaptation techniques, so-called adaptation types and strategies, are presented in this chapter.

Chapter 5: presents goals and requirements for specifying adaptive 3D VLEs. Furthermore, a conceptual modelling framework for adaptive 3D VLEs is presented to describe different components required for performing adaptations inside the 3D VLE.

Chapter 6: describes our authoring approach for specifying adaptive 3D VLE. It first provides some important features which should be supported by an authoring tool in order to create adaptive 3D VLE effectively and easily (i.e. the requirements). Before we discuss the visual modelling languages that we propose, the chapter reviews important guidelines and principles related to visual modelling languages in general. After that, the chapter introduced the pedagogical model language, the adaptive storyline language and the adaptive topic language, which together allow specifying an adaptive 3D VLE. For each of the visual modelling languages, their visual notations and structure are given along with illustrating examples.

Chapter 7: demonstrates the visual languages through an example course about the solar system. The chapter also describes shortly the actual solar system course that corresponds with the specifications for the course given by means of the visual languages.

Chapter 8: focuses on the conducted user evaluations for the proposed visual modelling languages. This was done to validate the usability and the effectiveness of the visual languages (pedagogical model language, adaptive storyline model language and adaptive topic model language). Evaluation principles and data analysis are discussed together with demographic data about the participants. Usability, acceptance, and qualitative feedback results are discussed thoroughly.

Chapter 9: discusses a prototype authoring tool for the proposed approach. Sketches of the graphical user interface for the different parts of the authoring tool are provided. Also some technical specifications are given. In addition, the chapter describes the associated process flow for designing an adaptive 3D VLE.

Chapter 10: provides a summary of the research results obtained in the context of this dissertation. Contributions are discussed, as well as limitations of the work. Further research directions are presented.

Chapter 2: Background

This chapter deals with the general background related to this dissertation. It provides an overview of the potential of classical (i.e. non-adaptive) 3D Virtual Environments (3D VEs) in different domains. Moreover, it also briefly explores some existing web-based 3D VEs in the context of education; pure educational applications as well as and serious games are presented. The chapter also describes the technology available for realizing 3D web applications. Next, it presents adaptive hypermedia, which can be considered as the base for adaptive 3D Virtual Environments. Principles of adaptation techniques and mechanisms in adaptive hypertext and hypermedia are reviewed. This will give insight in how adaptation could be provided in 3D Virtual Environments.

The chapter is structured as follows. Section 2.1 reviews a number of 3D web based Virtual Environments in the context of learning, collaborative learning, training, as well as virtual environments for social and medical purposes. 3D web-based technology for implementing web-based 3D virtual environments is discussed in Section 2.2. Section 2.3 presents adaptive hypermedia principles and adaptation criteria. Moreover, it explains the adaptation taxonomy used in Adaptive Hypermedia: adaptation for navigation and adaptation for content presentation. Section 2.4 summarizes what has been presented in this chapter.

2.1 Web Based 3D Virtual Environments

This section presents aspects related to using 3D and VR in different settings, and illustrates that 3D and VR is widely used in different fields. The reviewed work is limited to work developed to deliver 3D virtual environment through the Internet by means of web-based 3D applications.

In the first subsection, we review some existing work in using 3D environments for different purposes and in different domains like tourism, educational training, and e-learning. The purpose of this overview is to illustrate the advantages and potential of 3D virtual environments. The next subsection will review the use of 3D web-based games for educational and training purposes, as this is also relevant for our work.

2.1.1 Existing 3D Web-Based Virtual Environments

In this section, we present different 3D Virtual Environments developed as web applications in several domains like tourism, educational training, medical, e-learning and social contexts.

In the tourism domain different applications have been developed to enable users to explore 3D virtual cities and 3D virtual museums. The advantages of such applications can be illustrated by the following examples:

- Avatars are used to guide the users toward important sights and interesting places to be visited like in (Outline3D, 2000; Udine3D, 2011).
- Visitors (users) are able to interact and chat with each other inside the 3D virtual museum, city. For instance, Khufu pyramid[9] and Leonardo's Ideal City in (Barbieri & Paolini, 2001) are developed to enable visitors to interact with each other while they are visiting different sights of Khufu pyramid or Leonardo city. This adds a social dimension to the virtual environments.

Also in the maintenance and training domain, several 3D applications and approaches are provided. Some of the important advantages of 3D models and applications in this domain are the following:

- They provide rich user interaction techniques and supporting the feeling of presence inside the 3D virtual world (Sastry & Boyd, 1998; Slater, 1999).
- Avatars are used to train users in performing tasks and to simulate performing different activities like in (Yusoff et al., 2011; Schneider et al., 2011).

The medical educational field is also another important domain for 3D VE. Some of the factors that influence the success of 3D VEs in the context of medical domain are the following:

- Increased learning perception for both patients and doctors is realized using 3D technology. Interesting examples are provided in (Lu et al., 2005; Korocsec et al., 2005; McDerby et al., 2001) which aimed at supporting learners with immersive environments to visualize different medical concepts.
- Events (virtual meetings) are supported for users inside the 3D virtual 'meeting' rooms (Wiecha et al., 2010). Such an event can have different activities like a guest speaker session (see Figure 2.1), and users are

[9] http://www.3ds.com/khufu

able to share information and resources during the event with high level of social relations.

Figure 2.1: Guest speaker presentation (taken from (Wiecha et al., 2010))

As far as the educational and e-learning domain is concerned, researchers revealed different advantages for using 3D Virtual Learning Environments (3D VLE). Some of the advantages are the following:

- Supporting different educational activities such as uploading grades and assessments and supporting communication between the lecturers (tutors) and the learners with chat techniques are considered as an advantage of 3D VLE (Jenkins et al, 2005).
- Combining 3D VLE with hypertext application is considered as a credit since such applications enable the students (learners) to use the textual materials along with being able to view 3D models corresponding to related topics and concepts. For instance, (De Byl & Taylor, 2007) tried to integrate web 3D technologies for educational purposes as their main goal for the AliveX3D[10] platform. AliveX3D allows students to interact with 3D models which are related to the different concepts by using dollylinks. Dollylinks are used to hyperlink text into its corresponding 3D models. Figure 2.2 depicts the integration of text contents with a 3D model that shows the solar system.

[10] http://www.alivex3d.org/

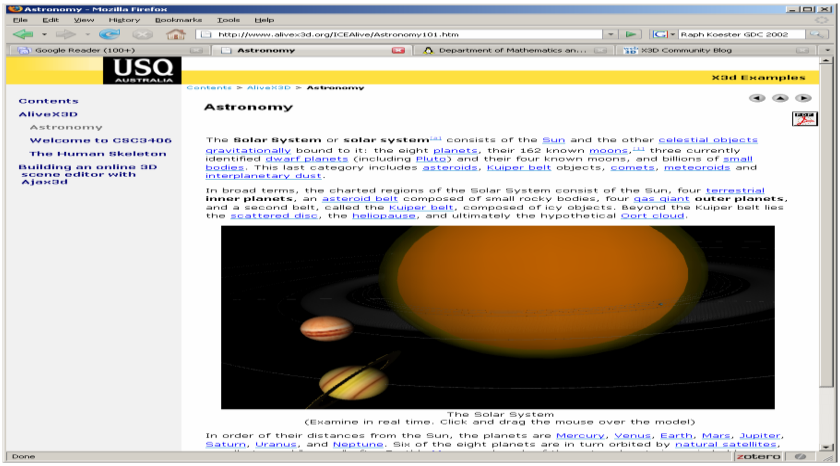


Figure 2.2: AliveX3D project (taken from (De Byl & Taylor, 2007))

- Another interesting example is given in (Livingstone & Kemp, 2008). They introduced the open source Sloodle[11] environment, which stands for Second Life Object Oriented Dynamic Learning Environment. The project aimed to integrate SecondLife[12] (a 3D Virtual Environment) with Moodle[13] (a Course Management System). Some of the important features in any course management system like blogging, classroom gesture, chat, voice, and video calls are also supported in Sloodle. Similarly, integrating Secondlife and ActiveWorld[14] with Facebook[15] has been investigated in (Cai et al., 2009). The authors proposed an approach to enable student teams to develop the 3D virtual world integrated with the social communication applications.

2.1.2 3D Web-Based Games

This section, we focus on the use of 3D web-based games for educational and training purposes.

Recently, integrating 3D games and learning has received more and more attention (Eck, 2006; Prensky, 2001). As 3D Games and VR are rather close from a technological point of view, we believe it is relevant to also consider the work done in the context of 3D games for educational purposes. Next, some examples of using 3D games for educational purposes are presented.

[11] http://www.sloodle.org/
[12] http://secondlife.com/
[13] http://moodle.org/
[14] http://www.activeworlds.com/
[15] https://www.facebook.com/

Starting with medical domain, Durkin, in (Durkin, 2010), argued that video games has different benefits like increasing the medical information and training learners to acquire surgical skills. As a result, different work has been conducted to integrate serious games in the context of the medical domain. For instance, the authors in (Kato et al., 2008; Green & Bavelier, 2003; Ferguson et al., 2007) proposed approaches for improving health care training purposes. Moreover, other researchers proposed games which were designed for practicing specific surgical skills like gastroscopy in (Schlickum et al., 2009; Enochsson et al., 2004). Other researchers proposed games which were used to help patients in learning how they can do their physical treatment exercises (O'Connor et al., 2000; Hoffman et al., 2004; Vilozni et al., 2001).

Turning to educational 3D games, researchers also investigated their use in the context of collaborative environments. One of the examples in this context is the KM Quest[16] project. It is a web-based simulation game aiming at developing core knowledge management skills with situated problems using collaborative learning. Another example is Virtual U[17] which is a project that aimed at creating a computer simulation of university management in a game form. It is based on the idea of creating a "SimCity" for universities. It has been used across universities on America and demonstrates the use of games for E-learning.

2.2 3D Web Based Technologies

This section gives the reader an overview of the different technologies available to deliver interactive 3D contents through the Internet using a web browser.

Recently, many 3D web technologies (programing languages and software tools) are available to create and deliver 3D virtual environments through a web browser (Chittaro & Ranon, 2007d). 3D web technologies are based on some common aspects like requesting 3D resources form the server through *http* request protocol and displaying the 3D models inside the web browsers using special plugins. Furthermore, both the advances of graphics computing hardware and network bandwidth make it possible to deliver 3D virtual environments through the Internet. 3D technologies provide different features to make it easy to integrate different 3D models with other contents in the same web page. However, full integration of text, images, videos and audio can also happen inside the 3D VE. This can be achieved by using the different features provided by 3D modelling languages such as Virtual Reality Modelling Language (VRML), eXtensible 3D (X3D), and Java 3D. We describe these technologies briefly in the next subsections.

[16] http://www.kmquest.net
[17] http://www.virtual-u.org/

Consequently, most of the 3D contents can be experienced through different web browsers like *Internet Explorer, Safari, FireFox,* etc. This is achieved by either using web VR-plugins such as *Vivaty*[18], *BScontact*[19] or by the recent HTML5[20] technology which enable the users to experience 3D models without the need for VR-plugins.

2.2.1 Virtual Reality Modelling Language (VRLM)

Virtual **R**eality **M**odelling **L**anguage (VRML) is a Web3D Consortium[21] open standard. The VRML specifications originated back from 1994. Later, as a result of modifications which tackled different shortcomings like real time animation and interaction in previous version of VRML, a new version of VRML called VRML97 was published in 1997 (Hartman & Wernecke, 1998). It was adopted as the first ISO standard for 3D modelling languages. The new version came with much more powerful features which allow programmers to develop web based interactive 3D contents with different properties and functionalities.

Using VRML, programmers are able to define a scene in which all the 3D objects are allocated. They are also able to create 3D objects which have different material characteristics, appearance properties and texture mapping. In addition, programmers are able to integrate sound and video files inside the 3D virtual world. They are also able to define navigation paths and viewpoints to be followed by the users. Moreover, different interaction techniques with the created 3D virtual objects can be specified in VRML. As a result, the user will be able to interact with the virtual world like drag and drop objects, open doors, turn on/off virtual machines. Besides, programmers are able to define simple and/or complex behaviours for 3D objects like rotation, moving, walking, driving a car or airplane. For more details about the VRML grammar, we refer to (ISO/IEC-14772, 1996; Brutzman, 1998).

Any VRML 3D model file is considered as a hierarchal graph scene which should include a main node called scene node (see Figure 2.3 VRML Scene Graph). The scene node has different properties fields like lights, sounds, navigation techniques and viewpoints. Lighting nodes have different attributes. Basic ones are intensity, which specifies the brightness of the light source, and colour, which is specified by RGB mechanism. Furthermore, sound is represented by *AudioClip* nodes, which has loop, start time, and stop time attributes to be altered. Also, navigation techniques are defined by specifying one of the navigation types like "FLY", “WALK", "EXAMINE", and "NONE". Moreover, VRML provides a NavigationInfo, which contains attributes for avatar size, speed, and navigation type. *ViewPoint* node, on the other hand, is

[18] http://vivaty.wordpress.com/
[19] http://www.bitmanagement.com/products/interactive-3d-clients/bs-contact
[20] http://www.w3.org/html/wg/drafts/html/master/
[21] http://www.web3d.org

used to support users with different positions where they can view the general scene or specific 3D object. Both position and orientation are the basic attributes for viewpoint nodes.

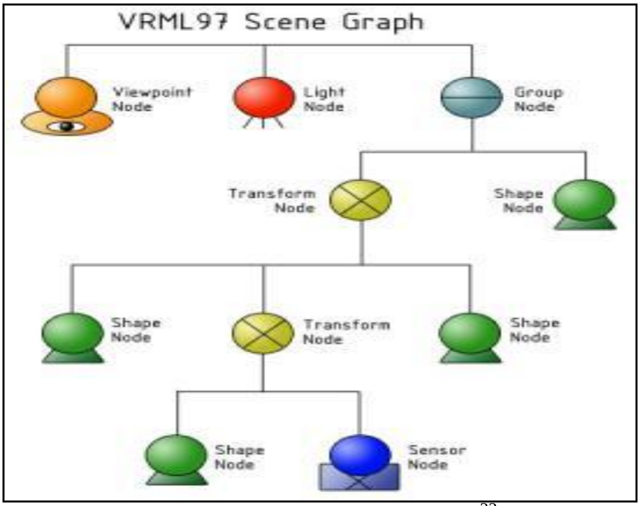
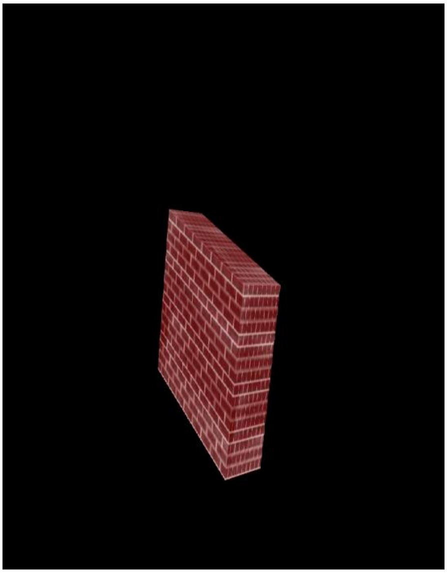


Figure 2.3 VRML Scene Graph[22]

Every 3D virtual object is defined by using what they call a *Shape* node (see Figure 2.4). Every Shape node has different properties represented by two child nodes called *Geometry* and *Appearance*. A geometry node has different types like Box, Text, or Cylinder. It has also different fields representing different primitive parameters for 3D object like size, position and orientation. An appearance node includes attributes for specifying colour, transparency and texture mapping. Optionally, grouping nodes can be used like *Group* and *Transform* to create a relationship between 3D objects which has the same shapes properties.

[22] http://thana.org/GISVisualization/html/finalReport.html



```
#VRML V2.0 utf8
#brickWall.wrl
#andy Harris

# demonstrates use of textures

Shape {
 appearance DEF theTexture
Appearance {
   texture ImageTexture {
    url ["brick1.jpg"]
   } # end texture
  } # end appearance

  geometry Box {
   size 1 2 3
  } # end geometry
```

Figure 2.4: VRML example[23]

Regarding user interaction, interaction techniques are represented by adopting a number of sensors for a 3D object. There are different sensors nodes related to time and user interactions. For example, a *TimeSensor* node could be used for triggering events in specific time stamp. This sensor type has different properties like time to start and time to end. Loop attribute could be used for continuous simulation or animation. Another interesting sensor type is called *ProximitySensor,* which is responsible for triggering events when the user's position is detected inside the proximity area of the 3D object. *TouchSensor* is another sensor, which is needed to trigger events when the user clicks on the 3D object. All provided sensor types are helpful to define the different interaction techniques between the user and the 3D virtual objects inside the virtual world. The triggered event caused by the sensor nodes is defined by so-called ROUTE, which is responsible for passing events from one node to another.

Defining behaviours for 3D objects could be implemented by using a *Script* node which could be used either for using VRML scripts or integrating code in other languages like Java and JavaScript which can change 3D object properties or behaviours using specific script.

Generally, VRML is considered as a high expressive and powerful interactive 3D format which is supported by quite a number of tools and API (Lansdale, 2002; Brutzman, 1998). There are a number of authoring tools

[23] http://www.cs.iupui.edu/~aharris/mm/texture/texture.html

developed to help programmers for visualizing VRML source code. One of such tools is the VIM editor[24], which is developed to be used on different platforms like UNIX, MS-Windows, and Macintosh. Another editing tool is Viper[25], which is a Java-based editor. VrmlPad[26] is a commercial tool for visualizing VRML files.

In summary, VRML has some advantages over other 3D web technologies like viewing high quality of visual representation for the 3D models with standard hardware requirements for computers. Also programmers are able to define objects' behaviours. However, only well skilled programmers in VRML are able to understand its syntax so it is considered as not intuitive (Ross et al., 2003). Furthermore, 3D models coded in VRML need a wide bandwidth.

2.2.2 eXtensible 3D Modelling Language (X3D)

Extensible 3D (X3D) is an open standard XML-based format which is used to model interactive 3D materials. X3D is an ISO standard and the successor of VRML. Therefore, it inherits most of the modelling features and structure format of VRML. The X3D specifications[27] were published in 2004 by the Web3D Consortium and International Organization of Standards (ISO).

X3D comes with improvements and modifications. For example, based on the scene graph structure for X3D files, new nodes are invented to support advanced and multi texture mapping; improve sensor nodes, and providing more powerful dynamic behaviour mechanisms. Besides, X3D supports three encoding: VRML encoding. XML based encoding, and binary encoding to optimize file size and loading. XML based encoding is considered as a very powerful feature which is adopted to integrate 3D contents with standard materials in web-based applications (see Figure 2.5: Geometric nodes in X3D). Another important feature is that X3D includes the Scene Authoring Interface (SAI), which is used to define the communication between external applications and Scene nodes in X3D files.

Different authoring tools are developed for creating X3D files such as X3D Edit[28], BS Editor[29], SwirlX3D Editor[30], Flux Studio[31], etc.

[24] http://www.vim.org/about.php
[25] http://xsun.sdct.itl.nist.gov/~mkass/Viper/index.html
[26] http://www.parallelgraphics.com/products/vrmlpad
[27] http://www.web3d.org/x3d/specifications/
[28] www.web3d.org/products/detail/x3d-edit/
[29] http://www.bitmanagement.de/products/bs_editor.en.html
[30] http://www.pinecoast.com/swirl3d.htm
[31] http://mediamachines.wordpress.com/flux-player-and-flux-studio/

However, (Liu & Gun, 2006) reported that X3D has some disadvantages in term of difficulty in updating and modifying the 3D models and the high cost of creating complex X3D models.

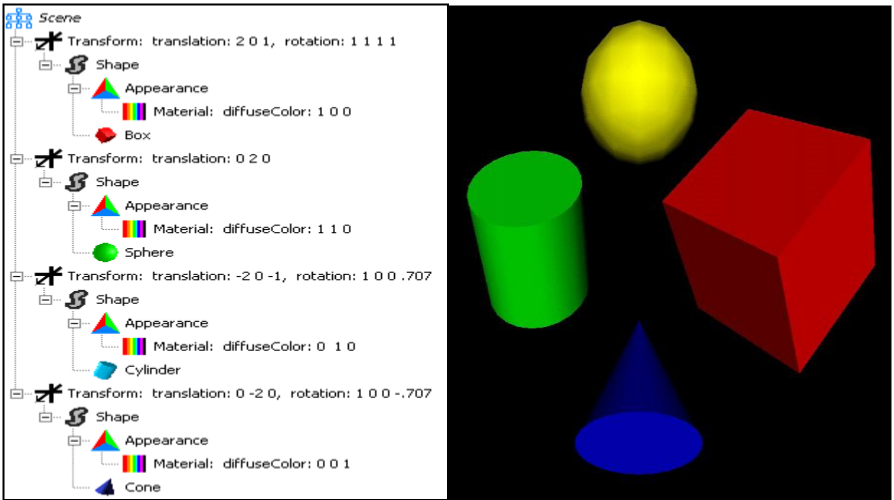


Figure 2.5: Geometric nodes in X3D[32]

2.2.3 Other 3D Web modelling languages

This section presents some other interesting modelling languages to create web-based 3D virtual worlds.

First, *Java3D* is a Java programming language with extended capabilities and functionalities to create 3D models. Java 3D API provides different interfaces and classes to render and program 3D objects with defining their behaviour if needed. More details about Java 3D programming library can be found in (Palmer, 2001). One of the important advantages of creating 3D models using Java3D is the platform independency and programmers are able to design 3D contents which deal with complex problems. However, delivering resulted 3D contents is complex (Ross et al., 2003).

Secondly, *OpenInventor*[33] is another 3D programming language which was originally developed by Silicon Graphics International Corp. (SGI)[34]. It is an object oriented 3D graphics toolkit for the C++ programming language. Its 3D functionalities are integrated with OpenGL[35] graphics library.

[32] http://x3dgraphics.com/slidesets/X3dForWebAuthors/TutorialX3dSceneGraph
[33] http://www.sgi.com/products/software/
[34] http://www.sgi.com/
[35] http://www.opengl.org/

Thirdly, 3DMLW[36] (3D Markup Language for Web) is a markup language (XML based format), developed by 3D Technologies R&D. It is used for creating 3D and 2D contents on the Web. The interactive 3D models are represented by XML syntax. A scripting language, so called Lua, is used to define the interaction functionalities. 3DMLW is considered as easy to understand and to combine 3D and 2D materials. However, it is only supported under the Windows platform and does not integrate any video file formats.

Fourthly, XML3D[37] is also another markup language and was developed by Computer Graphics Group of the Saarland University. It is considered as an extension for XML to support 3D modelling. One of the most important features in XML3D is the fact that there is no need for an external plugin as the resulted 3D models can be integrated with (X)HTML page contents directly. However, one of its disadvantages is that it is not standardized yet from W3C. Furthermore, as it is relatively new, so some bugs can still show up.

Fifthly, asynchronous Javascript and XML (AJAX) (Garrett, 2005) is a technology which enable clients to download or upload new web page contents from or to server without reloading the whole web page. Figure 2.6: Class web application model VS Ajax web application model (taken from (Garrett, 2005))depicts the difference between the classic web application and the Ajax web application.

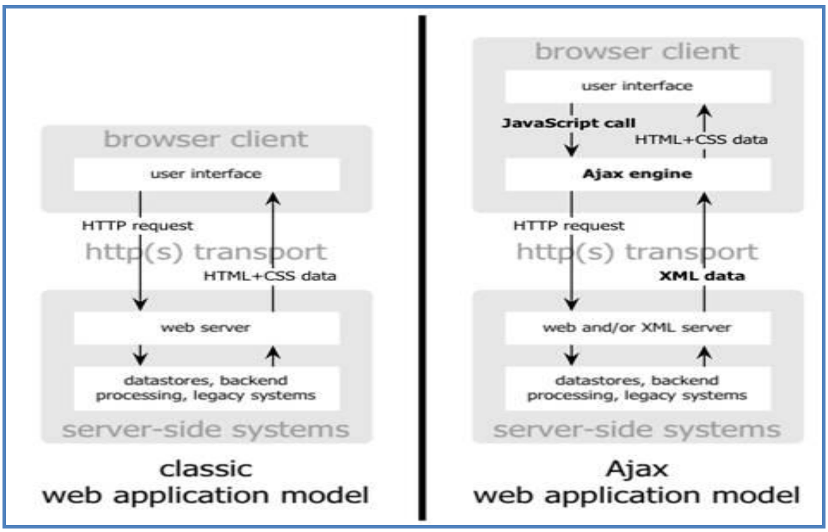


Figure 2.6: Class web application model VS Ajax web application model (taken from (Garrett, 2005))

[36] http://www.3dtech-rd.net/3dmlw
[37] http://www.xml3d.org/

AJAX3D is built on top of the technologies AJAX and X3D. AJAX3D is invented by the Media Machine [38] and comes up with several advantages. One of the advantages of using AJAX3D is that it uses Ajax's XMLHttpRequest object to communicate with the X3D Scene Access Interface (SAI). Scene Authoring Interface API provides a mechanism to be able to change the 3D models (3D virtual world) on the fly without having to reload the web page. The SAI allows access (through JavaScript) to the 3D scene, this means you can send and retrieve JavaScript objects to the scene in order to retrieve information from it or update it directly.

Finally, there is also HTML5[39], which is the latest evolution of HTML developed under the supervision of the World Wide Web Consortium (W3C)[40]. Among the new HTML5 features, there is the Canvas, which is used to draw graphics, on the fly, on a web page, and SVG, which stands for Scalable Vector Graphics and is used to embed 2D graphics; both features are quite widely supported and actively used in most of the web browsers. HTML5 introduces different application programming interfaces (API) like WebGL[41] for the creation of different web application and 3D web applications. More details about integration of 3D applications and HTML5 can be found in (Yang & Zhang, 2010).

2.3 Adaptive Hypermedia

In this section, we give a general overview of adaptive hypermedia since we consider this domain as the base for applying adaptation to 3D virtual environments.

The term hypertext refers to different contents (text format) that are connected to each other in a network structure. The hypermedia term was given to hypertext as a generalized term (McKnight & Richardson, 1991; Seaman, 1993; Gygi, 1990). Different definitions are available for both hypertext and hypermedia. For example Nelson defines hypertext, in (Nelson, 1981), as "*I mean non-sequential writing - text that branches and allows choices to the reader,. . . this is a series of text chunks connected by links which offer the reader different pathways . . .*" and hypermedia is defined by "*Hypermedia simply extends the notion of the text in hypertext by including visual information, sound animation and other forms of data . . .*". The main hypermedia's characteristic is that the linkage between the nodes (pieces of content) is nonlinear.

The way that information is organized on the web is based on the principles of hypermedia. However, with the wide growth of available

38 http://mediamachines.wordpress.com/
39 http://www.w3.org/TR/2011/WD-html5-20110525/
40 http://www.w3.org
41 http://www.khronos.org/webgl/

information on the web, researchers started realizing that it can become difficult for users to find what they are looking for. In additions, users become impatient and will leave a website if they are not able to find what they are looking for quickly. Researcher stated that "one size fits all" is not sufficient. This has given rise to the idea of proposing hypermedia systems which adapt to the user needs, like providing different navigations (Brusilovsky et al., 1998b).

In the next subsection, we discuss the difference between the concepts *adaptivity* and *adaptability,* as both concepts are different. Next, we present the adaptation criteria, principles of adaptive hypermedia systems, and their use in the educational domain.

2.3.1 Adaptivity and Adaptability

Adaptation is defined as any action that adapts the information or services provided by a website to the needs of a particular user or a set of users, taking advantage of the knowledge gained from the users' navigational behaviour and individual interests, in combination with the content and the structure of the hypermedia system. A rigorous definition of an adaptive hypermedia system was given by Brusilovsky who defines adaptive hypermedia systems as "*By adaptive hypermedia systems we mean all hypertext and hypermedia systems which reflect some features of the user in the user model and apply this model to adapt various visible aspects of the system to the user. In other words, the system should satisfy three criteria: it should be a hypertext or hypermedia system, it should have a user model, and it should be able to adapt the hypermedia using this model"* (Brusilovsky, 1996).

It is also important to distinguish between *adaptive* hypermedia and *adaptable* hypermedia. Adaptive hypermedia is meant to provide dynamic changes applied in the system depending on user profile. Authors in (Frasincar & Houben, 2002) defines adaptivity as follow: "*Adaptivity (or dynamic adaptation) is the kind of adaptation included in the generated adaptive hypermedia presentation, i.e. the generated hypermedia presentation changes while being browsed*". The same authors define adaptability as "*Adaptability (or static adaptation) means that the generation process is based on available information that describes the situation in which the user will use the generated presentation*" (Frasincar & Houben, 2002).

In this thesis, we aim for adaptivity, i.e. we aim for 3D VLEs that dynamically adapt to the actions and behaviour of the user. We refer the reader to (Oppermann et al., 1997; Hauger & Kock, 2007; Cristea, 2004) for more details about the different features that exist in adaptable and adaptive learning systems.

2.3.2 Different Models for the Adaptation Criteria

Adaptive hypermedia uses some models to perform proper adaptation to the hypermedia system's contents and navigation. We provide a short explanation of the different models.

First, adaptation can be done according to the specified data in the *user model* (also called *learner model* in the context of educational applications). Such a model is needed to include all the required data for adaptation like, in the context of learning, the learners' prior knowledge and skills, learning preferences or styles, learning capabilities, performance level and knowledge state, interests, personal circumstances (location) and motivation (Kareal & Klema, 2006).

The authors in (Triantafillou et al., 2006) reviewed different work related to providing adaptation in hypermedia systems and summarizes the different characteristics that could be used to formulate the user model. A literature review of standards and methodology that are used to form the user model in the past decades is discussed in (Martins et al., 2008).

Secondly, another important model used in adaptive hypermedia is the *domain model,* which is responsible for describing the content by means of different concepts that are linked to each other by means of relations. Most of the provided adaptive hypermedia systems are concept-based (Aroyo et al., 2006b). In other words, to structure the information, a concept model, domain model, or content model is used. Based on this model, different content can be selected in different situations.

Thirdly, the *adaptation model* is also one of the key models which is used to specify what kind of adaptation techniques should be used on which contents and when the defined adaptation techniques should be triggered. To achieve this, triggering adaptation rules are used. Adaptation rules can be defined based on concepts relations which are defined in the domain model (Brusilovsky et al., 1998a). Another approach is adopted in (De Bra & Calvi, 1998); the approach is used to bound adaptation rules to concepts or group of concepts.

In principle, the user model, domain model, and adaptation model are the typical models that are used in the context of adaptive hypermedia (Paramythis & Loidl-Reisinger, 2003).

2.3.3 Principles of Adaptive Hypermedia

Now, we turn to what can be adapted in hypermedia systems. According to Brusilovsky, adaptation can be applied for hypermedia systems in two ways (see Figure 2.7). Firstly, adaptation could be applied to the presentation of content, which will be adapted to the user's model data like knowledge and experience level. For example, a qualified user can be provided with more detailed and deep information, while a novice can receive additional explanations. Secondly, orientation and navigation support will be helpful for the user to locate the required information. The adaptive navigation support will depend on the user goals and objectives. Thereby, providing such an adaptation technique guarantees that the users will not be in the risk of losing their orientation and focus.

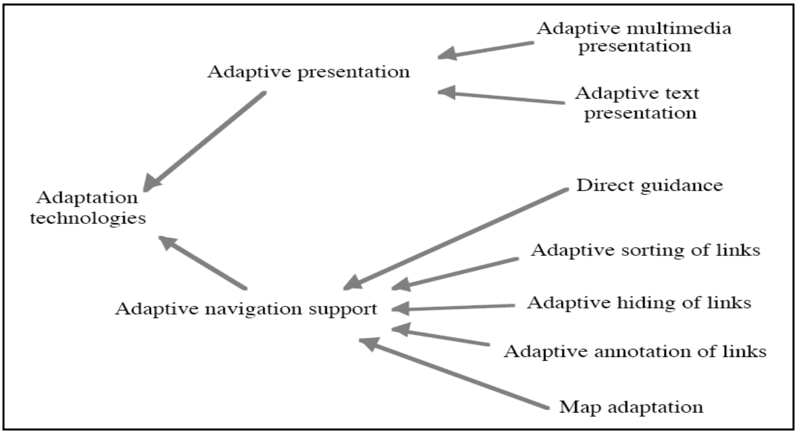


Figure 2.7: Adaptation Techniques (taken from (Brusilovsky, 1996))

We now review the techniques that are proposed to provide adaptation to the contents and navigation in adaptive hypermedia. Concerning adaptive content presentation, there are different techniques to provide suitable materials (like text, images, video) according to the user model (current users' knowledge, background, preferences, goal, and experience). For example, users with low knowledge about a concept could get texts with few details. Brusilovsky (Brusilovsky, 1996) suggests adaptive hypermedia techniques which are concerned with content presentation to contain additional explanation, pre-requisite explanations, comparative explanations or explanation variants. The following methods and techniques have been developed for content to explain a specific concept (Brusilovsky, 1996):

- *Additional explanations*: some group of users are given additional information to read. This additional information is hidden/displayed depending on the current values of specific attributes in the user model. Additional explanation is implemented by using conditional text, stretch text, and frame-based techniques.
- *Prerequisite explanation*: the adaptive hypermedia system automatically inserts explanations of prerequisite concepts that the user is not familiar with. It will be done before presenting a current topic. It is implemented by using conditional text, stretch text techniques, and frame-based techniques.

- *Comparative explanation*: the adaptive hypermedia system gives user similarities and differences between current topic and the related one. Similar to above methods comparative explanations is implemented by using conditional text, stretch text techniques, and frame-based techniques.
- *Explanation variant*: This technique provides users with different explanation about the same topic depending on their current knowledge on the concept. This can be implemented by conditional text, fragment variant, page variant, and frame-based techniques.

The second group of techniques is meant to be used for providing adaptive navigation support. The structure of adaptive navigation determines the level of guidance and freedom granted to the user within the system. Adaptive navigation support can be provided in the form of direct guidance, hiding, sorting, and annotation (Brusilovsky, 1996):

- *Direct Guidance*: The system suggests to the users which link to follow next from the given nodes while browsing, depending on their user model.
- *Hiding*: Adaptive hiding is the most common technique for adaptive navigation support. Links are adaptively hidden so that users cannot be exposed to some information until the user model decides that the user is ready for that information. If the user model decides that a link is relevant, then that link is visible. If the system considers the information irrelevant, the link will be hidden.
- *Sorting*: Adaptive sorting order the links on a page according to the user model so that the most relevant links for the user are presented first and the least relevant are presented last.
- *Annotation*: Adaptive annotation technique uses textual or verbal indications to suggest to the user which links to visit in which order. Links may have different icons, be colour-coded, or have different font sizes or font types to distinguish between types of links. The annotation used is adaptively changed based on the user model.

We explained Brusilovsky's adaptation techniques and methods as they are considered one of the major mechanisms in the domain of adaptive hypermedia in general. They clearly indicate what can be adapted in hypermedia systems and what kind of adaptation methods could be performed. For more details about the different adaptation methods and the techniques that are proposed to adapt the hypermedia systems we refer to (Knutov et al., 2011; Conlan, 2003; Brusilovsky, 2004; Bunt et al., 2007; Cristea & Ghali, 2011).

2.3.4 Adaptive Educational Hypermedia Systems

Adaptive hypermedia systems are used in different domains. One common example is the educational domain where they are called Adaptive Educational Hypermedia Systems (AEHS). They use the above techniques (adaptive presentation and adaptive navigation support) to deliver adaptive courses in the context of E-learning. The adaptive courses compose student's profiles explicitly by quizzes and exams or implicitly by monitoring student's navigation and activities through the hyperspace of the course.

There are number of examples of adaptive hypermedia applications. A first example is KN-AHS (Kobsa et al., 1994) that is developed to provide adaptive hypertext according to current learner knowledge. Another example is Interbook (Brusilovsky et al., 1998a) which is used to provide adaptive electronic textbooks which transform plain text to specific HTML files. It supports the learner with adaptive navigation support, adaptive help presentations, and adaptive guidance. Another interesting application is called MetaDoc (Boyle & Encarnacion, 1994) that introduces an adaptive presentation technique based on stretch text which is used to collapse all non-relevant text and un-collapse all relevant text. Similarly, My online Teacher (MOT) (Cristea & De Mooij, 2003) was also developed to support educators in developing adaptive hypermedia courses. MOT supports delivering courses with adaptation techniques based on Brusilovsky's taxonomy (see previous section). Furthermore, Adaptive hypermedia for All! (AHA!) (De Bra et al., 2003) is an interesting architecture which is used to provide adaptive techniques like adaptive content presentation based on different relevant fragments as well as link hiding and annotation. Furthermore, it enables teachers to develop their own adaptive courses by means of an authoring functionality (details about the AHA! authoring tool will be discussed in the next chapter).

More explanations about adaptive educational hypermedia systems is discussed in (Mulwa et al., 2010; Hauger & Kock, 2007).

2.4 Summary

This chapter presented the background related to this work.

The chapter started with illustrating advantages of the use of 3D VEs in different domains like tourism, training in medical or social environments, as well as in E-learning and collaborative environments. In addition, the chapter also pointed to the use of 3D educational games in educational settings. Some advantages are for instance: users are able to have the sensation of presence by representing them as avatars inside the 3D VE, users are able to interact with 3D virtual objects and other users inside the same 3D VE. Furthermore, users are able to perform complex tasks like performing "virtual surgery" as in the case of medical training. Furthermore, social aspects can be also realized in 3D

VE by enabling users to meet, chat, perform video calls, visit specific virtual places together, share pictures and materials, etc.

Next, we reviewed technologies to realise 3D web-based virtual environments. The reviewed technologies are Virtual Reality Modelling Language (VRML), eXtensible 3D Modelling Language (X3D), Java3D, OpenInventor, 3DMLW, XML3D, Asynchronous Javascript and XML (AJAX), AJAX3D and HTML5.

We also reviewed the adaptive hypermedia domain. As mentioned earlier, in hypermedia, adaptation can be applied basically to the presentation of content and to the navigation. An important work in this context is the work of Brusilovsky who proposed generic adaptation techniques for both aspects (content presentation and navigation) (Brusilovsky, 1996).

Chapter 3: Related Work

The objective of this chapter is to present related work which has been done in the context of adaptive 3D VEs and more in particular about adaptation techniques and authoring adaptive 3D Virtual Leaning Environments (3D VLEs). Various authoring approaches, methodologies and frameworks proposed in this context are reviewed. We also indicate, were relevant, how the related work has contributed to our work.

This chapter is structured as follows. On the one hand, we consider related work in the context of adaption techniques for web-based 3D VE (section 3.1), in particular, in the context of two domains: 3D educational games and 3D VLE. Moreover, user navigation and interaction are also discussed in both domains. On the other hand, we discuss related work in the context of authoring adaptivity. Section 3.2 reviews authoring tools that are proposed in the context of adaptive e-learning applications. In particular, authoring tools for adaptive hypermedia in e-learning are presented in section 3.2.1. In section 3.2.2, authoring approaches, conceptual models, and visual languages are reviewed in the context of 3D educational games and 3D VLE. Finally, section 3.3 concludes this chapter.

3.1 Adaptation Techniques for 3D VLE

Before we discuss different work developed to author adaptivity in 3D VEs, we investigate what kind of adaptation techniques have been proposed by previous work so far.

Most of the 3D VEs contain predefined characteristics and properties for 3D virtual objects. In general, the material properties such as colour and texture mapping for the 3D scene and the 3D virtual objects remain the same along the lifetime of the 3D VE. Therefore, users will be limited to interact with and navigate through the same scene and 3D virtual objects again and again. As mentioned earlier, especially in large 3D VEs, this can create the risk that users are experiencing difficulties in both navigation and interaction and being less motivated to visit the same 3D VE several times. One of the possible ways to overcome these obstacles is to provide adaptation to the 3D VE according to the user experience, interests, activities, etc.

An extensive body of literature exists on different adaptation techniques to change the 3D VE or 3D game behaviour according to the user profile and the recorded activities inside the 3D virtual world. We focus on the work relevant for this research work.

It is of interest to review also work done in the context of adaptive 3D educational games, as a 3D game world is similar to a 3D VE. Note that different terms are used in literature such as adaptive serious games (Annetta, 2010), and adaptive edutainment games (Dondlinger, 2007). We use the term adaptive 3D educational games further on.

The next sub sections discuss adaptation methods and techniques that have been applied in 3D VE; each section deals with another aspect of the 3D VE. Section 3.1.1 explains the different adaptation techniques and methods that have been used to provide *contents adaptation* in the context of e-commerce, e-learning, and educational games. After that, section 3.1.2 explains adaptation methods that can be applied to *user navigation* in the context of the previous three domains. After that, section 3.1.3 is describing adaptation methods and techniques that are related to *user interaction* also in the previous contexts. Finally, a summary section is presented in section 3.1.4.

3.1.1 Content Adaptation

In the past, there has been a spate of interest in how to adapt 3D contents according to different criteria. For instance, researchers (Chittaro & Ranon, 2000; Lepouras & Vassilakis, 2006) have proposed changing the 3D content presentation according to the user's interest. Moreover, others (Dos Santos & Osório, 2004b) have proposed to monitor user interactions and activities inside the 3D VEs to reconstructed the 3D VE accordingly.

In what follows, we present 3D content adaptation techniques that are used in the context of e-commerce, e-learning and 3D educational games.

E-commerce Context

The goal of the adaptive 3D VE in the context of e-commerce is not only to help users in navigating towards products that they are looking for, which is considered the common adaptation technique in this domain (Chittaro & Ranon, 2007a), but also to change the product's presentation like colour, size, or its spatial order inside the scene dynamically depending on the user's interests and preferences. Figure 3.1: Virtual Showrooms (Virtual Showrooms, 2011) is an example of web based virtual shop.

Figure 3.1: Virtual Showrooms (Virtual Showrooms, 2011)

The first step towards this technique was presented by (Chittaro & Ranon, 2000, 2002a; Lepouras & Vassilakis, 2006) who described, in general, how virtual shops and mall products can be adapted to the customer profile. For instance, the researchers in (Chittaro & Ranon, 2000), proposed an approach which is called *Adaptive Virtual Reality Store* (ADVIRT). The approach suggested having a direct mapping between the user preferences and interests with the related information about 3D virtual contents and the virtual store, like size, style, specific 3D virtual objects. They also claimed that keeping track of user's positions and interactions with the virtual objects could provide more desirable adaptation actions. The work is quite relevant as it provided a number of content adaptations that are also considered in the context of our work.

In addition, (Bonis et al., 2007) proposed 3D contents adaptation like having a common template for 3D virtual rooms which can be selected according to the collected user information (user model) and the semantic relationships between the 3D contents. Although the work is oriented towards virtual exhibitions, it was inspiring in term of providing consistent content adaptation techniques to groups of users.

E-learning Context

Studies in educational 3D VE (or 3D VLE) (Aquino et al., 2005; Dos Santos & Osório, 2004b; Iqbal et al., 2010) have shown some important content adaptation techniques like hiding and semi-displaying, level of details, associated learning materials, and device adaptation. We present some work conducted in this context.

To begin, researchers have proposed hiding contents when the learner is not yet ready to learn about them (Chittaro & Ranon, 2007c, 2007b). Others have suggested to present different complexity level of the 3D virtual objects according to their suitability to the learner's knowledge (Aquino et al., 2005). Likewise, the authors of (Dos Santos & Osório, 2004b) proposed adaptive 3D contents generation according to the learner's interaction. Their work was limited to inserting or deleting relevant resources according to the user interests.

Furthermore, other researchers also proposed adaptation actions which are not only applied to 3D models but also to multimedia contents associated with the 3D models (Estalayo et al., 2004). For instance, voice contents are considered in the adaptation process, as in (Iqbal et al., 2010) who proposed playing sounds according to user knowledge when he comes close to learning object.

The work conducted by (Dachselt et al., 2006) showed an approach that provided mainly content adaptation to the user's device. Their approach suggested 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 described a showcase where the seat capacity of a conference room can be adapted. The presented work is more on adapting the content for large audience, not on personalization.

Based on their previous work on adaptation (Chittaro & Ranon, 2004), Chittaro and Ranon extended their approach toward supporting adaptation in e-learning platforms (Chittaro & Ranon, 2007b). The authors introduced the Adaptive Educational Virtual Environment, which is tailored to the knowledge level of a student and to their preferred style of learning. To achieve adaptivity in the context of the Educational Virtual Environment, they have used the AHA! engine which was originally developed for adaptive hypermedia applications (Chittaro & Ranon, 2008). To achieve contents adaptation, they have associated every concept from the domain model with a proper part of the 3D VE code. For each adaptive code fragment, alternative variants and the conditions under which a variant should be inserted in the resulting code are specified. When a student queries a specific concept, the AHA! engine will use the user model and adaptation model to choose which alternative code fragment to use. This technique was already explored in the context of Web-based hypermedia like optional fragments and altering fragments (Chittaro & Ranon, 2007a).

3D Educational Games Context

In the context of 3D educational games, researchers suggested different content adaptation techniques (Dondlinger, 2007; Hocine & Gouaïch, 2011; Derryberry, 2010; Lopes & Bidarra, 2011b). Basically, the proposed adaptation techniques in this context are toward changing the representation of non-player characters (NPC), inserting or deleting game elements, quests and scenarios, changing virtual scene and sounds (Yannakakis & Togelius, 2011; Charles et al., 2005; Westra et al., 2009; Magerko, 2008). For instance, the authors in (Charles et al., 2005) suggested a framework for applying adaptation based on continues monitoring of the player performance and based on recorded data, the NPCs' characteristics, the virtual environments, player type can be adapted in real time. Also the work presented in (Westra et al., 2009) proposed a game adaptation model that considered three concepts in the adaptation process: trainee, game objectives, and agents. The proposed approach is applicable for complex serious games.

First, adapting Non-Player Character (NPC) is meant to have more challenging behaviour, entertaining, competing against the learner (Magerko, 2008). In addition, researchers in (Linek et al., 2007) investigated the possibility of adapting the personality of the NPC as well as the role of naturalism and colour in the sense of similarity attraction with the player's personality. Other researchers adapt the NPC dialog in learning situations to users skills (Peirce et al., 2008). Such techniques are mainly oriented towards (educational) games.

Secondly, and interesting to consider is the narrative theory explored in the context of 3D educational games. Researchers in (Hulpuș et al., 2010) investigated the possibility of providing personalized sequence of stories to fit the learning plan. During the execution of the story, the learner will experience different situations that are defined to develop his competence. Every story has a story template that accommodates several situations. Another work, presented in (Göbel et al., 2009), described how the game stories need to be generated adaptively depending on a non-dynamic so-called narrative model. More about providing adaptation for interactive and narrative plans can be found in (Thomas & Young, 2007; Hodhod et al., 2009). Basically, narration can also be explored in the context of adaptive 3D VLE. In other words, supporting learners with different topics to be learned can also be realized in adaptive 3D VLEs. This is further explored in this dissertation.

Finally, another important work is (Moreno-Ger et al., 2008d) which presents an approach called <e-Adventure>. The approach facilitates the creation of interactive educational content. Content adaptation is realized using assessment rules which are inspected by the game engine. Furthermore, content adaptation is realized at the beginning of the game based on the student's profile. However, the content is mostly 2D (not true 3D). A new version of the platform is created towards 3D contents <e-adventure3D>[42].

To the same end, the authors of (Lopes & Bidarra, 2011a) present an adaptation methodology to generate adaptive game content based on player performance. Adaptation happens by using a game observer, which is responsible for monitoring the player's behaviour inside the game and mapping it with the game contents. In other words, case-based mappings between content and player experience is used to generate contents adaptation. It is mainly provided for inserting new 3D models inside the game

[42] http://e-adventure3d.e-ucm.es

environments. The work has an interesting methodology for generating adaptation for games, as it is model-based.

3.1.2 Adaptive Navigation Support

An extensive body of literature exists on the adaptation techniques related to user navigation in the 3D VE (Dos Santos & Osório, 2004a, 2004b; Hughes et al., 2002; Dijk et al., 2003; Brusilovsky, 2003). Further on, we will present adaptation techniques for navigation support in the context of e-commerce, e-learning, and 3D educational games.

E-commerce Context

In the context of e-commerce, one of the adaptation techniques proposed for navigation support is applied to avatars (Frery et al., 2002; Chittaro & Coppola, 2000; Dos Santos & Osório, 2004a). For instance, (Chittaro & Ranon, 2000) proposed so-called *Walking Products* (WP) which are used to increase the exposure level of a product according to the customer interests. A WP is represented as an avatar that represents a specific product in the shop and navigates through the virtual shop toward its corresponding product, which could be an interesting product for the user. Likewise, work presented in (Dos Santos & Osório, 2004a) proposed so-called Intelligent Virtual Agents (IVAs) which are used to direct users to their preferred product area inside the virtual shop. The IVAs can be also called explicitly once the users need help or more information about a specific product. This technique is interesting because it simulates the real life shopping where one can ask for help to a salesman. The previous work provides us with interesting adaptive navigation techniques that can also be used in the context of adaptive 3D VLE. Using such a technique, learners can be directed towards specific learning materials and topics.

Other researchers like in (Hughes et al., 2002) extended some adaptive hypermedia methods to 3D VEs by developing a so-called *Attentive Navigation* technique. The *Attentive Navigation* techniques provide algorithmic mechanism to find the ideal viewpoints for the users' view camera depending on the user profile. To achieve that, the authors proposed so-called *Annotation, Hiding, Sorting,* and *Direct Guidance* techniques. For instance, the *Annotation* technique is used to mark objects that are interesting for the users. This technique, for instance, can change the colour of the link to be annotated, add icons, or add special marks to indicate objects' position. In Figure 3.2, two techniques are illustrated to get the user attention for interesting objects. In part (a) of the figure, an arrow pointing to the possible interesting object is used to direct the users to the object. Part (b) of the figure shows a flashlight technique to mark the virtual object. This work is interesting because it validates the idea of extending some of the adaptive hypermedia methods (such as direct guidance, hiding, sorting which were presented in section 2.3.3) to 3D VEs. Our work also follows a similar approach by extending some of the adaptation techniques proposed in the context of hypermedia applications (see next Chapter).

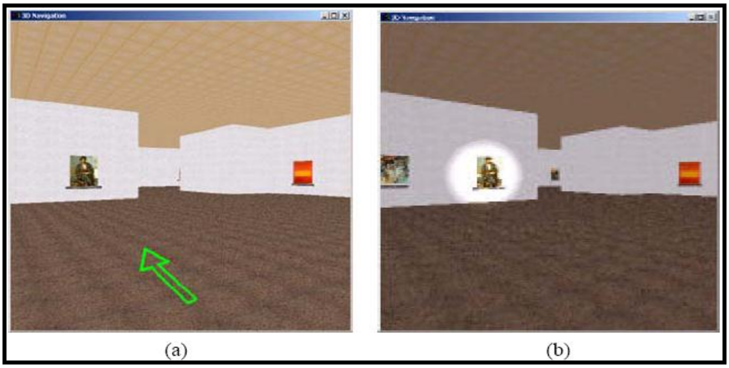


Figure 3.2: Adaptive Navigation Technique (taken from (Hughes et al., 2002))

Another adaptive navigation technique is related to providing an adapted navigation path. For instance, authors of (Nussbaumer et al., 2009; Chittaro et al., 2004; Brusilovsky, 2004) provided a navigation guidance to the learner by highlighting the path through the 3D virtual learning landscape and using 2D or 3D maps for the 3D virtual world to support users in their navigation. Such an adaptation technique is also useful to consider in the context of adaptive 3D VLEs, to provide a specific learning path to the learner rather than providing a learning sequence based on pedagogical aspects.

E-learning Context

It was realized that learners should be able to locate the required 3D learning materials easily. To achieve that, researchers explored different adaptive navigation techniques. Some adaptation techniques are used to restructure the 3D virtual space so that the learner can find the suitable materials easily and fast as in (Aquino et al., 2005; Hossain et al., 2004) or to insert some landmarks objects adaptively as in (Castelli et al., 2008).

Next, some examples on how to provide adaptive navigation techniques in the context of e-learning applications are presented.

Back to the work presented in (Chittaro & Ranon, 2007b), the work support adaptation in navigation by using a specific tag that can be translated by the AHA! engine as a conditional tag. The tag is used to enable learner to navigate to different 3D objects or to read specific learning materials. Depending on the user model, the AHA! engine will decide when to use the code inside the conditional tag. To demonstrate their work, they have developed a course on learning X3D where adaptive navigation techniques are realized rather than applying adaptation to only 3D content representation (Chittaro & Ranon, 2008).

Returning to Santos and Osorio work (already mentioned in the previous section on e-commerce), the authors also validated their AdapTIVE approach by applying it in the context of distance learning (Dos Santos & Osório, 2004b). The 3D VE has two types of users: students (consumer) and teachers (provider). All the data related to the users are stored in the user model. According to the user model, the content generator will organize the virtual environment. To have the 3D VE similar to a real learning environment, they divided the 3D VE into different rooms. Every room has study material related to a specific domain. For example, if students are studying computer science, then they can find a room for Artificial Intelligence, a room for Programming languages, etc. The artificial intelligence room is also divided into sub rooms for different fields of artificial intelligence like Artificial Neural Networks, Generic Algorithms, and Multi Agent System. The spatial order of the sub rooms are automatically changed depending on the field that is interesting for the student. Most interesting rooms are located before the less interesting ones. Additionally, the user is able to call the Intelligent Virtual Agent (IVA) whenever it is needed. They can communicate with the IVAs in a verbal and nonverbal way. The idea behind applying this approach to the learning domain was to prove that the AdapTIVE approach is generic enough to be applied in different domains. However, adaptation cannot happen at run time as it generates the contents with the new adaptive behaviour after predefined number of visiting sessions. The work is also interesting as it divides the courses into different learning rooms related to specific topics. However, adaptation cannot happen at run time as it generates the contents with the new adaptive behaviour after a predefined number of visiting sessions. Our approach aims at supporting adaptation at run time.

3D Educational Games Context

Similarly, in the context of educational 3D games, researchers considered providing adaptive navigation techniques based on virtual agents. For example, the authors of (Westra et al., 2009, 2011) proposed how to adapt agents guidance inside the 3D educational game based on the player model. The authors of (O'Neil et al., 2005) proposed adaptive navigation by providing navigation maps which reduced the users' cognitive load. Such an approach can also be adopted in the context of adaptive 3D VLE. An extensive overview about the literature related to providing adaptation to virtual agents in educational and serious games is presented in (Hocine & Gouaïch, 2011).

3.1.3 Interaction Adaptation

In addition to providing adaptation to content presentation and navigation, researchers also revealed that user interaction can also be adapted according to the user's knowledge and background (Octavia et al., 2010; Celentano & Pittarello, 2004; Dong et al., 2008; Blanco et al., 2009; Hunicke, 2005). For instance, adaptation techniques can be applied to different activities like building 3D structures, moving 3D objects in the space, interacting with 3D virtual objects, etc. The following presents some adaptation techniques that can be applied to user interaction in the context of e-commerce, e-learning, and educational 3D game.

E-commerce Context

Adaptation techniques for user interaction in the context of e-commerce are usually integrated with adaptation of 3D content and navigation. For instance, the authors of (Celentano & Pittarello, 2004) developed a prototype which supports adaptive navigation and interaction techniques. A user's behaviour 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 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 during interactive execution of tasks and are encoded in finite state machines. 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. Another example is presented in (Chittaro & Ranon, 2002b) which used the approach proposed in (Chittaro & Ranon, 2002a) to introduce some interaction adaptation techniques like interaction with the specific so-called walking products. Researchers in (Lepouras & Vassilakis, 2006) limited user interaction depending on the recorded interaction time with a specific object.

E-learning Context

The authors of (Celentano & Nodari, 2004) validated their approach also in the context of e-learning. The authors of (Dong et al., 2008) attempted to provide adaptation interaction techniques in the context of desktop based VR application. In principle, their approach is also applicable for web-based applications. The authors proposed adaptation interaction techniques which can be applied to the zoom level, mouse sensitivity, and rotation origin. Such techniques are used to fit user experience for performing training tasks.

To the same end, the authors of (Cubero et al., 2004) proposed online adaptation to student interaction depending on his performance of solving exercises. According to student performance, the exercises difficulty level will be adjusted.

Work presented in (Octavia et al., 2010) proposed "(1) switching between interaction techniques, which offers the most suited interaction technique for a user in a certain situation, (2) adapting the interaction technique itself that is adjusting parameters of an interaction technique to control how the user should perform it, and (3) enhancing the interaction technique with modalities". The proposed approach is not developed in the context of web-based 3D VE, but in principle it is also applicable in this context.

Adaptation techniques are also proposed to deal with learning activities within collaborative 3D VEs (Monahan et al., 2008; Weusijana et al., 2007), but as we are not dealing with collaborative environments they are out of scope.

3D Educational Games Context

One of the most obvious interaction adaptation techniques in video games is the adjustment of the game difficulty level and challenges to fit users' skills (Moreno-Ger et al., 2009; Hunicke, 2005). Such a technique is mainly useful in the context of entertainment and educational games.

Other researchers suggested adding more complex tasks rather than trivial ones to avoid having too easy game. This is presented in (Westra et al., 2011; Katsionis & Virvou, 2004). Furthermore, the authors of (Kickmeier-rust et al., 2008) proposed delivering learning situations, which require different interactions and activities to be performed, according to the overall skills level of the learner. This technique is also interesting for 3D VLE, as it can be used to deliver adapted tasks to the learner.

Adaptive interaction is also supported by a coherent feedback. For instance, encouragement and congratulation messages or even task performance failure messages are all considered in providing adaptive interaction techniques (Hocine & Gouaïch, 2011). In general, supporting learners with feedback is considered important in the context of educational 3D VE. For instance, some remarks or grades should be given to the learner to encourage and motivate him during his learning process. This is partly considered in our work.

Some researchers proposed interaction adaptation to be applied to the user's learning style (e.g., visual or auditory learning style) (Sancho et al., 2009). However, we did not consider this further, as we believe that adaptation to the learning style is best performed before starting the learning and not at run time (which is our focus). Indeed, the learning style of a person is rather stable and will not change during the learning process.

3.1.4 Summary

Providing adaptation techniques in the context of 3D VE has been studied in the past years. We have examined these techniques and categorized into three groups: adaptation techniques for content and materials, for user navigation, and for interaction techniques.

Concerning applying adaptation to 3D contents, most of the proposed solutions are related to adapting the virtual scene to fit the user's interest and knowledge, and changing virtual objects behaviours. It is also noticed that content adaptation is applied by inserting, deleting, updating level of details, and changing colour of 3D virtual objects.

With regard to adaptive navigation support inside 3D VEs, some of the proposed adaptation techniques are used to identify navigation paths through the virtual world by highlighting and annotating paths towards 3D virtual objects. In addition, adaptation techniques can also be applied to the avatar's gazing and orientation while the user is moving along a specified navigation path.

With reference to adaptive user interaction, also a number of adaptation techniques were proposed in different contexts. For instance, adaptation can be applied to the navigation speed and zoom level. In addition, some techniques are adapting the flow of the user interaction. In the context of educational games, game difficulty adjustment is also used.

The adaptation process in most of adaptive 3D VE and educational games starts by monitoring the user's behaviour and his activities inside the 3D VE. This is to build and update a user model. After that, adaptation can be applied to 3D contents, objects' behaviours, and difficulty level to fit the user's experience and knowledge level.

Although a lot of different adaptation techniques have already been proposed, most of the proposed adaptation techniques are specific to certain 3D VEs. We believe that it would be useful to provide a systematic overview of possible adaptation techniques for 3D VEs. Such a systematic overview will also allow identifying more adaptation techniques that can be applied in the context of adaptive 3D VLEs. Furthermore, there is a need to express the adaptation techniques at a more abstract level, which can be applied by novice users during the authoring of 3D VLE. This work can be found in section 4.3.

3.2 Authoring for E-learning

In this section, we review authoring tools and approaches relevant for adaptive e-learning. This section is divided into two subsections. The first section reviews common authoring tools and approaches in the context of adaptive hypermedia. Secondly, a literature review related to authoring adaptivity in the context of 3D educational games and 3D VLE is presented.

3.2.1 Authoring Adaptive Hypermedia E-Learning

As mentioned in the beginning of this chapter, we consider adaptive hypermedia as an inspiring domain for our work. Therefore, we will discuss authoring aspects used in the context of adaptive hypermedia.

There are many authoring tools developed to support creating adaptive hypermedia applications. Some of them, which are related to our work, are more oriented to specify the pedagogical aspects along with adaptation techniques. To this end, one of the most important challenges is to hide the low-level implementation details and allow the author to concentrate on both adaptive and pedagogical aspects.

Besides the standard functionalities that should be supported, like enabling authors to define the structure of the hyperspace, adaptive educational hypermedia authoring tools enable authors to define, from a high level point of view and without resorting to specific scripting languages, how the user model should be updated to perform adaptivity tasks dynamically (Brusilovsky, 2003).

The next sub sections will discuss the well-known authoring tools for adaptive educational hypermedia systems. Namely, *MOT* (Cristea & De Mooij, 2003), *ACCT* (Dagger et al., 2004)*, LAMS* (Lams, 2004), *AHA!* (De Bra et al., 2003), and *GRAPPLE* (De Bra et al., 2010). However, more authoring tools are proposed for designing adaptive educational hypermedia applications. Note that a detailed review of adaptive hypermedia applications and their corresponding authoring tools is given in (Brusilovsky, 2003; Paramythis & Loidl-Reisinger, 2003; Georgouli, 2011). A state of the art report (till 2007) concerning authoring adaptation in the context of hypermedia can be found in (Hauger & Kock, 2007).

3.2.1.1 My Online Teacher

My Online Teacher (MOT) (Cristea & De Mooij, 2003a) is an authoring system for designing and creating adaptive hypermedia courses. MOT is considered to be a simple and user-friendly authoring system. The authoring tool allows authors to define different models based on the Layered WWW AHS Authoring Models (LAOS) (Cristea & De Mooij, 2003b), which uses a clear separation between the course content and pedagogical aspects, like the domain model, the goal model, the presentation model, the user model, and the adaptation model. In general, the adaptation model is specified using the LAG (Layers of Adaptation Granularity) framework (Cristea et al., 2003c).

For instance, the MOT authoring tool allows educators to define the domain model based on a hierarchical tree structure of learning concepts, which have different attributes such as content type. Based on the defined domain model, MOT also supports educators to construct the goal map to create an adaptive lesson. More particularly, the author can define adaptation strategies using a dedicated authoring tool and such adaptation strategies are defined using the LAG adaptation language (Cristea et al., 2009). A number of adaptation strategies have been proposed. For instance, the Visual-Verbal strategy, which is used to determine either to display the course contents as text or as images and videos. Another strategy, so-called the Relation-based strategy, is used to show leaning concepts based on the type of user (such as beginner, intermediate, or advanced). A description of the supported adaptation strategies can be found on the adaptation strategy website[43].

[43] http://prolearn.dcs.warwick.ac.uk/strategies.html

Different extensions to the original MOT have been developed. For instance, MOT 2.0 is an extended version that supports authors and learners with a social web adaptive hypermedia authoring tool and delivery environment (Ghali et al., 2008). Furthermore, some new functionalities and an interface displaying the domain model in graphical way, were introduced. MOT3.0 (Foss & Cristea., 2009) comes with more usability features such as drag-drop mechanism, WYSIWYG, etc. Moreover, it enables educators to import different type of learning materials and resources such as PowerPoint, Wikipedia, SCORM, etc. which are defined by other authors.

In general, MOT is considered as an interesting authoring tool, as it supports interoperability with different adaptation engines such as in AHA! and GRAPPLE. Some aspects of the authoring approach are also used in our approach, e.g., the use of high-level adaptation strategies, and the need to support non-expert authors in designing an adaptive course.

3.2.1.2 Adaptive Course Construction Toolkit

Adaptive Course Construction Toolkit (ACCT) (Dagger et al., 2004, 2005) was developed to support authors (educators) to create adaptive hypermedia courses.

The authoring approach is mainly depending on defining four models: domain model, goal model, pedagogical model, and adaptive model. First, the author needs to identify the objectives and goals of the course (goal model). Based on the created goals, the author should define appropriate pedagogical strategies (pedagogical model). Next, the author needs to create the different learning concepts for the course domain model. Of course, he is able to associate the concepts with learning materials and resources. After that, adaptation will be realized by applying adaptivity techniques to the pedagogical elements based on so-called adaptive axes like learning style, preferences, and knowledge (adaptation model). This is achieved by associating narrative attributes (adaptive axes) with the concepts and learning activities of the course. As a result of the defined adaptation techniques (like object inclusion/ exclusion, link annotation, etc.), the learning material will be selected adaptively.

ACCT has interesting features. It incorporates, for instance, a graphical language for creating the different learning concepts that populate the course domain model. Furthermore, before publishing the course, the author is able to test the different functionalities of the developed course by using the Adaptive Personalized eLearning Service (Conlan et al., 2002).

The tool was evaluated with secondary school teachers and other domain experts. Results were promising as the tool allows novice authors to create an adaptive course from scratch. However, one of its limitations is the fact that the author is not able to define his own adaptive rules as the personalization and adaptive rules are predefined.

3.2.1.3 Learning Activity Management System

Learning Activity Management System (LAMS) (Lams, 2004) is authoring tool for designing online collaborative learning courses. The tool provides the author with a graphical user interface. Furthermore, a visual environment with drag and drop functionality is provided by the authoring tool for creating different learning activities like chat session, documents, and media files.

One of the examples about using LAMS is given in (Campbell & Cameron, 2009; Dalziel, 2003) which reports some promising results. More details can be found on the website of LAMS[44]. However, LAMS does not enable authors to define adaptivity for a course explicitly. In fact, the adaptation happens according to the course content and according to the defined user's activities, which are related to selected predefined pedagogical template.

3.2.1.4 Adaptive Hypermedia for All!

AHA! (De Bra et al., 2003) provides an adaptive hypermedia architecture which proposes authoring tools to help the teacher or instructor to design their own adaptive courses.

The authoring approach is also depending on four models, which are the domain model, the pedagogical model, the user model, and the adaptation model. The domain model includes the learning concepts. The pedagogical model includes conceptual relationships such as the prerequisite relationship between learning concepts. The user model is used to identify the attributes that are used to represent information about the user such as number of visits, and knowledge level. Finally, the adaptation model is related to the adaptation behaviour for the conceptual relationships.

In AHA!, adaptation rules are responsible for updating the user model. Moreover, adaptation rules support adaptation techniques related to user navigation and content representation. On the one hand, to help users in their navigation, the suitable links will be displayed in specific colours (blue for unvisited links, purple for visited ones), otherwise, the links will be in black (see Figure 3.3). Using these techniques, learners will easily see what they already read and what still need to read. On the other hand, for the content presentation, every page can have conditional fragments that may contain some more detailed text about a specific concept. Requirements that determine the visibility of such fragments are specified by (Boolean) expression using user model attributes. Therefore, the learner will get suitable detailed information depending on his/her level of knowledge.

[44] http://www.lamsinternational.com/

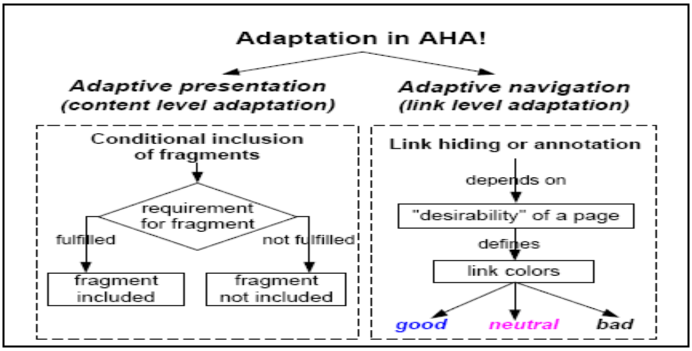


Figure 3.3: Adaptation Techniques in AHA! (taken from (De Bra et al., 2003))

One of the interesting features in AHA! is that it provides two graphical authoring tools, namely the Graph Author and the Concept Editor. The Graph Author tool enables authors to develop their own applications without the need to be familiar with the low level adaptation rules required between the course's concepts. Such adaptations rules will be generated automatically from the author graph. If the authors like to create their own adaptation rules, the Concept Editor tool gives them the possibility to do this using a graphical user interface. Therefore, AHA! supports both novice and advanced educators in authoring an adaptive hypermedia course. Our work follows the same approach for enabling both types of educators (beginners and experts) in designing adaptive 3D VLE.

3.2.1.5 GRAPPLE

Another interesting project is the GRAPPLE[45] (Generic Responsive Adaptive Personalized Learning Environment) project, an EU FP7 STREP project aimed at the construction of a generic adaptive learning environment that can be used/accessed at home, school, work, or on the move.

The GRAPPLE project is mainly oriented towards classical (text and multimedia) learning resources. However, the project also investigates the use of other types of learning materials, such as VR and simulations. It also allows importing information about the learner from external learning management systems (De Bra et al., 2010). Furthermore, GRAPPLE includes authoring tools that enable educators to specify adaptation strategies for content and activities.

The authoring tool of GRAPPLE allows a course author to define a course at a conceptual level. This is done by means of three sub-components: the Domain Model tool, the Pedagogical Relationship Types tool, and the Conceptual Adaptation Model tool. The authoring tool is implemented as a web-based application (Hendrix et al., 2008). The Domain Model describes the

45 http://www.grapple-project.org/

concepts that should be considered in the course. The Conceptual Adaptation Model expresses at a high-level and by using pre-defined pedagogical relations, such as the prerequisite relation (which are defined using Pedagogical Relationship Tool), how the content and structure needs to be adapted at runtime. The authoring tool allows an author to specify the Domain Model as well as the Conceptual Adaptation Model in a graphical way.

As already mentioned earlier (section 1.5), the work presented in this dissertation is based of the work done on adaptive 3D VLEs in the context of GRAPPLE, which was on its turn based on the authoring tools developed for the classical learning materials.

3.2.1.6 Summary

The five authoring tools, shortly reviewed, provide an easy to use authoring interface to create adaptive hypermedia courses. The authoring tools allow authors to define the structure of the course, specify adaptation techniques, and define the pedagogical and instructional aspects that need to be considered in the adaptive course.

In principle, different functionalities of these authoring tools developed in the context of adaptive hypermedia can also be considered in an authoring tool for adaptive 3D VLE. Indeed, an adaptive 3D VLE authoring tool also needs to enable educators to create the pedagogical structure of the 3D VLE (i.e. course) and design adaptation behaviour of the 3D VLE (i.e. course). Furthermore, using visual languages, as done in the adaptive hypermedia authoring tools, for defining different aspects in the authoring process can also be investigated.

Next, we will discuss previous research done in the context of authoring adaptive video games and in the context of 3D virtual environments for e-learning (section 3.2.2).

3.2.2 Authoring Adaptive 3D Games and 3D VLE

This section presents different approaches and methodologies proposed to author and develop adaptive 3D VLE, as well as educational games. It is important to mention that authoring adaptive 3D VLEs is still in its research phase, with only few works conducted in this domain. Therefore, we also investigated authoring adaptation in educational game, as, to some extent, there are similarities with 3D VLEs.

For educational games, there are quite a number of projects that developed authoring tools targeting non-programmer educators and aim to develop adaptive and interactive 3D educational games based on visual languages representing story content and flow, and educational aspects.

Among those projects are INSPACE[46], U-CREATE (Sauer et al., 2006), ELEKTRA[47] and 80Days[48] which are reviewed in the next subsections.

Note that we will not consider tools for the actual creation of 3D games, as we will also not consider the actual creation of the 3D VLE.

The following subsections give an overview about authoring approaches (section 3.2.2.1), conceptual models (section 3.2.2.2), and visual languages (section 3.2.2.3) that are proposed to design adaptive educational 3D games and 3D VLE.

3.2.2.1 Authoring Approaches

This section reviews authoring approaches that are proposed in the context of 3D VLEs, as well as in the context of educational 3D games. We start with authoring tools and approaches for educational 3D games.

Authoring of Educational 3D Games

Quite some research has been performed to support non-specialist authors to create adaptive educational 3D games. Some researchers proposed an agent-based approach which delivers adaptive agent behaviour depending on the user profile (Hocine & Gouaïch, 2011; Conati & Manske, 2009; Ram et al., 2007; Spronck et al., 2006). Others suggested approaches depending on composition of a soft skill simulation (NexLearn, 2009; ExperienceBuilder, 2000). In general, a narrative-based approach is the common authoring approach (Dickey, 2006; Marchiori et al., 2012; Kiili, 2005a; Lindley, 2005; Göbel et al., 2008). We focus on this latter approach as our work is also meant to deliver an adaptive storyline inside the 3D VLE.

In (Moreno-Ger et al., 2007), in the context of the <e-adventure> approach (Moreno-Ger et al., 2008a, 2008d, 2008b), a documental approach for the development of graphical adventure (point-and-click) videogames is proposed. The key development element is the game's storyboard that is given using a markup language <e-Game> (Moreno-Ger et al., 2005), from which the different art assets needed are referred, and the final executable videogame is automatically generated. We considered this approach as interesting as it enables authors to define stories, assessments, and adaptation mechanism and to end up with a running game. However, although the authors present this <e-Game> language as an authoring approach suitable for non-programmers, we rather see this language as a candidate for an intermediate language between a true end-user authoring tool and a game engine.

[46] http://www.inscapers.com
[47] http://www.elektra-project.org/
[48] http://www.eightydays.eu/

In addition, based on gathered feedback and evaluation of the actual development of an educational game in the context of the <e-adventure> platform, a new work called Writing Environment for Educational Video games (WEEV) was proposed (Marchiori et al., 2012). The work presents an authoring approach based on an comprehensive description for games based on the narrative metaphor (Marchiori et al., 2012). In particular, the proposed approach is used to simplify authoring a story based on the explicit definition of the player interaction and the different elements of the game. In general, WEEV came to reduce the complexity of authoring task in <e-adventure> approach.

The ELEKTRA project[49] aimed to bridge the gap between cognitive theory, pedagogy, and gaming practices. ELEKTRA's pedagogical conceptual model combines Bloom's taxonomy of educational objectives (Bloom, 1956) with the eight Learning Event Model (Verpoorten et al., 2007), which results in 48 learning activities (Kickmeier-Rust et al., 2007). In this project, they also integrated an adaptive framework for storytelling. A new terminology was introduced namely macro- and micro adaptivity. Macro-adaptivity refers to traditional techniques of adaptation such as adaptive presentation and adaptive navigation. Micro-adaptivity is adaptation within learning tasks. Micro-adaptivity 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 used an adaptation system that provide recommendations to the game engine, but it is the ultimate decision of the game engine whether or not to enact a recommendation (Kickmeier-Rust et al., 2007). In the authoring approach, on the basis of identified curricula for a learning topic, the author needs to identify learning objects. Accordingly, both game genre and background story will be determined. After that, a knowledge domain ontology, which includes required skills and the prerequisite relations among the skills, has to be established. Finally, both learning and assessment situations must be developed along with storytelling and game design (Bopp, 2008). Authors are also able to design adaptive Non-Player Character (NPC) which takes the role of a teacher by providing guidance, hints, and companionship (Linek et al., 2007).

Based on the research results and developments achieved within projects like INSPACE, U-CREATE[50], a StoryTec platform for interactive Digital Storytelling application was developed in the context of the European research project 80Days (Göbel et al., 2008; Sauer et al., 2006; Göbel et al., 2007). The 80Days project[51], the successor of ELEKTRA, addresses two major objectives,

[49] http://www.elektra-project.org/
[50] U-CREATE is a CRAFT project funded by the European Commission under reference COOP-CT-2005-017683.http://www.u-create.org
[51] http://www.eightydays.eu

(a) advancing frameworks of non-invasive personalization and adaptation on the macro as well as the micro levels and (b) establishing a methodology for reducing the costs of developing Digital Educational Games. This project follows the concepts of macro- and micro-adaptivity introduced in the ELEKTRA project. Pedagogically, the project is grounded in the framework of self-regulated personalized learning (SRPL) (Law & Rust-Kickmeier, 2008). The project consists of two major components: an authoring environment, as well as a runtime engine. The authoring environment allows educators to specify both micro and macro adaptivity. The authoring platform is composed of five modules and editors: Story Editor, Stage Editor, Action Set Editor, Property Editor, and Asset Manager, with graphical user interface. The components are also connected to each other to have a consistent data and structure for the created 3D game. For instance, if a new scene is inserted in the story structure then the other editors will be informed about that. One of the drawbacks of this authoring tool is the complexity of the authoring tasks themselves.

Authoring of 3D Virtual Learning Environments

Similar to educational 3D games, researchers proposed agent-based approaches to author adaptation inside a 3D VLE (Aquino et al., 2005; Smith et al., 2004; Greeff & Lalioti, 2001). Another work, presented in (Buche & Querrec, 2011), targets to deliver a pedagogical virtual agent providing instructive assistance. In the context of the Virtual Human Project (Virtual Human Project, 2002), researchers, in (Knöpfle & Jung, 2006), proposed an authoring approach which support authors to create a virtual character (avatar) in term of its representation, dialog behaviour, and animation in 3D VE. First, the authors are able to create virtual characters using 3D modelling packages like Maya or 3DSMax. Then the author needs to define a story script. A story script is stored as SceneAct-Nodes that are connected to end up with a nonlinear storyline. The project also incorporates storytelling structure inside 3D VLE. In general, the project provides means to intuitively model the virtual character in term of different specifications. However, providing adaptation mechanism was not explored in the context of this project.

Looking back to the work of Chittaro and Ranon (Chittaro & Ranon, 2007b, 2008, 2007a), discussed earlier, these researchers also proposed an authoring approach similar to AHA!. The author is able to define the conceptual relationships between the learning concepts. Furthermore, the author is able to apply adaptation techniques to user navigation and interaction using conditional fragments mechanism as in adaptive hypermedia (see section 2.3.3). However, they do not provide a dedicated authoring tool for specifying adaptation techniques and pedagogical aspects and previewing customized 3D models inside the virtual environment. There is still a great deal of programming involved in creating adaptive 3D course, as adaptation techniques are provided using some low level ad-hoc implementation.

As mentioned earlier (section 3.2.1.5), also the GRAPPLE project (De Bra et al., 2010) includes authoring tools, among which one dedicated for VR. The project is using an author-driven approach for specifying the adaptations, i.e. the author of the course is given the full control over the adaptation process. During the design of the course, the author needs to specify the adaptations explicitly through a set of rules. The created VR authoring tool (De Troyer et al., 2010) was realized by extending the regular authoring tool (i.e. for the classical learning material). The authoring tool allows an author to specify the required adaptation using *pedagogical based Adaptation Rules*. This is a very powerful approach, as the author of a course only needs to specify the desired results and not how this needs to be achieved. However, some drawbacks are revealed with the approach as well as with the tool, like rigid authoring tasks, and difficulties in designing a specific storyline. More details are published in (Ewais & De Troyer, 2014). As already mentioned earlier (section 1.5 and section 3.2.1.5), the approach proposed in this dissertation is based on the approach used in GRAPPLE for authoring 3D VLEs.

3.2.2.2 Conceptual Models for Authoring

Different approaches in the context of authoring adaptive educational 3D games and 3D VLE, have introduced certain conceptual models to support adaptation. In this section, we will review them and indicate which models we also use in our conceptual framework (described in section 5.3).

In general, most of the proposed frameworks are mainly based on three conceptual models (Charles et al., 2005; Missura & Gaertner, 2009; Lopes & Bidarra, 2011a), namely, the player model (or user model), the content model (or domain model), and the tasks and goals model. Figure 3.4 depicts how an adaptive engine along with the player model, tasks and goals model, and game content model cooperate to deliver a 3D educational game depending on the player profile.

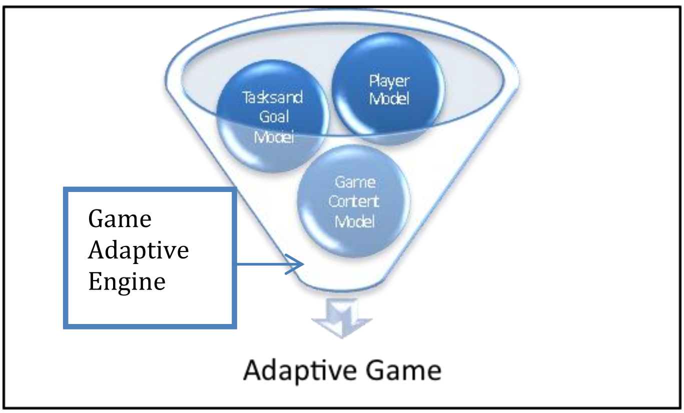


Figure 3.4: Generating adaptive game.

The following lists a number of conceptual models that are used most in the proposed frameworks:

- User (player) model is responsible for capturing information about the user (player), like his preferences, skills, knowledge level, age, and gender. This model is used as an input to deliver adaptive 3D educational games in (Göbel et al., 2009; Yun et al., 2009; Charles et al., 2005; Missura & Gaertner, 2009; Hodhod et al., 2009). Other researchers proposed to record the user activities to define the type of the player and update the user model accordingly (Togelius et al., 2007; Magerko, 2006). For instance, work presented in (Magerko, 2005; Thue et al., 2007) used Artificial Intelligence algorithms in order to analyse the player's actions or machine learning techniques as in (Beck et al., 2000). However, researchers also refer to the user model as *student model* or *player model,* which are all used for the same purpose. In the context of this research work, a user model is also considered as one of the important models in the proposed conceptual framework (see section 5.3.4). However, a conceptual extension to the standard user model concept is required to fulfil the adaptation requirements in the context of 3D VLE.

- Domain model is used to define the concepts that are covered in the educational game and the relationships that are defined between the different concepts (Hodhod et al., 2009; Kickmeier-Rust & Albert, 2007; Torrente et al., 2008). Furthermore, some researchers use a *content model* which is responsible for providing the adaptive engine or the game engine with the resources that will be manipulated, inserted, deleted, highlighted, etc. (Lopes & Bidarra, 2011a; Moreno-Ger et al., 2008c; Kickmeier-Rust & Albert, 2007). Our work also uses a domain model to define the different learning concepts to be covered during the learning process (see section 5.3.1).

- Instructional model is used to define how (strategy) the learner should learn about the learning concepts rather than just defining what (resources) the learner needs to learn (domain model or content model). This is outlined in (Begg et al., 2005; Mills & Dalgarno, 2007). The adaptation techniques are defined in the instructional model to express how the learner should proceed. Other researchers called it the *pedagogical model* which is defined on the basis of constructivist theories, situated learning, problem-based learning, etc. (Westra et al., 2009; Pivec & Pivec, 2008). We also follow a similar approach. A pedagogical model is used in our authoring approach to specify the pedagogical structure of the content for the 3D VLE (see section 5.3.2).

- Task Model is used to utilize user skills during his learning. Therefore, researchers suggest realizing adaptation by providing an adaptive sequence of the learning tasks depending on the user's progress. For instance, work presented in (Bellotti et al., 2010; Westra et al., 2009, 2011; Medler, 2009) proposed some adaptation mechanisms related to adaptive task selection. To a certain extent, we also utilize such a model but it is integrated in the adaptation model.

- Dialog Model represents the possible conversations that learner may have with virtual characters inside the educational 3D games. This model is considered as an essential model specially for dialog-based games (NexLearn, 2009; Gaffney et al., 2010; ExperienceBuilder, 2000). Our work does not support dialog aspects during the learning process, as this is a technique more common in the context of games.

- Agent Model or *opponent model* is responsible for performing opponent's behaviours inside the educational 3D games (Spronck, 2005). Researchers also investigates the possibility of providing adaptivity using agent model as discussed in (Hocine & Gouaïch, 2011; Medler, 2009; Westra et al., 2011). Adaptation can be applied to agent tasks and his behaviour inside the 3D VLE (Aquino et al., 2005; Kang & Tan, 2010; Buche & Querrec, 2011). Such model is most common in the context of game based educational applications and therefore not considered in our work.

- Story Model is basically used to enforce the learner to follow a specific story flow to reach the learning goals. Sometimes it is called *narrative model*. This model is considered as an essential model to create story-based educational games (Carro et al., 2002; Magerko, 2005; Hodhod et al., 2009; Togelius et al., 2007; Lindley, 2005; Dade-Robertson, 2007). More details about the story model will be presented in the next section. As our work aims at delivering a specific learning path by grouping related learning concepts in a specific sequence, we also need such a model. However, in our approach, such a story model is on the one hand integrated into the adaptation model, and on the other hand given by what we call the Course Model. More elaboration on that can be found in section 5.3.3

After reviewing a number of authoring approaches and the conceptual models mostly used, the following section will present techniques and notations that are used in the visual languages proposed to author and create adaptive 3D educational games and 3D VLE.

3.2.2.3 Visual Languages for Authoring

In this section, we will review interesting work that focus on the authoring of adaptive and story-based 3D games and 3D VLE using visual languages. We start with visual languages for authoring educational 3D Games.

Visual Authoring Languages for Educational 3D Games

There are different attempts to adapt UML and flowchart diagrams for authoring story-based games. For instance, LaMothe (LaMothe, 2003) adapted the state transition diagram from UML to describe the artificial intelligence characters in adaptive educational 3D games. Likewise, Lewinsky (Lewinski, 2000) proposes the use of flow charts to represent the story flow in the same context. However, the graphical notations in both works do not cope with complexity and the consequence is that the models quickly become complex and difficult to be understood. Therefore, they are not suitable for educators who don't have a good modelling background.

A more simple visual language is called SCRATCH (Resnick et al., 2009). The SCRATCH language can be used by young children (primarily ages 8 to 16) to design an interactive game and to present the program logic using different visual representations. To the same end, researchers proposed the ScriptCard visual language in the context of the Adventure Author project (Robertson & Good, 2005), to enable young authors (10-14 year old children) to express their story events in their preferred order. In general, both works are more oriented towards providing an easy to use interface for programming the game logic.

Also, in the context of the StoryTec tool (80Days project), visual languages are used; scenes and complex scenes are represented by rectangles and transitions by arrows in the Story Editor. Similarly, researchers in WEEV (Marchiori et al., 2012), defined the game story by using "a state-transition diagram" (see Figure 3.5), "where each state represents a point in the game story and each transition an interaction by the user with the system, which moves the story along." (Marchiori et al., 212). However, such visual languages do not support defining adaptation explicitly.

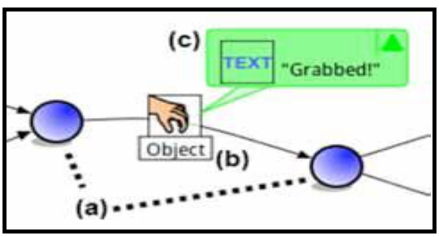


Figure 3.5: story-flow language's elements (taken from(Marchiori et al., 2012))

Defining interactive elements is also supported by using graphical notations based on UML-Activity Diagram[52] (see Figure 3.6). The proposed language is used to define the logical relation between the different scenes of

[52] UML Activity Diagram is part of the UML 2.0 specification (www.uml.org).

the story. In other words, "the logic is described by a set of rules which are composed of actions (e.g., walkTo 'Door', playSound 'intro.wav') and conditions (e.g., 'if player1 enters the castle')" (Göbel et al., 2008). In more details, an Action element is represented by a rectangle and a condition by a rhomb, which are sequentially arranged. As this visual language is based on the activity diagram in UML, it requires some background in modelling.

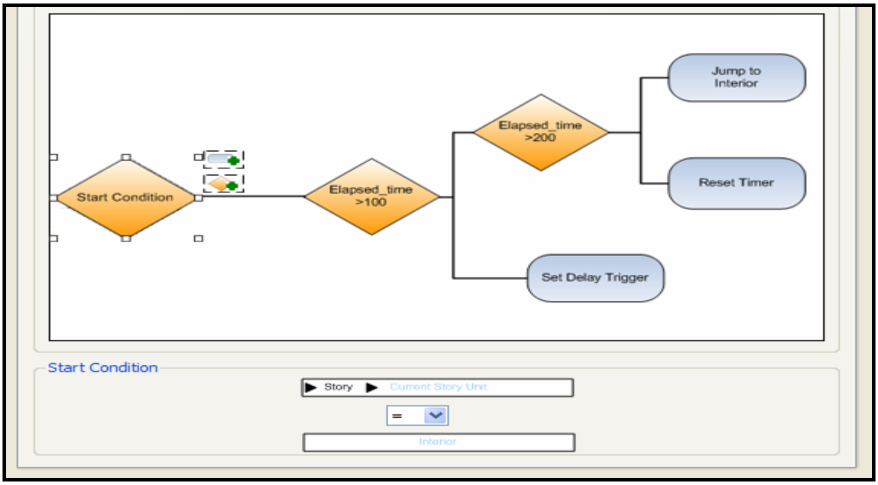


Figure 3.6: Defining the logical relation between the different scenes (taken from (Göbel et al., 2007))

Visual Authoring Languages for 3D Virtual Learning Environments

The Virtual Human Project[53], which is aimed at creating virtual human character and its behaviour specifications inside the 3D VLE, allows creating scenarios and topics for the created virtual human character (Schneider, 2002; Knöpfle & Jung, 2006). The tool is composed of a form-based editor where the virtual character specifications are defined and a graph-based visualization for creating a storyline where the flow of the topics inside the 3D VLE is defined. However, more details about the storyline like when the learner needs to move to a specific story and what can be adapted in the 3D VLE cannot be specified. The authoring tasks are quite complex as the authoring tasks includes many textual instructions and forms rather than visual languages. Therefore, a considerable amount of scripting is involved in the authoring process.

In the context of GRAPPLE, a visual language is used for defining the VR conceptual adaptation model (De Troyer et al., 2010). Figure 3.7 depicts a screenshot of the conceptual adaptation editor. Oval shapes are used to indicate start and end. A square annotated with *AB* denotes an Adaptation

[53] http://www.virtual-human.org

Block and is used as a placeholder for the learning concepts. Adaptation blocks are connected with arrows. The named arrows represent VR Pedagogical Relationship Types, which contains the adopted adaptation state and the adaptation rules. During evaluation, we observed that integrating pedagogical aspects with adaptation techniques in this visual language was perceived as confusing by the authors.

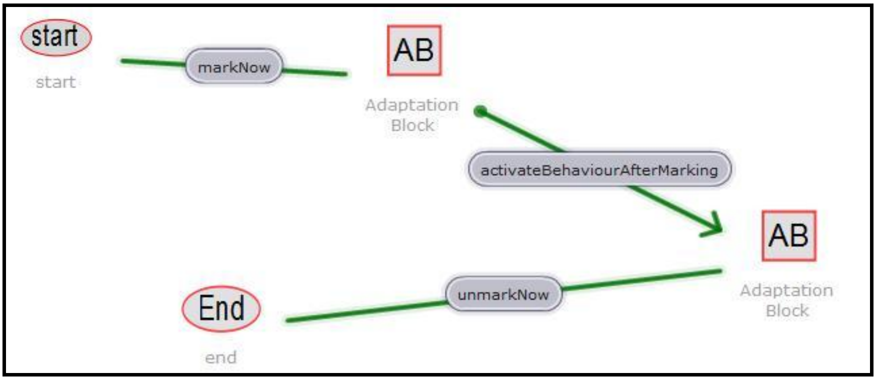


Figure 3.7: Visual language for the CAM in GRAPPLE (taken from (De Troyer et al., 2010)

3.2.2.4 Summary

In the previous subsections, we reviewed authoring approaches that are proposed to enable non-programmer educators or youngsters to create adaptive 3D educational games. As mentioned earlier, the proposed approaches are aimed to abstract from low level details and the adopted technology. Furthermore, a narrative-based approach is a common authoring approach in educational games. It reduces designing an educational 3D game into defining a sequence of different stories.

To realise the proposed authoring approaches, researchers proposed the use of different models. As a result, several models like player model, content/domain model, instructional/pedagogical model, and story model are defined to model different aspects needed for performing adaptation inside educational 3D games.

Researchers also proposed visual languages to help authors in the authoring tasks and to avoid the need for resorting to technical programming languages. Most of the proposed visual languages are graph-based and represent states of the story and transition between them using different notations. Some of the notations are adopted from UML diagrams, flow charts, and others are specially created for the purpose.

In general, most of the proposed authoring approaches and visual languages are created in context of large projects, such as e-adventure, Adventure Author, ELEKTRA, 80Days, and WEEV platforms. Table 3.1

summarizes some important features and aspects of those projects in term of authoring approach, authoring task complexity, pedagogical aspects, and adaptation techniques. Adventure Author and WEEV use visual languages for their authoring approaches. The others use easy to use graphical user interface editors to support educators in their authoring task. Authoring complexity is perceived as easy for some of the approaches and difficult for others (Marchiori et al., 2012). Such results are confirmed in different experimental evaluations. Educational aspects are covered in different way such as integrating educational materials in the game stories like in Adventure Author. While in e-adventure and WEEV platforms, researchers proposed some assessments and evaluation mechanisms to realize the pedagogical aspects in the educational games. Moreover, pedagogical theories like 8-Learning Events Model and Self-regulated personalized learning are used in ELEKTRA and 80Days respectively. Finally, adaptation mechanisms and techniques differ from one platform to another. For instance, in WEEV, researchers proposed the adaptation techniques to be used to deliver stories flow adaptively. While in e-adventure, the authors enable adaptation techniques to 3D games contents. A new term for adaptation techniques, so-called micro and macro adaptation, was used in the context of ELEKTRA and its successor project (80Days).

	Authoring Approach	Authoring Task Complexity	Educational/ Pedagogical Aspects	Adaptation Techniques
e-Adventure	Editor Tool	High	Learners Assessments and Evaluation	Content based
Adventure Author	Visual Language (ScriptCards)	High	Narrative and dialog aspects	NA
ELEKTRA	Editor Tool	High	8-Learning Events Model	Micro and Macro Adaptivity Techniques
80Days	Editor Tool	Low	Self-regulated personalized learning	Micro and Macro Adaptivity Techniques
WEEV	Visual Language	Low	Learners Evaluation	Story based

Table 3.1:A summary of different features in some related projects

3.3 Conclusion

From surveying the related work about adaptation techniques and authoring adaptation in the context of 3D VE and educational 3D games, we can draw the following general conclusions. Different researchers have investigated new enhancing technologies and techniques to improve both 3D educational games and 3D VE by providing adaptive capabilities. Although quite some research and projects are performed in the context of authoring adaptive 3D educational games, such as ELEKTRA and 80Days. there is little work on investigating the possibilities of involving casual educators (i.e. non-VR experts) in authoring adaptive 3D VLE.

As a starting point for providing adaptation in 3D VLE, adaptation mechanisms and methods developed in the field of adaptive hypermedia can be considered, such as Brusilovsky's taxonomy (Brusilovsky, 1996). In fact, in the context of 3D VLE, there are already different approaches, as in (Hughes et al., 2002; Nussbaumer et al., 2009; Chittaro et al., 2004; Brusilovsky, 2004), which support adaptation capabilities that are inspired by the adaptation methodologies and techniques proposed for hypermedia. However, both environments, hypermedia and 3D VLE, are significant different in term of components, structures, interaction, navigation, and communications techniques, therefore it is necessary to define a dedicated set of adaptation types and strategies for 3D VLE. We will elaborate on this in the next chapter (Chapter 4).

Regarding adaptation techniques, most of the adaptation techniques proposed are applied either to the 3D contents' representation, user navigation, or user interaction as in (Castelli et al., 2008; Chittaro & Ranon, 2007b; Aquino et al., 2005). However, there may be other possible adaptation techniques that can be applied to aspects different than those three aspects. In addition, it would be useful to provide a systematic overview of possible adaptation techniques for 3D VEs. Furthermore, there is a need to express the adaptation techniques at a more abstract level, which can be applied by novice users during the authoring of 3D VLE. This work can be found in the next chapter in section 4.3.

Conclusions about authoring adaptive 3D educational games and 3D VLE can be formulated as follows. Firstly, most of the proposed approaches (in both domains), deal with predefined adaptation rules to define the adaptation flow inside a game or a 3D VE as in (Moreno-Ger et al., 2007; Marchiori et al., 2012; Resnick et al., 2009). As a result, the authors cannot explicitly define the adaptation flow for the games and 3D VLEs. Secondly, in the context of adaptive educational games, researchers proposed to enable authors to define a storyline and pedagogical aspects as in (Dickey, 2006; Marchiori et al., 2012; Kiili, 2005a; Lindley, 2005; Göbel et al., 2008). However, there is no work in the context of authoring adaptivity for 3D VLEs that enables authors to consider these aspects explicitly.

From surveying the related work that proposed different models to capture different aspects needed for the adaptation, we can conclude that, presently, the learner or user model typically captures the learner's data such as knowledge, skills, preferences, and styles. Another important model is the domain model that defines the different learning concepts that are to be presented to the students during their learning process. Another common model is the adaptation model that contains required techniques, mechanisms, and rules that are defined to infer how content presentation and navigational aids should be adapted. Those models are the most commonly used models, explicitly or implicitly, in the authoring process. However, there are other models used depending on the genre of the 3D game or 3D VE. For instance, a dialog model is used for a dialog-based 3D game.

Authoring tools that target casual educators, i.e. non-specialists in 3D/VR concepts, have mainly been constructed in the context of large projects like INSPACE, ELEKTRA, and 80Days. and were dedicated to developing adaptive, interactive and story-based educational games. In the same context, different visual languages were proposed to abstract from the actual implementation details for programming 3D educational games such as in (Göbel et al., 2008; Resnick et al., 2009; De Troyer et al., 2010; Marchiori et al., 2012;). We will also follow this approach to enable the casual (i.e. 3D novice) educators to specify when to apply adaptation techniques, to define pedagogical aspects of the 3D VLE, and specify the storylines to be followed by the learners. Furthermore, our work seeks to avoid the shortcomings and drawbacks that are revealed in the reviewed literature by providing an authoring approach that simplifies the design process of an adaptive 3D VLE and by clearly separate pedagogical and storyline aspects.

Chapter 4: Adaptation Techniques for 3D VLEs

The previous chapter presented and discussed the related work. It started with different adaptation techniques and mechanism used in the context of 3D VE. After that it focused on approaches and frameworks related to authoring adaptation in adaptive hypermedia, educational games, and 3D VE.

The main goal of this chapter is to present a systematic overview of possible adaptation techniques for 3D VEs. As already mentioned in section 3.1, a lot of different adaptation techniques have already been proposed for 3D VEs but these are scattered over different works. It would be useful to have a systematic overview of possible adaptation techniques for 3D VEs. Such a systematic overview will not only provide a collection of adaptation techniques that can be used for authoring adaptive 3D VLEs, but also allow identifying more adaptation techniques that can be applied in the context of adaptive 3D VLE. Proposing such a systematic overview of possible adaptation techniques is the purpose of this chapter.

The chapter is structured as follows: Section 4.1 presents the different components of 3D VE, i.e. the anatomy of a 3D environment. This 3D virtual environment anatomy makes it easy to recognize the different possible adaptations. Before doing so, we first discuss the difference between adaptation in hypermedia and adaptation in 3D VEs to motivate the need for a new set of adaptation types (section 4.2). In section 4.3, different adaptation types and strategies are proposed for different 3D VLE components. Finally, Section 4.4 summarizes the chapter.

4.1 3D Virtual Learning Environment's Anatomy

In order to provide a systematic overview of adaptation techniques, we will first investigate what can potentially be adapted in a 3D VE. For this purpose, we first analyse a 3D VLE's anatomy. Next, based on this anatomy, we can identify which components of a 3D VLE can be subject of adaptation and how (section 4.3).

As already explained, 3D VLEs are 3D virtual environments especially designed to teach people about a specific subject or assist them in learning some topics, learning concepts, or skills. They are realized using Virtual Reality

(VR) technology and they can range from rather simple web-based applications using desktop VR technology to very advanced cave applications where the user is completely immersed into the 3D environment. As we mentioned earlier, our work is focussed on desktop VR applications and more in particular on VR applications that can be experienced through a web browser. In desktop VR, a three-dimensional computer representation of a space is displayed on a normal computer screen and the users can move their viewpoints freely in the space and perform several actions, like interacting with objects (e.g., drag and drop an object or touching an object) and invoking behaviours associated with these objects. The objects in the space are in general 3D objects but also 2D objects are possible.

Conceptually, we can distinguish the following components and associated functionality in a 3D Virtual Environment (see Figure 4.1 for an illustration):

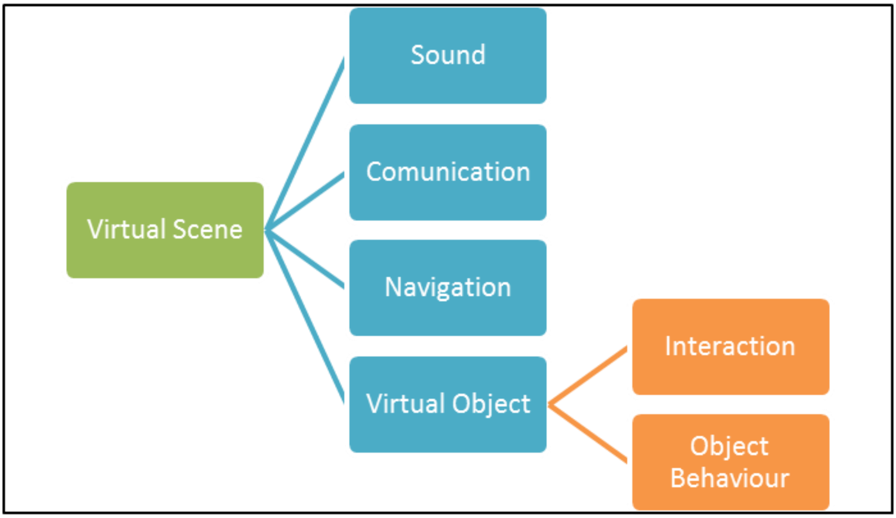


Figure 4.1: 3D Virtual Environment Components.

1. **The virtual scene:** The scene corresponds to the 3D space in which the virtual 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.

2. **The virtual objects:** The virtual objects are usually 3D objects, but there can also be 2D objects in the 3D space. They have a visual representation with colour and material properties, a size, a position in the space, and an orientation.

3. **Object behaviours**: The virtual objects may have behaviours. Behaviours may reflect real life actions and activities. For instance, objects may be able to move, rotate, change size, and transform. Objects behaviours can be triggered by some user interaction or by behaviours performed by other objects in the 3D space, or they may occur at or after a specific time.

4. **User interaction**: The user is able to interact with virtual 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 interactions can also trigger object behaviour, e.g., clicking on an object may start its behaviour. In desktop VR, 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. More details on this can be found in (Gutiérrez et al., 2008).

5. **User navigation**: The user can navigate in the 3D virtual environment by walking, running or flying through the 3D space. He can navigate by means of a so-called avatar. 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. Furthermore, the user can navigate through the 3D space by following a certain path, or marks can be used to guide users towards objects in the 3D space.

6. **Communication:** Nowadays, more and more 3D VE 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, however for others, learning environments do not need this collaborative aspect and are therefore single-user environments. In this research, we consider single-user environments.

7. **Sound:** A 3D 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.

We refer the reader to the following references (Kipper & Palmer, 2000; Bowman & Hodges, 1999) for more details about different components of 3D VE. More details about virtual object behaviours can be found in (Bernie Roehl, 1995).

Based on the different components that compose a 3D VE, we will present in section 4.3 a systematic overview of how these components can be adapted based on the learner's profile. However, first we will compare adaptation in web-based 3D VEs with adaption in hypermedia systems to justify the need for such a new set of adaptation types.

4.2 Adaptation in Hypermedia Systems versus Adaptation in Web-based 3D VEs

As already indicated by Brusilovsky (1996), and given the fact that in hypermedia applications (or standard web applications) the content is mostly text or 2D objects, the adaptive techniques for hypertext target either content or its presentation, or navigation. However, as indicated in the previous section, a 3D VE includes 3D objects populating a virtual scene; 3D object may have different behaviours and support different interactive techniques. This has an impact on possible adaptation techniques and their use. For instance, adaptation techniques that rearrange 3D contents may cause losing the correct structure of the 3D virtual scene. Moreover, users can navigate and interact through/with the 3D VE using different techniques such as walking, flying, jumping, touching, and approaching.

Table 4.1 shows the main differences between 2D and 3D web based systems. It is based on the differences table between hypermedia systems and 3D websites given in (Chittaro & Ranon, 2007a). The table compares web based hypermedia environments and 3D VEs in term of different aspects.

	Hypermedia Environment	3D Virtual Environments
Contents Type	Text, Image, Video, etc.	3D Object Representations
Contents Placeholder	Web Page	3D Scene
Navigation	By hyperlinks	Fly, walk, jump, etc.
Interaction	Click on contents, Enter text, Upload files	Touching, Clicking, Passing by, and Manipulation
Contents behaviour	Display, hide, and animate	3D object behaviour, such as simulate, rotate, move, etc.
Users Communication	Chat and talk	Chat, talk, and perform special activities

Table 4.1: Comparison between hypermedia and 3D Virtual Environment (Adapted From (Chittaro & Ranon, 2007a))

Firstly, content type for hypermedia applications can be text, images, video, and flash objects that are presented in a webpage. 3D VEs, on the other hand, have 3D representations for the virtual objects that populate the 3D scene. Furthermore, properties of content elements in both environments are different. In hypermedia, the text style, colour, font, and size are the main properties that can be adjusted. For 3D VEs, the 3D objects have a texture, size, position, orientation, and colour that could be adapted.

Secondly, the presentation container in hypermedia environments is a webpage, whereas in a 3D VE it is a virtual scene. This scene (i.e. a 3D space) also has properties such as lights and viewpoints, and possibly also some physical characteristics (like gravity). Such characteristics increase the possibilities of what can be adapted and adjusted in the 3D virtual environments.

Thirdly, another different aspect is the navigation. In hypermedia, it is simply done by clicking on hyperlinks, whereas users in 3D VEs will be able to navigate through the 3D space in different ways, e.g., by walking, flying, or jumping.

Fourthly, a user can interact with hypermedia environments by following hyperlinks or filling in forms. For a 3D VE, the user can also interact with 3D objects by touching, clicking, passing by, or manipulation the 3D object (e.g., drag and drop, resize, turn). Moreover, a user can simulate tasks as driving a car in a training program.

Fifthly, another important difference is concerned with behaviours for 3D objects. Some objects can have behaviours that are triggered by specific actions performed by the user or by other objects in the 3D VE, or by events that occur. Example behaviours are for instance changing position, colour, size, or start a rotation or movement.

Finally, users can communicate with each other in hypermedia environments by chatting and talking, whilst it could be more complicated in a 3D VE, for instance avatars can be used for the communication, or users can collaborate to perform 3D activities to accomplish certain tasks.

4.3 Adaptation Techniques for 3D VLE

Since the learning process is driven by the learner's navigation and interaction in both hypermedia/hypertext systems and 3D VLE (Chittaro & Ranon, 2007b), adaptive techniques from the context of hypermedia (i.e. Brusilovsky's adaptation taxonomy) could be extended or adapted to be applied for adaptive 3D VLEs. However, we prefer to use the 3D VE anatomy as starting point for our systematic overview, as this allows us to systematically consider what can be adapted in a 3D VLE and subsequently how.

In principle, adaptation can happen for each of the components of a 3D VLE. An adaptation can be limited to a single component of the 3D VLE, but it can also involve many different components of the 3D VLE. We will first describe adaptations that apply to single components, i.e. possible adaptations for objects, for behaviours, for interaction, and for navigation. We call these *adaptation types*. For the moment, we did not consider any adaptations types yet for the scene or for sound. Adapting certain properties of the scene should be done with care as it may affect many other components. The added value of adapting the scene in the context of education is also not investigated thoroughly. Sound can be done in the same way as adapting behaviour (actually we can consider sound to be a kind of behaviour of an object or of the scene). Communication is also not considered because we focus on single-user 3D VLEs.

Next, we will consider more high-level adaptations that involve more than one component. These kinds of adaptations we called *adaptation strategies*.

Finally, we also introduce *adaptation themes* that are even more high-level specifications of adaptation actions. An adaptation theme is a collection of adaptation types and strategies.

The following sections will discuss the proposed adaptation types, adaptation strategies, and adaptation themes and illustrate different situations in which they can be used to realize different objectives. Where relevant, we will make the link with existing hypermedia adaptation types.

4.3.1 Adaptation Types

Adaptation types are adaptations that apply on a single component. Different components allow for different adaptations. As a result, we have defined adaptation types for 3D virtual objects, for 3D object behaviours, for user interaction, and for navigation.

Adaptation Types for Objects

3D virtual objects populating the 3D VLE have a visual appearance in term of geometry (shape) and material properties (colours and textures). To enhance the usability of a 3D VLE for a learner, it may be advisable to change the visual appearance of an object during the lifetime of the 3D 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 a concept being represented in the 3D VLE, the visual appearance of the object can 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 category 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. We have identified four adaptation types to adapt the visualization of an object and two adaptation types concerned with hiding an object:

– ***semiDisplay***
<u>Description</u>*:* this adaptation type is used to display the object in a semi-manner, by having a semi-transparent bounding box around it. See figure 4.2(a) for an example: the Sun has been semi-hided. Such adaptation type is rather specific for the context of 3D Virtual Environments.

<u>Application</u>*:* this adaptation type can be used to draw learners' attentions towards the semi displayed 3D objects.

The *semiDisplay* adaptation type is defined as follows:
<semiDisplay> ::= semiDisplay <objectId> <bboxsize <colour> <transparency>
Where

<objectId>:	Id of the object on which the semi-transparent bounding box should be applied
<bboxsize>:	the size of the bounding box that should surround the object
<colour>:	the colour of the bounding box
<transparency>:	the transparency ranging from 0 to 1 where 0 is no transparency and 1 is completely transparent

– ***changeSize***:
<u>Description</u>: this adaptation type is used to change the visual appearance of an object by changing its size. In adaptive hypermedia, this adaptation type is considered as a way to annotate a hyperlink or text. Furthermore, this adaptation technique is also proposed in the context of 3D VE, e.g., in (Aquino et al., 2005).

<u>Application</u>: this adaptation type can be used to change the size of the 3D models, for instance for attracting the attention or depending on the learner knowledge.

The *changeSize* adaptation type is defined as follows:
<changeSize> ::= changeSize <objectId> <size>
Where

<objectId>:	id of the object for which the size needs to be changed
<size>:	the new size of the object

– ***changeMaterialProperties***:
<u>Description:</u> this adaptation type is used to change the material properties (colour or texture) of an object. Figure 4.2(b) shows an illustration of changing the texture of the sun. This could be compared to some extent to the *explanation-variant* adaptation technique for adaptive hypermedia systems (see section 2.3.3) in the sense that different explanations about a

specific concept or topic can be provided depending on the learner knowledge. It is also similar to the change level of details technique as presented in (Moody, 2009; Aquino et al., 2005). However, our change material properties adaptation type is used only to change the current colour of the 3D object or the associated texture map.

<u>*Application:*</u> this adaptation type can be used to indicate the status of the knowledge about the object or provide a more detailed texture for the 3D object when the learner is ready to learn about it.

The *changeMaterialProperties* adaptation type is defined as follows:

<changeMaterialProperty> ::= changeMaterialProperties <objectId> (colour <colour> | texture <texture>)

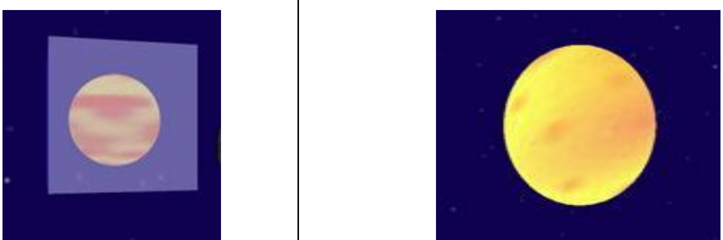
Where

<objectId>: id of the object for which the material properties needs to be changed

<colour>: the new colour of the object

<texture>: a new texture of the object

Note: the symbol "| "means "or ".



(a) after semiDisplay | (b)after changeMaterialProperties

Figure 4.2: Different visual representation for Sun

– ***changeVRRepresentation***:
<u>*Description*</u>: this adaptation type is used to change the visual representation of an object completely; its current visualization will be replaced by a different representation. This could also be compared to some extent with the *explanation-variant* adaptation technique in the adaptive hypermedia systems (see section 2.3.3), in the sense that a different explanation about a specific concept or topic can be provided depending on the learner knowledge. It is also close to changing level of details as presented in (Moody, 2009; Aquino et al., 2005). However, our changeVRRepresentation adaptation technique is only used to replace the current 3D representation for the concept with another 3D representation.

Application: this adaptation type can be used to introduce a more complex representation for the 3D model when the learner is ready to learn about it.

The *changeVRRepresentation* adaptation type is defined as follows:
<changeVRRepresentation> ::= changeVRRepresentation <objectId> <representation>

Where

<objectId>: ID of the object for which its representation needs to be changed.

<representation>: the new representation of that VR object expressed in X3D format.

– ***semiHide***:
Description: this adaptation type allows the visual appearance of objects to be semi-hidden by means of a semi-transparent bounding box.

Application: this is similar to the semiDisplay adaptation type but the purpose is slightly different, i.e. to hide the object in a semi-manner instead of displaying it in a semi-manner. This adaptation type can be used to semi hide 3D objects, which are not prominently required in the current learning process.

The *semiHide* adaptation type is defined as follows:

<semiHide> ::= semiHide <objectId> <bboxsize> <colour> <transparency>

Where

<objectId>: id of the VR object on which the semi-transparent bounding box needs to be applied

<bboxsize>: the size of the bounding box surrounding the object

<colour>: the colour of the bounding box

<transparency>: the transparency ranging from 0 to 0.5 where 0 is no transparency and 0.5 is the maximum of transparency allowed for semi-hidden.

– ***hide****:*
Description: this adaptation type allows hiding an object visually. Also in the context of adaptive hypermedia, this adaptation technique exists. Brusilovsky proposed hiding as a kind of adaptive navigation support. Also in the context of 3D VE hiding was already proposed (Chittaro & Ranon, 2007c, 2007b; Dos Santos & Osório, 2004b) as an adaptation technique to prevent learners from being involved in learning too many concepts at the same time.

Application: this adaptation type can be used to hide all information that is not relevant at a certain moment in the learning process and/or might confuse the learner.

The *hide* adaptation type is defined as follows:
<hide> ::= hide <objectId>
Where
<objectId>: id of the object that needs to be hidden

– ***display**:*
Description: this adaptation type allows displaying an object that has been hidden before. Similar to the previous adaptation type (hide), this technique is also proposed in the context of adaptive hypermedia, as well as for adaptive 3D VE.

Application: this adaptation technique can be used to show a new 3D object according to the current learner knowledge.

The *display* adaptation type is defined in a similar way as hide:
<display> ::= display <objectId>
Where
<objectId>: id of the object that needs to be displayed

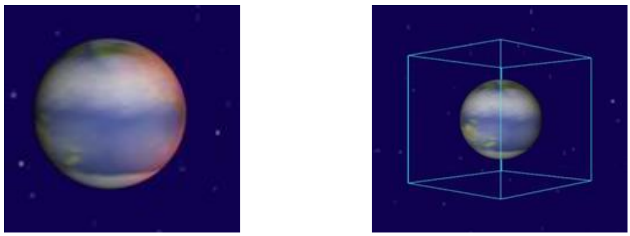
Furthermore, like in classical textual learning material, it may be useful to mark objects. A reason for marking an object is for instance to draw the attention of the learner that the material associated with the object has (or has not yet) been studied or to indicate the importance of the object. In 3D VLE, marking an object can be done in different ways. We distinguish two different adaptation types for marking because they are essentially different: spotlight and highlight.

– ***spotlight***:
Description: this adaptation type allows marking a 3D object by putting a spotlight on the 3D object. In principle, this is also used in adaptive hypermedia where the text can be annotated with a specific colour. However, spotlight has different properties that can be customized, i.e. colour, insensitivity, and angle.

Application: this adaptation type can be used to draw the attention of the learner to an object. Figure 4.3(a) shows an illustration of the planet Earth being highlighted with a red spotlight.

The *spotlight* adaptation type is defined as follows:
<spotlight> ::= spotlight <objectId> <colourRay> <angle>
Where

<objectId>: id of the object on which a spotlight needs to be applied
<colourRay>: the colour of the ray
<angle>: the angle of the spotlight according to the object subject of the spotlight



(a) spotlight (b) highlight

Figure 4.3 Earth with spotlight and highlight techniques

– ***highlight***:
<u>*Description*</u> this adaptation type allows to mark an object by drawing a box around the object, where only the edges of the box are displayed. Figure 4.3(b) shows the planet Earth being highlighted.

<u>*Application*</u>: this adaptation type can be used for the same purpose as the previous adaptation type (spotlight), i.e. to mark important 3D objects.

The *highlight* adaptation type is defined as follows:
<highlight> ::= highlight <objectId> <bboxsize> <colourEdge>
Where
<objectId>: id of the object on which the semi-transparent bounding box needs to be applied.
<bboxsize>: the size of the bounding box surrounding the object.
<colourEdge>: the colour of the edges of the box

Note that marking is considered different from changing the material properties (like the colour) of an object, as by marking we do not 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 annotations described next.

In a 3D VLE, it may also be useful to add annotations to objects. Annotation can be used not only to attach explanations, comments, or names to objects but also to add e.g., an audio explanation, video, another 3D object, or a web-link. Annotations should only be shown when appropriate for the learning process as otherwise they might clutter the 3D VLE too much. In addition, it may be necessary to adapt the annotations to the profile of the learner. Therefore, we consider annotations as an adaptation type. The following adaptation types are identified for annotations:

- ***displayAnnotation***:
  <u>*Description:*</u> this adaptation type allows displaying an annotation with a 3D object. This adaptation technique is already explored in (Chittaro & Ranon, 2007b) using conditional fragments adaptation technique from the adaptive hypermedia domain. However, our adaptation type is not limited to drawing a bounding box as in (Chittaro & Ranon, 2007b). Furthermore, a 3D object can be annotated using textual information, but also with audio or video, even with another 3D object. An example about textual annotation to a 3D object representing earth is shown in Figure 4.4.

  <u>*Application:*</u> similar to the objective of annotation technique in adaptive hypermedia, this adaptation type can be used to support learner in their learning by providing him with some extra information or comments about the annotated 3D object.

  The *displayAnnotation* adaptation type is defined as follows:
  <displayAnnotation> ::= displayAnnotation <objectId> <distance> <type> <value>
  Where
  <objectId>: id of the object for which an annotation needs to be displayed
  <distance>: the distance from the VR object to a point in space from where the annotation will be displayed
  <type>: the type of the associated annotation such as text, audio, video, etc.
  <value>: the resource to be used to annotate the 3D object (depending on the annotation type)

- ***hideAnnotation:***
  <u>*Description*</u>: this adaptation type allows hiding an annotation associated with a 3D object.

  <u>*Application*</u>: this adaptation type can be used to prevent having a cluttered 3D VLE.

  The *hideAnnotation* adaptation type is defined as follows:

  <hideAnnotation> ::= hideAnnotation <objectId> ((<type>)+ | all)
  Where
  <objectId>: id of the object for which the annotation needs to be hidden.
  <type>: the annotation type to hide (such as text, audio, video, etc).

Figure 4.4: displayAnnotation

Adaptation Types for Behaviours

Behaviours are used to make the 3D VLE dynamic, i.e. to create environments where objects are active by performing some behaviour. For instance, in a 3D VLE for a solar system we can have planets that are rotating and comets that move through the universe. However, to guide the students during his learning it may be useful to disable and enable behaviours when appropriate, e.g., a solar system where all planets are rotating at the same time may be very confusing for a beginner. It may also be useful to adapt the parameters of a behaviour, for instance to show the behaviour of the sun with a different value for its temperature. Possible adaptation types for behaviours are:

– ***enableBehaviour****:*

Description: this adaptation type allows enabling behaviours associated with an object. This adaptation type is typically for the context of 3D VEs.

Application: as mentioned earlier, to avoid confusing the learner by showing all behaviours of all 3D objects populating the virtual world, this adaptation type can be used to enable only a specific number of 3D object behaviours, for instance depending on the learner knowledge.

The *enableBehaviour* adaptation type is defined as follows:

<enableBehaviour> ::= enableBehaviour <objectId> (<behaviourId> <triggerOn>)$^+$

Where

<objectId>: id of the object for which the behaviour(s) need to be enabled.

<behaviourId>: id of the behaviour to be enabled.

<triggerOn>: the instruction to turn on the behaviour.

Note: the symbol "$^+$ " means one to more.

– ***disableBehaviour****:*

Description: this adaptation type allows disabling behaviours associated with a 3D object. This adaptation type is again typical for the context of 3D VEs.

Application: this adaptation type can be used to disable already running behaviours of a 3D object, in order to allow learners focusing on other aspects.

The *disableBehaviour* adaptation type is defined in a similar way as enableBehaviour:

<disableBehaviour> ::= disableBehaviour <objectId> (<behaviourId> <triggerOff>)+

Where

<objectId>: id of the object for which the behaviour(s) need to be disabled

<behaviourId>: id of the behaviour to be disabled

<triggerOff>: the instruction to turn off the behaviour

– ***changeBehaviour***:
Description: this adaptation type allows changing a behaviour by modifying the values of its parameters.

Application: this adaptation type can be used to change speed (or other properties) of a 3D object's behaviour so it becomes faster or slower depending on specific criteria to enable learner to observe specific aspects of the behaviour.

The *changeBehaviour* adaptation type is defined as follows:

<changeBehaviour> ::= changeBehaviour <objectId> (<behaviourId> <changeInstruction>)+

Where

<objectId>: id of the object for which the behaviour needs to be changed.

<behaviourId>: id of the behaviour to be changed.

<change-Instruction>: the instruction to change the parameters of a behaviour.

Adaptation Types for User Interactions

In a 3D VLE, there are different ways to interact with a 3D object, e.g., by clicking on an object, by touching an object, by passing closed by an object. Furthermore, user interaction can trigger the start (or end) of behaviour. In the context of a 3D VLE, it may also 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 can also be used as a kind of reward after some successful study, and disabling interaction after some time can avoid that the learner is spending too much time with some appealing features.

Adapting interaction comes down to *enabling* or *disabling* the possible types of interaction provided for an object. Possible interaction types considered for objects are: touching, clicking, passing by, and manipulation:

– ***enableInteraction***:
  Description: this adaptation type allows enabling an interaction type (given as parameter) for an object. To some extent this can be compared to the adaptive technique of showing links in hypermedia when appropriate for the learner.

  Application: as mentioned earlier, this adaptation type can be used as a kind of reward after some successful study so that the learner will be able to interact with new 3D objects. It can also be used to enable the learner interacting with specific 3D models to acquire more knowledge.

  The *enableInteraction* adaptation type is defined in a similar way as enableBehaviour:
  <enableInteraction> ::= enableInteraction <objectId> (<interactionType> <triggerOn>)+
  Where
  <objectId>: id of the object for which the interaction type(s) need to be enabled.
  <interactionType>: type of interaction to be enabled.
  <triggerOn>: the instruction to turn on the interaction type.

– ***disableInteraction***:
  Description this adaptation type allows disabling an interaction type (given as parameter) for a 3D object. Similar as for enablingInteraction, to some extent this adaptation technique can be compared to the adaptive technique of hiding irrelevant or non-suitable links in hypermedia.

  Application: It can be used to disable interaction after some time to avoid that the learner is spending too much time with some appealing features. It can also be used to prevent the learner from being overloaded with stimuli by prohibiting interaction with a 3D object as long as the learner has not obtained a certain level of knowledge.

  The *disableInteraction* adaptation type is defined in a similar way as disableBehaviour:

  <disableInteraction> ::= disableInteraction <objectId> (<interactionType> <triggerOff>)+
  Where
  <objectId>: id of the object for which the interaction type(s) need to be disabled.

<interactionType>: type of interaction to be disabled.
<triggerOff>: the instruction to turn off the interaction type.

Adaptation Types for Navigation (Avatars)

Avatars play a special role in 3D VLEs. An avatar can be used to provide the view of the user inside the 3D VE, i.e. the view follows the 3D object that represents the avatar. 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 (see section 4.1). Furthermore, an avatar has behaviours like *Jumping* and *Walking* that allow the avatar to progress inside the virtual environment (i.e. navigate). Jumping means that user will navigate from one point to the next by changing the avatar's position directly and without following a specific path. Walking means going from one point to the next smoothly (without jumping) and by following a certain path. Note that other behaviours are also possible like flying. It may be useful to be able to change the visual representation, as well as the behaviours of an avatar, for instance based on the learning style of the learner or his preferences, or to keep the learner engaged. .

The representation and/or the behaviour to be used for an avatar can be changed as follows:

<avatar> ::= Avatar <avatarId> <visualRepresentation> <navigationBehaviour>

Where

<avatarId>: unique id identifying the avatar
<visual-Representation>: the geometry with the material properties (express in X3D format) that should be used to represent the avatar
<navigation-Behaviour>: behaviour to be used for navigation

The following figure depicts all the proposed adaptation types. Together they cover a large range of possible adaptions types for 3D VEs. Note however, that adaptation types for the virtual scene, sound, and for communication (collaborative environments) are not considered. Although, we tried to be as complete as possible for the adaptation types concerning 3D objects, behaviours, navigation, and interaction, it is always possible to come up with new adaptation types. For instance, for interaction it is possible that new interaction devices, such as devices based on gesture recognition or digital pens, will allow introducing new adaptation types. However, the use of the 3D VE anatomy as the underlying principle for defining the adaptation types, will allow easily extending the given set of adaptation types.

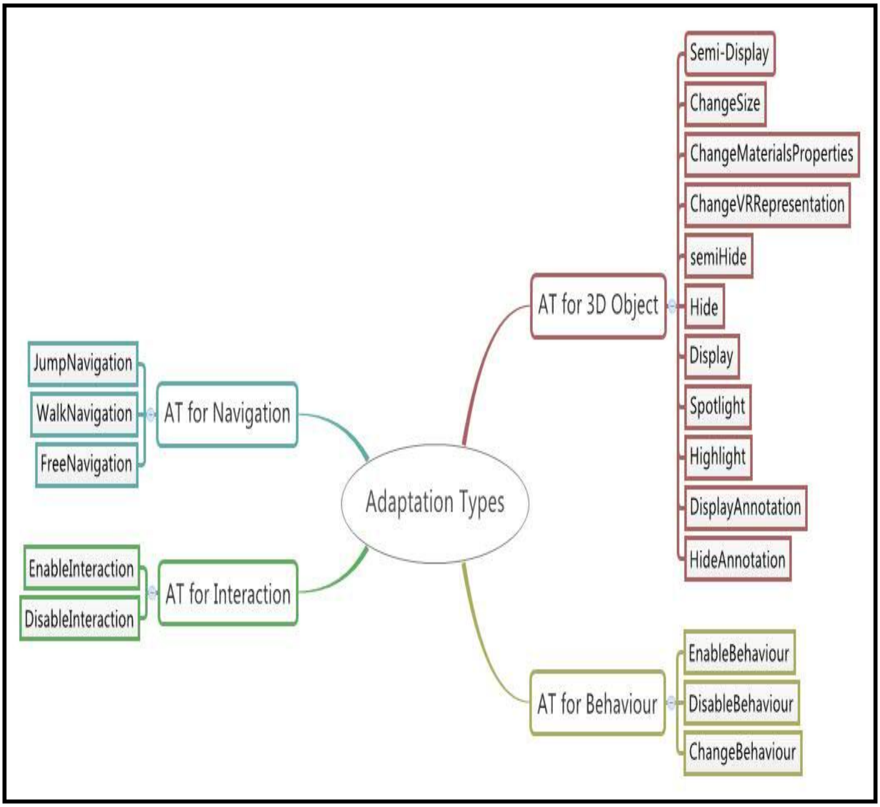


Figure 4.5: Set of Adaptation Types.

4.3.2 Adaptation Strategies

In the previous section, we have discussed possible adaptations for individual components (object, behaviour, interaction) of a 3D VLE. In this section, we deal with adaptations that go beyond the adaptation of a single component. These adaptations have an impact on several components of the 3D VLE at the same time or on a part of the 3D VLE. We call them *adaptation strategies*, as they can be used as strategies to adapt a 3D VLE. In this thesis, we focus in particular on adaptation strategies for learning purposes.

The adaptation strategies presented here are only a subset of possible adaptation strategies (De Troyer et al., 2009). It is of course possible to define much more adaptation strategies or to define strategies that are a combination of other 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 3D VLEs. It is of course always possible to define a new adaptation strategy if there is a need for it.

A first group of adaptation strategies defined are those that will have an impact on how the learner can navigate through the 3D VLE:

– ***restrictedNavigation**:*
<u>*Description*</u>: this adaptation strategy allows restricting the navigation towards 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 from a sequence of objects. Furthermore, the navigation can be restricted to a particular type of navigation e.g., jumping or waking. With jumping, the avatar will go from one object to the next without viewing the part of the scene between the two objects. In general, the *restrictedNavigation* adaptation strategy is similar to the hiding adaptation technique (from Brusilovsky's Taxonomy) in adaptive hypermedia systems in the sense that hypermedia systems can simply disable the hyperlinks to inappropriate or inconvenient webpages. However, in the context of 3D VLE, navigation is done differently by specifying avatars' position and orientation (Hughes et al., 2002).

<u>*Application*</u>: to specify the list of objects allowed to navigate to, different criteria can be used. In the context of a 3D VLE, it is important to allow using pedagogical criteria. For instance, it can be useful to select objects based on the learner's knowledge, or based on some pedagogical relations (such as pre-requisite) between the corresponding learning concepts. In general, this adaptation strategy can be used to force a learner to visit only a pre-defined number of objects in the scene.

The *restrictedNavigation* adaptation strategy is defined as follow:
<restrictedNavigation> ::= restrictedNavigation <avatar> (all | (<objectId>)$^{+}$ *) where <condition>*

Where

<avatar>:	specifies the avatar with its representation and navigation behaviour (see section 4.3.1)
<objectId>:	id of the object potentially allowed to visit
<condition>:	specifies the condition that the objects needs to satisfy to be in the list of objects to visit. If "all" is used, the condition is applied on all objects in the 3D VLE, otherwise on the objects listed

– ***navigationWithRestrictedBehaviour*:**
<u>*Description*</u>: this adaptation strategy allows restricting the possible behaviours of objects while navigating. The restricted behaviour can apply on all objects (in the 3D VLE) or on a specified list of objects. This adaptation strategy is typical for the context of 3D virtual environments.

<u>*Application:*</u> similar as for *restrictedNavigation*, the possibility to use pedagogical criteria is important to specify the list of objects. This strategy is for instance useful to allow a learner to first explore a 3D VLE (or a part of it) without being annoyed by objects showing all kinds of behaviour; afterwards when the learner is more familiar with the 3D, VLE behaviour can be enabled (using another adaption strategy).

The *navigationWithRestrictedBehaviour* adaptation strategy is defined as follow:

<navigationWithRestrictedBehaviour>::=navigationWithRestrictedBehaviour<avatar> (all | (<disableBehaviour>)$^{+}$ *) where <condition>*

Where

<avatar>: specifies the avatar with its representation and navigation behaviour (see section 4.3.1)

<disable-Behaviour>: specifies the behaviour to be disabled (see section 4.3.1)

<condition>: specifies the condition that the objects needs to satisfy. If "all" is used, the condition is applied on all objects in the VLE and the behaviours of all objects satisfying the condition will be disabled, otherwise the objects are specified by means of <*disableBehaviour*>. The behaviour of these objects will only be disabled when the condition is true.

– ***navigationWithRestrictedInteraction***:

<u>*Description*</u>: this adaptation strategy allows restricting the possible interactions with some objects while navigating. Also here, the possibility to use pedagogical criteria is important to specify the list of objects for which the restriction applies. Also this adaptation strategy is typical for 3D VLE.

<u>*Application*</u>: Similar as for *navigationWithRestrictedBehaviour*, this adaptation strategy can be used to allow a learner to first explore the 3D VLE (or a part of it) without being able to fully interact with the objects in the scene; afterwards when he is more familiar with the 3D VLE, interactions can be enabled (using another adaption strategy). This adaptation strategy is defined in a similar way as *navigationWithRestrictedBehaviour:*

The *navigationWithRestrictedInteraction* adaptation strategy is defined as follow:

<navigationWithRestrictedInteraction>::=navigationWithRestrictedInteraction <avatar> (all | (<disableInteraction>)$^{+}$) where <condition>

Where

<*avatar*>: specifies the avatar with its representation and navigation behaviour (see section 4.3.1)

<*disable-Interaction*>:	specifies the interaction to be disabled (see section 4.3.1)
<*condition*>:	specifies the condition that the objects needs to satisfy. If "all" is used, the condition is applied on all objects in the VLE and all interaction of all objects satisfying the condition will be disabled, otherwise the objects and their interaction type(s) are specified by means of <disableInteraction>. The interaction type of these objects will only be disabled when the condition is true.

– ***TourGuide***:
Description: this strategy can be used to provide a tour guide to the learner. A tour guide takes the learner through a tour in a 3D VLE using a predefined navigation path. This adaptation strategy is similar to the direct-guidance adaptation technique proposed in (Hughes et al., 2002). However, *TourGuide* adaptation strategy is more oriented toward navigation to a group of 3D objects rather than changing camera orientation viewpoint for the same 3D object as done in the direct-guidance adaptation technique.

Application: 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 3D VLE.

The *TourGuide* adaptation strategy is defined as follow:
<tourGuide> ::= tourGuide <avatar> <path>
Where

<avatar>:	specifies the avatar used with its representation and navigation behaviour (see section 4.3.1).
<path>:	specifies the different positions inside the virtual environment that form the path to be followed by the avatar.

The following adaptation strategies allow specifying that the learner can navigate freely in the 3D VLE. However, we distinguish two types of free navigation strategies, namely *completelyFree* and *freewithSuggestions*.

– ***completelyFree***:
Description this adaptation strategy will allow the learner to navigate freely in the 3D VLE.

Application: for instance, this strategy can be used in exploration-based learning.

The *completelyFree* adaptation strategy is defined as follow:
<completelyFree> ::= completelyFree <avatar>

Where
<avatar>: specifies the avatar with its representation and navigation behaviour.

– ***freeWithSuggestions***:
Description: this adaptation strategy will allow the learner to navigate freely in the 3D VLE but in addition some objects will be "suggested". Suggesting is done by using marking (i.e. spotlight or highlight) or annotations. This technique is also similar to the sorting adaptation technique (Hughes et al., 2002) in the sense that both techniques suggest navigation towards interesting information with the possibility of ignoring the suggestions. Such an adaptation technique is also investigated in (Mackinlay et al., 1990).

Application: also here, pedagogical criteria should be considered to specify the list of objects to suggest. This strategy can be used to give the learner a lot of freedom but still provide some guidance.

The *freeWithSuggestions* adaptation strategy is defined as follow:
<freeWithSuggestions> ::= freeWithSuggestions <avatar>
(<highlight> | <spotlight> | <displayAnnotation>)$^+$ where <condition>
Where
<avatar>: specifies the avatar used with its representation and navigation behaviour (see section 4.3.1).
For <higlight>, <spotlight>, and <displayAnnotation> see section 4.3.1.
<condition>: specifies the condition that the objects need to satisfy.

A second group of adaptation strategies are those that allow adapting a group of objects:

– ***filterObjects***:
Description: this adaptation strategy allows filtering the objects that should be available (visible) in the 3D VLE. It is similar to the hiding adaptation technique in the context of adaptive hypermedia and in 3D VE as presented in (Hughes et al., 2002). However, this adaptation strategy is focusing on populating the 3D VLE with 3D objects instead of supporting the navigation to certain 3D objects as in the *restrictedNavigation* adaptation strategy.

Application: the filtering is possible according to pedagogical criteria. This strategy can be used to avoid that the learner get lost in the 3D VLE or that he doesn't know on which objects to focus first. It also allows gradually building the 3D VLE; the more knowledge the learner obtains the more objects could become visible.

The *filterObjects* adaptation strategy is defined as follow:
<filterObjects> ::= filterObjects (<display>)$^+$ where <condition>

For <display> see section 4.3.1; <condition> has its usually meaning. .

- ***markObjects***:
  Description: this adaptation strategy allows marking (i.e. by highlight or spotlight) a number of objects in the 3D VLE. The marking will be done for all objects satisfying certain (pedagogical) criteria. In principle, this is similar to the annotation adaptation technique in the context of adaptive hypermedia and the annotation adaptation technique proposed in (Satalich, G., 1995; Hughes et al., 2002).

  Application: 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 3D VLE as not attractive anymore. The strategy can also be used to mark the objects already studied. markObjects is defined in a similar way as filterObjects.

  The *markObjects* adaptation strategy is defined as follow:
  <markObjects>::= markObjects (<spotlight> | <highlight>)+ where <condition>
  Where <spotlight> and <highlight> as defined in section 4.3.1; <condition> has its usually meaning.

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

- ***displayAtMost***:
  Description: this adaptation strategy allows specifying (based on some condition) when some objects should not be displayed anymore. In principle, this technique is also applicable in the context of adaptive hypermedia where the learning materials can be displayed a specific number of times and then be hidden.

  Application: the condition can be given by means of pedagogical criteria like the knowledge level that the learner currently has for the object or by setting a limit on the number of times the object should be displayed. This last case can for instance be useful if the purpose is to let the learner do some tests in the 3D VLE without having the subject(s) of the study visible.

  The *displayAtMost* adaptation strategy is defined as follow:
  <displayAtMost> ::= displayAtMost (<hide >)$^{+}$ where <condition >
  Where <hide> as defined in section 4.3.1; <condition> has its usually meaning.

– ***displayAfter***:
<u>*Description*</u>: This adaptation strategy allows specifying the condition(s) that need to be satisfied for objects to be displayed. This is also applicable in the context of adaptive hypermedia systems.

<u>*Application*</u>: The condition can be given by means of pedagogical criteria or by means of a time criterion. This adaptation strategy can for instance be used to keep the 3D VLE appealing by dynamically changing some of the objects in the scene. This can avoid that the learner gets bored.

The *displayAfter* adaptation strategy is defined as follow:
<displayAfter> ::= displayAfter (<display>)$^+$ where <condition>
Where <display> as defined in section 4.3.1; <condition> has its usually meaning.

The next group of adaptation strategies consists of strategies for adapting the behaviour of some objects conditionally. As for the previous group of adaptations, the conditions can be pedagogical criteria or time criteria:

– ***behaviourAtMost***:
<u>*Description:*</u> this adaptation strategy allows specifying when (by means of a condition) behaviour should be disabled. This adaptation strategy is only applicable in the context of 3D VEs.

<u>*Application:*</u> 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 behaviour).

The *behaviourAtMost* adaptation strategy is defined as follow:
<behaviourAtMost>::=behaviourAtMost (<disableBehaviour>)$^+$ where <condition>
Where <disableBehaviour> as defined in section 4.3.1; <condition> has its usually meaning.

– ***behaviourAfter***:
<u>*Description*</u>: this adaptation strategy allows indicating when (by means of a condition) certain behaviour of an object (or some objects) should be executed. This is an adaptation strategy that is typical for 3D VLEs.

<u>*Application*</u>: this adaptation strategy can for instance be used to state that the behaviour for some specific objects should only start when the knowledge level of the learner for the corresponding learning concepts is above a certain threshold.

The *behaviourAfter* adaptation strategy is defined in a similar way as behaviourAtMost:

<behaviourAfter> ::= behaviourAfter (<enableBehaviour>)$^+$ where <condition>

Where <enableBehaviour> as defined in section 4.3.1; <condition> has its usually meaning.

– ***behaviourSpeed***:

<u>*Description:*</u> this adaptation strategy allows specifying the speed of a behaviour associated with some objects. This adaptation strategy is only applicable in the context of 3D VLEs.

<u>*Application*</u>: this adaptation strategy is in particular useful when the behaviour simulates real world behaviour. Being able to slow down the behaviour will allow the learner to better observe what is happing, especially for behaviours that are happing very fast in real time (e.g., an explosion, or the trajectory followed by a bullet).

The *behaviourSpeed* adaptation strategy is defined as follow:

<behaviourSpeed> ::= BehaviourSpeed (<changeBehaviour>)$^+$

Where

<changeBehaviour>: specifies the behaviour for which the speed has to be changed (see section 4.3). The change should be limited to a change of the speed.

The last group of adaptation strategies contains strategies for limiting the interaction with some objects by means of a condition. As for the previous two groups of adaptations, the conditions can be given by means of pedagogical criteria or by means of a time criterion. Their definition is similar as for behaviourAtMost and behaviourAfter:

– ***interactionAtMos*t**:

<u>*Description:*</u> this adaptation strategy allows specifying when the learner cannot interact with the objects anymore. In principle, this adaptation strategy can also be applied in the context of adaptive hypermedia to prevent learner from visiting resources for a learning concept more than a specific time.

<u>*Application*</u>: similar as for the behaviourAtMost strategy, this adaptation strategy can for instance be used to avoid that the learner keeps "playing" with objects.

The *interactAtMost* adaptation strategy is defined as follow:

<interactAtMost> ::= interactAtMost (<disableInteraction>)$^+$ where <condition>

Where <disableInteraction> as defined in section 4.3.1; <condition> has its usually meaning.

– ***interactionAfter***:
<u>*Description*</u>***:*** this adaptation strategy allows specifying when (by means of a condition) the learner can interact with some 3D objects. This is also can be used in the context of adaptive hypermedia.

<u>*Application*</u>: This adaptation strategy can be used to ensure that the learner has enough knowledge about the objects before he can interact with it.

The *interactionAfter* adaptation strategy is defined as follow:
<interactionAfter> ::= interactionAfter (<enableInteraction>)$^+$ where <condition>
Where <enableInteraction> as defined in section 4.3.1; <condition> has its usually meaning.

Next, Figure 4.6 depicts all the proposed adaptation strategies.

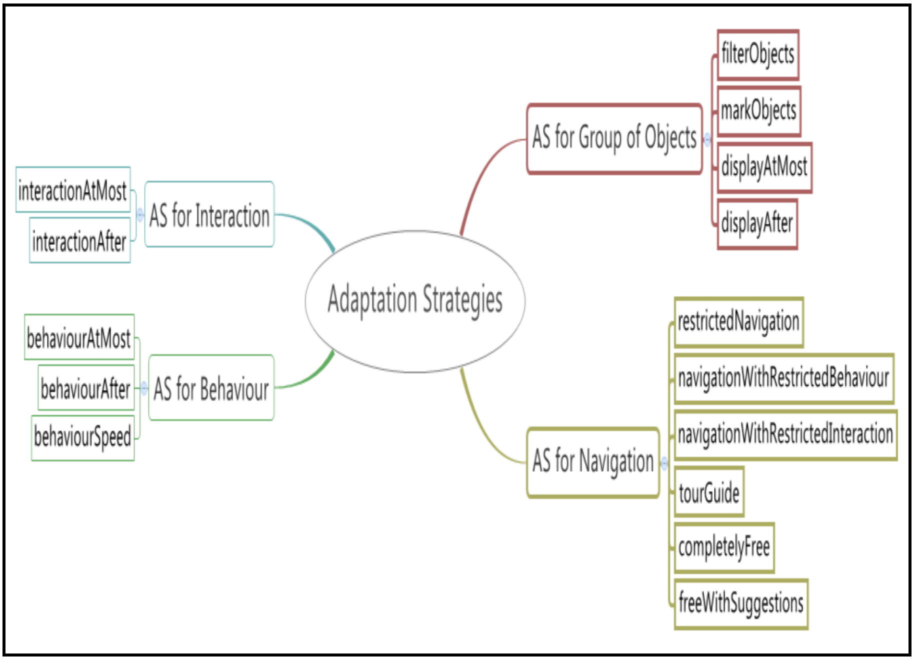


Figure 4.6: Adaptation Strategies.

4.3.3 Adaptation Theme

This section presents the concept of adaptation theme that aims to provide a meaningful (combination of) instantiations of adaptation types and strategies. In other words, an adaptation theme is defined as a collection of specific adaptation types and adaptation strategies. It can be compared with the concept of a theme used in a word processor. For example, it allows specifying that marking 3D virtual object should always be done using the highlighting adaptation type and with a fixed colour for the bounding boxes.

A possible use of the adaptation themes is to associate them with specific classes or groups of users. E.g., beginner learners can have a different adaptation theme than advanced learners. It is helpful for authors who like to use/define a specific adaptation style to fit the targeted users' needs. Moreover, by using an adaptation theme, authors will be able to have a consistent adaptation style over the complete adaptation process. Otherwise, it might be difficult for an author to achieve consistent specifications for the same kinds of adaptation in the overall adaptation process. For instance, an adaptation theme can avoid having different colours for bounding boxes with the same purpose.

In an authoring tool (see Chapter 9), authors should be able to specify the adaptation theme before starting specifying adaptations or they can reuse existing adaptation themes.

4.4 Summary

As in classical text-based learning environments, adaptivity can used to adapt a 3D 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, based on previous work, particularly (Brusilovsky, 1996; Chittaro & Ranon, 2007a), we have provided an overview of the differences between the different aspects related to hypertext and hypermedia systems and 3D Virtual Environments.

As a basis for proposing adaptation techniques for 3D VLEs, we decided to base our classification on the different components of a 3D VLE on which adaptation techniques can be applied. Therefore, the chapter started with defining the 3D VLE anatomy which is composed of seven components; *virtual scene, 3D virtual object, behaviour, user interaction, navigation, communication, and sound*.

Based on the 3D VE anatomy, we have proposed a set of adaptation types for 3D VLE, as well as a set of adaptation strategies specific for adapting 3D VLE according to the learner profile (i.e., knowledge, background, and behaviour). Adaptation types refer to the possible adaptations techniques that can be applied on a single component of a 3D VLE, i.e. possible adaptations for objects, for behaviours, for interaction, and for navigation. An example of such an adaptation type for a 3D object is for instance *changeMaterialProperties* that allows changing the colour and/or texture of a 3D object. Adaptation strategies refer to possible adaptation techniques that go beyond the adaptation of a single component, i.e., they refer to possible adaptations techniques that have an impact on several components of the 3D VLE at the same time. An example of such an adaptation strategiy is *restrictedNavigation* that restricts how the user can navigate through the 3D space.

Finally, the chapter ended by presenting the concept of an adaptation theme. An adaptation theme is composed of a customized set of adaptation types and strategies. The adaptation theme is used to allow for a coherent and consistent adaptation throughout the complete 3D VLE, and can, for instance, be defined for different group of learners: beginner, intermediate, and advanced learners.

Chapter 5: A Conceptual Framework for Adaptive 3D VLEs

The previous chapter presented different adaptation types, strategies, and themes that can be used for designing adaptive 3D VLEs.

The main goal of this chapter is to present a conceptual framework to support adaptation of 3D VLEs. The conceptual framework allows defining what should be adapted inside the 3D VLE, when, and how the components of 3D VLE should be adapted. This is realized by means of adaptation rules. In addition to that, the conceptual framework supports adaptation to happen on storyline aspects. In this way, learners will be provided with an adaptive storyline to achieve the required learning goals in a sequence adapted to their needs.

The remainder of this chapter is structured as follows: Section 5.1 explains and motivates the underlying approach, i.e. conceptual modelling. Section 5.2 explains the basic principles used for our approach and how we derived them. After that, section 5.3 presents our conceptual framework to support adaptation inside 3D VLE. Section 5.4 summarizes the chapter.

5.1 A Conceptual Modelling Approach for Adaptive 3D VLE

As already mentioned earlier in the problem statement (section 1.2), author-oriented approaches to specify adaptivity inside 3D VLE are still lacking. In this thesis, we propose a *conceptual modelling approach* to enable authors, educators usually not schooled in computer science, to specify adaptations in 3D VLEs.

By conceptual modelling we mean "the activity of formally describing some aspects of the physical and social world around us for purposes of understanding and communication." (Mylopoulos, 1992). Note that conceptual modelling is independent from any implementation, target technologies, programming languages, or any other physical considerations (Dillon & Tan, 1993). Both characteristics, being suitable for understanding and communication and being independent of any implementation technologies, make it a suitable approach for our purpose and goals.

To put it simply, our approach is a *model-based approach* meaning that high-level models (i.e. conceptual models) are used to describe the different aspects related to providing adaptation in 3D VLE. Besides, the models have sufficient details to enable the automatic generation of a full implementation.

In the rest of this section, we provide the basic principles formulated for our conceptual modelling approach. We also discuss their origin and provide motivations.

5.1.1 Principles for Designing Adaptive 3D VLEs

Based on our observation that it is vital to have coherent and clear principles for designing (and delivering) adaptive 3D VLE, we formulated a number of principles for our approach. Some of these principles are based on the literature reviewed in the context educational games (Amory et al., 1999; Pivec & Pivec, 2008; Annetta, 2010; De Byl, 2009) and others are inspired by our own work done in the context of the GRAPPLE project, which aimed at developing both authoring tools and a delivery environment for 3D VLEs.

- Different good principles have already been formulated by other researchers, more in particularly in the context of educational games. For instance, according to (Pivec & Pivec, 2008; Kiili, 2005b), pedagogical aspects need to be integrated with the story that defines the tasks that the learners should accomplish. This is revealed in educational and serious games (Amory & Seagram, 2003; Verpoorten et al., 2007). This leads to a first principle, being that *the pedagogical aspects should be clearly identified when designing an adaptive 3D VLE*.
- Furthermore, an interesting work, presented in (Carro et al., 2002), proposed a methodology that establishes a set of steps for an educational game environment design process. The work presents interesting aspects to be considered in our context. We have extracted them: "*determine the activities that are part of the environment", "organize them, for each type of user, in activity groups; specify the sequencing mode for these activities execution; and specify the prerequisites that can be established among them", "describe stories, where the game goals, activity feedbacks, agents and other multimedia elements can be included. These components are associated to the game structure components (activities and activity groups) and constitute altogether a story that can be independent from the concrete games presented to the users".* Although this work is done in the context of educational games (and it is not dealing with adaptivity), it provides a good basis for developing adaptive 3D VLEs. This provides a second principle: *deliver the content to the learner as different topics organized into a learning path, or so-called storyline.* This principle also emerged from the work done in GRAPPLE (but was not realized in the GRAPPLE project). It came up after conducting the evaluation and analysing the results (Ewais & De Troyer, 2013b). In GRAPPLE, the learning sequence is not specified by

the author; it is determined by the adaptive engine based on the pedagogical relationships between the learning concepts and the adaptation rules specified by the author. However, in a 3D VLE the learning objects already have some spatial relationships that authors may want to exploit during learning. Furthermore, providing a learning path may avoid some of the obstacles mentioned in the introduction, i.e. that the learner may get lost in the 3D environment, not knowing where to go next or what to do next. We believe that providing a learning path may increase the learning process in a 3D VLE, as it will help the learner to know what to study and to keep focus. It is generally known in pedagogy that a well-defined goal may increase the learners' motivation to complete his tasks, and it is in line with the work done in educational games where the concept of a storyline is also used. Moreover, *providing adaptivity inside the storyline* will allow adjusting the storyline (i.e. the learning path) to individual learners. Adaptive storylines will avoid providing the same storyline to all learners without considering their background or knowledge level.

In addition, some principles were already formulated for our work done in the GRAPPLE project (De Troyer et al., 2010, 2009):

- One of the important models in GRAPPLE is the Conceptual Adaptation Model (CAM) which 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 (Hendrix et al., 2008). In GRAPPLE, these adaptations are specified at the level of the learning concepts; which learning resources are used for a particular learning concept is not specified by the author but automatically derived by matching the needs of the learner with the meta-data associated with the learning resources. In the context of GRAPPLE, we already noticed that such an approach is not appropriate for specifying adaptation in 3D VLEs. *For 3D content, adaptation cannot (only) be specified at the level of the concepts but needs to be specified at the level of individual objects, behaviours, and interactions.* While in a classical text-oriented learning environment, it is easy to compose a page by combining different pieces of text or to adapt a page by replacing one piece of text by another; this is not straightforward in an adaptive 3D VLE. For instance, 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 3D VLE. They may be too large or too small or the texture used may not be appropriate. The same applies for behaviours. It is not always possible to replace one behaviour by another. Therefore, it is not possible (at least for the moment) to let the adaptive engine replace autonomously 3D/VR resources, and it is certainly not possible to let the adaptive engine compose the actual VLE automatically as all 3D virtual objects should fit together and need to be positioned and oriented in the 3D virtual space with care. It is even not possible to leave the selection of the appropriate learning resources to the adaptive engine as current meta data for learning objects does not specify 3D specific issues (such as size, texture, etc.) needed to be able to do an appropriate selection.

- From our work in GRAPPLE, we also know that in order to have coherent adaptation actions inside the 3D VLE it is important to monitor the user's behaviour and interaction. This was already recognized in the context of the adaptive educational game domain (Moreno-Ger et al., 2008d; Martínez-Ortiz et al., 2006; Missura & Gaertner, 2009; Charles et al., 2005). More in particular, *to keep the 3D VLE consistent with the current user knowledge and experience, it is important to record user navigation actions, positions inside the 3D VLE, and interactions with different 3D VLE.* In this way, adaptation actions can adapt the 3D VLE according to the recorded data.

The next section will present the different models that we require to specify adaptation types and strategies.

5.2 Adaptive 3D VLE Conceptual Framework

In this section, we present our conceptual framework for providing adaptivity for 3D VLEs.

To realise adaptivity, it is common to use different models (see section 3.2.2.2 for an overview of existing models), each having a certain goal and specification. In this section, we will specify the models used in our conceptual framework and focus on the models that are specific for supporting adaptation in 3D VLEs, namely the *adaptive topic model* and the *adaptive storyline model*. Furthermore, the *3D VLE activity history model* is used to keep track of the learners' activities inside the 3D VLE, since we believe that adaptation should happen not only according to the learners' characteristics such as user's preferences, knowledge, skills, but also according to the actions learners performed in the past and in current sessions in the 3D VLE (as mentioned in previous section).

Figure 5.1 depicts the different models needed to perform our adaptivity process inside a 3D VLE. It is important to mention that it is not necessary to always use all the models for creating an adaptive 3D VLE. However, for the scope of this conceptual framework, it is important to take all of them into account, as they are important for adapting the 3D VLE in different situations. We will start by briefly explaining the different models; more details will be given in further sections.

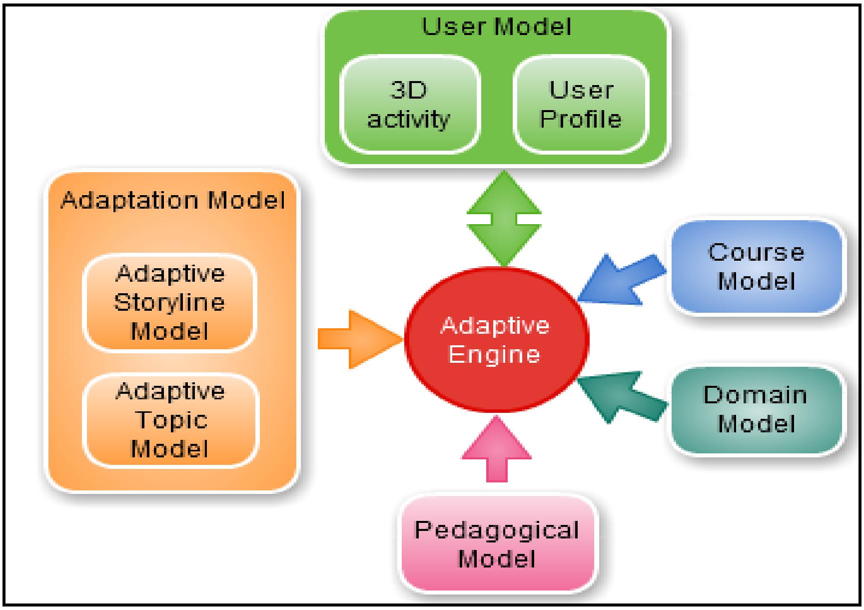


Figure 5.1: Different Models for adaptation

Domain Model: This model defines the different learning concepts used in the 3D VLE. Moreover, different relationships between learning concepts can also be defined in this model, e.g., to present the concept's hierarchical structure. This model actual gives a conceptual model of the learning domain.

Pedagogical Model: This model is used for defining pedagogical relationships between learning concepts that may be useful for adapting the 3D VLE. For instance, pre-requisite relations may be used to define a meaningful learning sequence between the learning concepts. Depending on these pedagogical relationships, different adaptation strategies can be used (defined in the adaptation model).

Course Model: This model is used to create groups of related learning concepts, called topics. These are the topics that need to be mastered by learners during a course. Furthermore, each topic can be given attributes that will be used in the adaptation model. For instance, the topic's difficulty level can be an important attribute to consider in the adaptation process.

User Model: Our user model consists of two sub models. The first sub model is called *user profile,* which is mainly used to maintain information about the learner and his learning progress. It is important to mention that the user profile is used to refer to all information maintained for a single user.

The second sub model is called *3D Activity Model* or *3D VLE Activity History Model.* Next to the information maintained in the user profile, in the case of a 3D VLE, it is also necessary to keep track at runtime of what is happing in the

3D VLE. Such information is needed in order to be able to adapt the 3D VLE to the activities performed by the user (learner) in the 3D VLE. For instance, it may be relevant to know how often the learner has already interacted with specific 3D virtual objects. Therefore, it is necessary to keep track of the objects with which the learner has interacted, which behaviours have been performed, the time spent with a certain object and in a certain location of the 3D VLE, etc. The more information is kept about the activities of the learner in the 3D VLE, the more this information can be taken into account to adapt the 3D VLE to the individual learner. Note that some data from this 3D activity needs to be translated into user model data; for instance, it needs to be defined which activities need to be performed in the 3D VLE by the learner to raise the knowledge level of a certain learning concept.

Adaptation Model: The adaptation model specifies the actual adaptivity defined for the 3D VLE. In general, this is done by means of adaptation rules that specify when, what needs to be adapted and how. For this purpose, we already introduced different modelling concepts, like adaptation types, adaptation strategies, and adaptation themes. The adaptation model is decomposed into two sub models: *adaptive storyline model* and *adaptive topic model*.

First, we describe the *Adaptive Storyline Model*: In this model the storyline that the learner should follow during his learning process is specified. The storyline should be adaptive, e.g., adapt depending on the user knowledge about specific concepts and actions performed. A storyline is composed of a number of topics (defined in the course model).

Secondly, we have the *Adaptive Topic Model*: Inside a topic, adaptation types and strategies can be applied to 3D learning objects to adapt the content, behaviour and interaction provided in the topic. As a result, the actual adaptive behaviour for the learning concepts in a topic is specified in this sub-model.

Adaptive Engine (Adaptation Process): This module is responsible for invoking the adaptation. Based on the information in the adaptation model (usually in the form of rules) it should determine when adaptations need to be performed and what adaptations to perform, and it should instruct to perform the adjustments and modifications needed to the 3D VLE. This module requires information from the different models. Figure 5.2 shows how the different models are used with the Adaptive Engine.

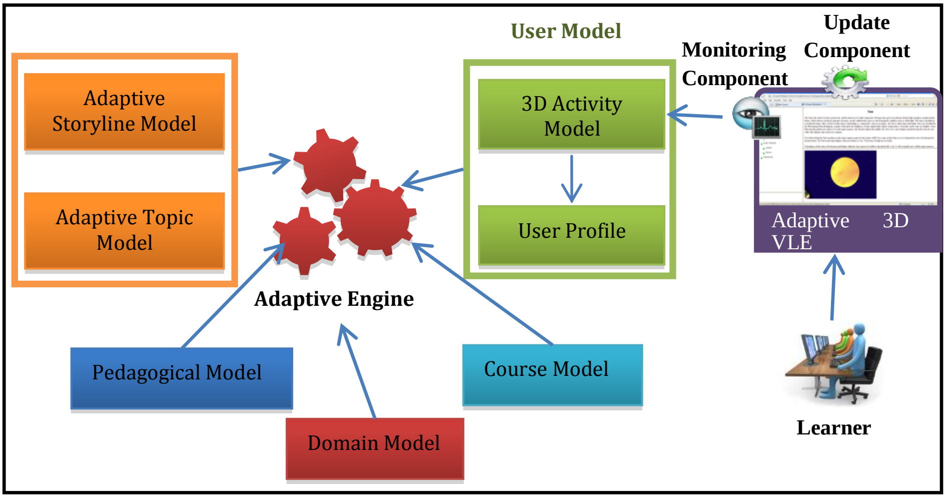


Figure 5.2: Conceptual Framework for Adaptive 3D VLE

An example of how the adaptation process is performed can be as following: while the learner is experiencing the 3D VLE, he interacts with 3D objects. If a 3D object corresponds to (is mapped to) a learning concept, then interaction with this 3D object will be recorded in the *3D Activity* model which can also induce altering values of the user profile (user model). As a result of updating the user model, adaptation rules registered in the adaption model can be triggered. Depending on the triggered rule, the adaptive engine will consult the following models (if needed): the domain model to select (references for) new learning resources; the adaptive storyline model to determine if the learner should start a new topic or jump to a different part of the storyline; the pedagogical model to update the knowledge level and suitability of other related learning concepts; the adaptive topic model to select which adaptation types or strategies should be applied to the 3D VLE.

In the next subsections, we will discuss the different models, as well as the adaptive engine into more details.

5.2.1 Domain Model

The purpose of the domain model is to specify and structure the concepts in the learning domain and their associated learning resources (learning content). As mentioned earlier, our work will not consider authoring the domain model as there is quite a body of literature in this context e.g., (Bonis et al., 2007; Dachselt et al., 2006; Bilasco et al., 2007; Bille et al., 2004; Buche & Querrec, 2011; Hubal, 2008).

As an example, for a solar system course, the domain model will contain learning concepts like, sun, planet, earth and different 3D representations (VR resources) associated with the previous concepts. For example, the learning concept named *Sun* can be linked to a 3D object representing the sun. See Figure 5.3.

It is also necessary to assign attributes to the learning content, such as the level of difficulty, 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) (Hodgins & Duval, 2002).

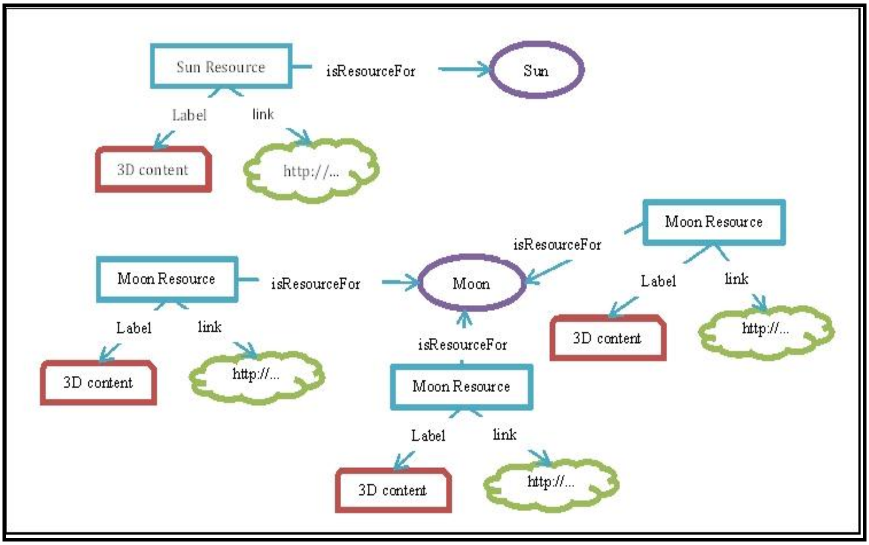


Figure 5.3: Fragment of the domain model for the Solar System showing the link with the VR resources

Furthermore, semantic relationships like *part-of* or *is-a* between the learning concepts are also important to be considered in the domain model, as they provide more structure and can be useful to structure the course. However, for 3D VLE, some additional relationships may be relevant. For example, the Planet concept has a relationship called *rotates-around* the Sun concept. It can for instance be useful to have a representation for this relationship in the 3D VLE (i.e. behaviour).

Let us consider an adaptive 3D solar system course which has a domain model containing concepts such as: Solar System, Sun, Planet, Satellite, Moon, Rotation, Structure, Orbit, Jupiter, Earth; and relationships between these concepts like Sun is-part-of Solar System, Jupiter is-a Planet and domain specific relations such as Planet rotates-around Sun (see Figure 5.4 for a possible graphical representation).

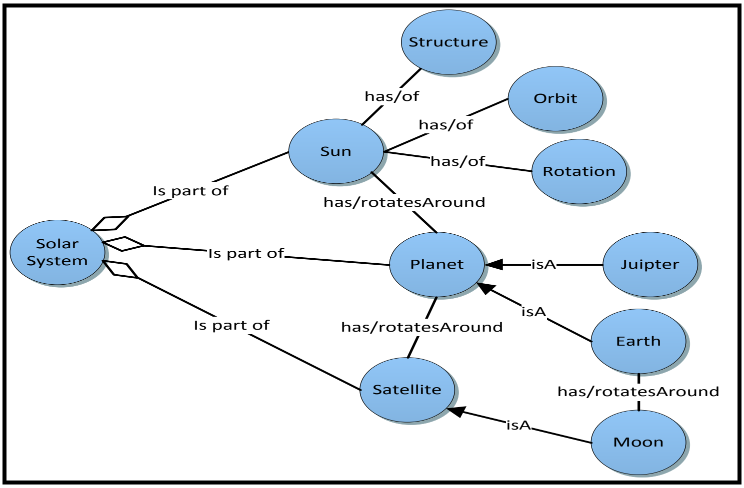


Figure 5.4: Part of Domain Model for 3D Solar System

5.2.2 Pedagogical Model

In classical adaptive systems, the adaptation is mainly based on *pedagogical relationships* between learning concepts such as pre-requisites, inhibitors, and example-of. These relationships can be used to derive a learning sequence (e.g., top-down or bottom-up). Pedagogical relationships are also used to define how the user model should be updated or how knowledge about one learning concept can propagate to other learning concepts on pedagogical bases. This approach is based on the work for GRAPPLE (De Bra et al., 2010), ACCT (Dagger et al., 2005), and LAOS (Cristea & De Mooij, 2003b).

Our work follows also the same approach for defining the pedagogical aspects related to 3D VLE in the sense that pedagogical relationship types are independent from the concepts in the domain model, e.g. the pre-requisite relationship type can be used for different domains. In general, they are predefined.

The pedagogical model is used to define the actual instances of pedagogical relationship types that exist between different learning concepts, as well as to define how the learner's knowledge about the involved learning concepts should be updated. This is realized by attaching so-called *pedagogical update rules* to every pedagogical relationship. For example, if concept X has a prerequisite relation with concept Y, then the learner's knowledge about concept Y will not be updated until he has learned concept X. Examples of such rules are presented in section 6.4.

5.2.3 Course Model

The course model is based on the domain model in the sense that it allows defining groups of related learning concepts that should be studied together by a learner in the context of a certain course, i.e. a subject to study. Those topics will be used later on to define the learning path that the learners need to follow by composing a storyline from the topics. This is similar to the LAOS approach (Cristea & De Mooij, 2003b) where a learning path can be defined implicitly in the goal and constraints model. Also WOTAN[54] (a Web-based Online Task-based Adaptive learNing system) designs the course model by different tasks and fragments which are connected to each other via defined rules to define the sequence of the task and the subtasks.

In our course model, some topics may have more learning concepts than others. It is the author's task to define which learning concepts belong to a topic. To do so, he can use the semantic and pedagogical relationships between these concepts.

We have opted to have a course model separated from the domain model, as different courses may use the same domain model but group learning concepts into different topics, i.e. have different course models.

To make it easy to have adaptation not only at the level of learning concepts but also at the level of topics, we will assume that some attributes are associated with topics, which can be given values for each learner. We will discuss them in the next chapter (topic model attributes in section 6.5.2). An example is the knowledge level of the learner for a topic.

5.2.4 User Model

The user model is an important model that allows the personalization of the learning process by allowing adaptations based on the different characteristics of the learner (i.e. user). In addition, the user model is used to hold all the information about the user, like preferences, goals, knowledge, interest, background and skills. The user model is a model commonly used in personalisation.

As explained earlier, we distinguish between the user profile (usually called the user model) and the 3D activity model.

5.2.4.1 User Profile

In general, the user model also includes demographic data such as name, age and gender, which can for instance be collected explicitly by asking the user to fill in a preliminary questionnaire. Such demographic data can be used to adapt the 3D VLE according to, for example, the user's gender. E.g., for female learners different colours could be used than for male learners. This kind of information is rather static and will (in general) not change during the lifetime of the user model. On the other hand, other data about the user may be more

[54] http://tangow.ii.uam.es/wotan/

dynamic, like user knowledge, preferences, and skills. These are likely to change during the learning process. This information can be kept up to date by using certain strategies for gathering information about the user's knowledge, preferences, and skills while taking the course.

There exist different approaches to structure the information in a user model. For instance, some approaches proposed a basic user model with a simple collection of variables and values. Another representation is a hierarchal model that considers relationships between the different user characteristics. A stereotype approach is also proposed to identify the user to groups and generalize their interaction behaviour into patterns to construct groups in the user model (Kobsa, 1993). More details about different user modelling approaches and frameworks can be found in (Kobsa, 2001). In the context of e-learning, it is common to have a user model that is an overlay of the underlying domain model (structure) like in AHA! (De Bra et al., 2003) and GRAPPLE (De Bra et al., 2010), as this allows to easily keeping track of specific data related to the different learning concepts for each user. We also follow the same approach as in GRAPPLE. Note that some systems adopt a centralized user model that several client applications like adaptive applications can benefit from it (Klobucar et al., 2004; Nejdl et al., 2002; Kobsa & Fink, 2006).

Initiating a user model is also an important issue. When a user starts an e-learning application, usually his knowledge for all learning concept in the domain model are initiated with zero values. This may be incorrect as the user already may have some knowledge about some concepts in the domain model. One way to solve such problem is discussed in (Brusilovsky et al., 2005) which propose exchanging user model data between different adaptive applications. However, privacy is an issue. Another mechanism is asking users to explicitly fill in a form which will initiate the user model with the current level of knowledge and preferences, etc. For instance, (Aroyo et al., 2006a) proposed a dialog game for acquiring needed information about users. An interesting contribution to user modelling with semantic web technologies is discussed in (Dolog & Nejdl, 2003; Dolog & Schäfer, 2001).

We will not consider the modelling or realization of the user profile in detail as this has already been considered in a lot of other work. However, we assume that the user model should provide the adaptive engine with the required data about a user in relation to the domain and topic model and that it is possible to update this user data. For instance, it is possible to use services to retrieve, access, and update all the required data concerning the user profile. Furthermore, we assume that the user profile provides a minimum number of required attributes. We define them in section 6.4 and section 6.5.

5.2.4.2 3D Activity Model

In this section, we will discuss another important model for our adaptive process of 3D VLE, called the *3D activity model* or *3D VLE activity history* model.

Actually, we consider this model to be part of the user model, as it maintains information per user, but we describe it separately because it is typical for 3D VLEs (De Troyer et al., 2009).

As mentioned earlier, to be able to adapt content to the learner, besides having information about the learner (personal preferences, background knowledge, etc.), it is also important to keep track of his/her learning progress. This is usually done through the use of the user model. However, in case of a 3D VLE, it is also necessary to keep track, at runtime, of what the learner is doing the 3D VLE. For instance, to be able to use some of the adaptation strategies mentioned earlier, it is necessary to keep track of the objects with which the learner has interacted, or which kind of actions have been performed, or the time spent with a certain object or in a certain location of the 3D VLE. We illustrate this with an example. To be able to perform the adaptive behaviour associated with the adaptation strategy *displayAtMost*, the system needs to keep track of how often the learner has already visited the object. Furthermore, in case of network disconnection or revisiting the 3D VLE, there should be a way to restore the 3D VLE with the latest modifications. This kind of information will be maintained in the 3D VLE activity history. The more information is kept about the activities of the learner in the 3D VLE, the more this information can be taken into account to adapt the 3D VLE to the individual learner. Note that some data from this 3D VLE activity history needs to be translated into user profile data, for instance it needs to be defined which activities performed in the 3D VLE by the learner will raise the knowledge level of a certain topic.

Note that to be able to record different learner's activities, some sort of sensors needs to be placed inside the 3D VLE. The sensors detect the actions performed by the learner and send the relevant data to the 3D VLE activity history model.

The same as for the previous model, we assume that the 3D activity model provides a minimum number of required attributes related to the user's 3D activities inside the 3D VLE. We define them in section 6.5.2.

5.2.5 Adaptation Model

As in any adaptive 3D educational system (video games or 3D VLE), an adaptation model is required to define the adaptive behaviour of the 3D virtual environment. As mentioned earlier, an adaptation model is used in (Marchiori et al., 2012) to define the story-based game. Furthermore, the adaptation model in 80Days[55] project is used to define the micro and macro adaptation techniques inside the 3D games. Our work also defines adaptation at two levels.

[55]http://www.eightydays.eu/

At the first level, our approach uses the concept of a *storyline,* which is represented by a sequence of *topics* that are used to form an instructional process for learning the course material and to guide the learner through a predefined learning path. The storyline can be adaptive in the sense that it is possible to specify how the storyline (i.e. the sequencing of the topics) should be adapted to the target learner. The adaptive storyline should be specified by the author. This adaptive storyline is described in the *Adaptive Storyline Model*. Adaptation is specified using a rule-based mechanism.

Usually, the learner acquires knowledge about learning concepts and makes progress in the learning process by experiencing and examining the 3D VLE and interacting with the 3D contents. Also at this level adaptations may be useful to guide the learner. Therefore, at the second level, we have the *Adaptive Topic Model,* which specifies the adaptation behaviour for each topic individually. For each topic, it specifies when and how the content and structure of the 3D VLE needs to be adapted. To achieve that, a rule-based mechanism is used.

Both the adaptive storyline model and the adaptive topic model will be described in more details in the next chapter (Chapter 6).

5.2.6 Adaptive Engine

The adaptive engine is the processing part in our conceptual framework. In general, the adaptive engine is used to drive the adaptations. In other words, it is responsible for executing the adaptation rules specified in the adaption model and the rules specified in the pedagogical model. Accordingly, it will perform updates to the user model and invoke changes inside the 3D VLE.

The adaptation uses an event driven approach. In other words, once an event occurs the adaptive engine starts the adaptation process. In our approach, we have two types of events. The first type is related to events that occur due to changes in the user model. As a result of such an event, the adaptive engine will consult the pedagogical update rules defined in the pedagogical model to check if any updates are required to the user model attributes. The second type is related to so-called VR events types like interacting with 3D objects, navigating close to 3D virtual objects, etc. Once a VR event occurs, the adaptive engine will consult the adaption model to check if any adaptation rules need to be executed. When the adaptive engine notices that the if-part of an adaptation rule is satisfied, it will be responsible for performing the action part, i.e. apply adaptation types or adaptation strategies to 3D VLE components as specified in the action part of the rule. In this way, the 3D VLE will be adapted at run time.

In brief, the adaptive engine receives a request from different events types: either to update the user model based on a triggered pedagogical update rule, to start a new topic based on an adaptation storyline rule, or to apply adaptation actions inside the 3D VLE based on an adaptation topic rule. The adaptive engine performs the adaptation tasks in a number of stages. The following are the tasks that are performed by the adaptive engine:

1. Load the required attributes values of the user model and 3D VLE activity model. The retrieved attributes are the ones composing the condition part of the rules.
2. Evaluate the rules associated with pedagogical relationship types for updating the user model (i.e. the pedagogical update rules), the adaptation rules defined in the adaptive storyline model and in the adaptive topic models.
3. Based on the triggered rule type, one of the following action will be executed:
   - For an adaptive storyline rule: Selects the appropriated topic according to the rule. Instruct the 3D VLE to update according to the select topic.
   - For an adaptation topic rule: Selects proper learning resources and adapted them; adapt the navigation possibilities; adapt interaction mechanism as needed. Instruct the 3D VLE to update accordingly.
   - For a pedagogical update: Update the relevant user model attributes.

5.3 Summary

In this chapter, we have proposed a conceptual framework to support the adaptation of 3D VLEs. In the first section, we discussed the basic principles defined for designing an adaptive 3D VLE. An important principle is the introduction of an adaptive storyline, which is composed of topics containing adaptable VLE components, to specify an adaptive 3D VLE.

Next, we introduce our conceptual framework, which relies on different models that are combined with an adaptive engine to drive adaptivity. The models are explained, their origin has been identified, and our motivation for having these models is given.

First, the domain model represents the learning domain on a conceptual level. It defines learning concepts and the hierarchical and semantic relationships between these concepts. The domain model is complemented by the pedagogical model, which defines pedagogical relations between the learning concepts. On top of the domain model, a course model is defined to group learning concepts into topics that should be studied together for a course. The course model defines the different subjects (i.e. topics) to be covered and mastered in the 3D VLE.

Secondly, the user model allows the personalization of the learning process by providing and keeping track of the learner's knowledge, preferences, and other relevant data. This model is complemented with the 3D activity model that keeps track of all user activities and interactions inside the 3D VLE.

Thirdly, the adaptation of the 3D VLE is specified in the adaptation model that consists of an adaptive storyline model and an adaptive topic model. These two models are used because we want to define adaptivity at two levels. Firstly, at the level of the storyline, i.e. the sequence of the topics will be selected adaptively based on the needs of the learner. Secondly, at the level of a topic, i.e. the 3D objects related to a topic should also be adapted to the needs of the learner. This approach, which divides the specification of the adaptive 3D VLE into an adaptive storyline model and an adaptive topic model, makes it possible to directly specify the overall flow of the course (i.e. the storyline) and makes it easier to define what can be adapted, how it can be adapted and when it should be adapted at the level of the storyline, as well as at the level of the individual topics. This separation actually provides two layers of abstraction.

In conclusion, we can state that our approach is a conceptual and model-based approach. The conceptual framework is explicitly addressing the problems that are recurrent in creating adaptive 3D VLEs: describing the learning domain and related pedagogical aspects, capturing and maintaining user data and 3D VLE activities, and specifying adaptation actions for the storyline as well as for the other components of the 3D VLE.

The next chapter will present the authoring approach proposed to specify (i.e. to author) an adaptive 3D VLE course.

Chapter 6: Authoring Adaptive 3D VLE

The previous chapter discussed the principles of our approach for designing adaptive 3D VLEs, as well as our conceptual framework for adaptive 3D VLEs describing the different models required for our approach.

In this chapter, we present our authoring approach, which is based on the principles and the conceptual framework introduced in the previous chapter. Starting from different scenarios for authoring adaptive 3D VLE, we derived a number of functional and usability requirements that should be supported by the authoring approach. Furthermore, we propose visual languages specifically conceived to allow authors to specify the adaptivity in 3D VLEs. In particular, we propose three complementary visual languages to support authors in creating adaptive course storylines and adaptivity in the 3D VLE.

This chapter is structured as follow: section 6.1 describes the requirements for our authoring approach. In section 6.2, presents the concept of adaptation theme. Section 6.3 discusses the use of visual modelling languages for the authoring in general. The proposed languages are discussed in section 6.4 and section 6.5. Section 6.6 summarises the chapter.

6.1 Requirements for the Authoring Approach

Similar as for classical adaptive learning system, there are essentially three different approaches for realizing adaptivity: an author-driven approach, a teacher-supervised approach, and a model-driven approach.

In the *author-driven approach*, the author of the course is given the full control over the adaptation process. During the design of the course, the author needs to specify the adaptations explicitly, usually through a set of 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 to do. 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 adaptive engine using intelligent algorithms. The models and algorithms can be engineered in advance or be based on advanced AI techniques (Manslow, 2004, 2002; Funge, 2000). This approach is used in so-called intelligent tutoring systems (Sleeman & Brown, 1982). Although, it relieves the work of the author, it inevitably also introduces in-transparency. The author does not know in advance how the learning environment will be adapted. The last approach, the *teacher supervised 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 do not need to preview all possible adaptation scenarios, 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 often used.

For our work, we have adopted the author-driven approach. As mentioned earlier, the advantage of this approach is that it gives the educators the possibility to define the adaptive behaviour of the 3D VLE explicitly. As a result, the author will have full control over the adaptation process. Even though, the disadvantage is that it requires the author to keep track, at design time, of all possible adaptation scenarios. However, we believe that supporting authors with appropriated tools can overcome this disadvantage.

In general, since the authoring process should allow an author to define an adaptive 3D course, he should be able to specify the different models that are adopted in the conceptual adaptation framework. To support this activity, an authoring approach is proposed in the context of this work.

The authoring approach should make it easy to develop the conceptual models required for the adaptive process. In other words, it should allow the educator (author) (1) to specify the learning concepts and the 3D resources of the course (i.e., the domain model), (2) to specify the pedagogical aspects that apply to the different learning concepts (i.e., the pedagogical model), (3) to specify the adaptive storyline to be followed by learners (i.e., the adaptive storyline model), and (4) when, which and how adaptation techniques should be applied for the different topics (i.e., the adaptive topic model). Note that the two other models, the user model and the 3D VLE activity history, are not created by the author; their structure is the same for each 3D VLE.

The most important required characteristic for adaptive 3D VLE authoring approach is that it should be *educator-oriented* which means that educators need to be directly involved in the development of the adaptive 3D VLE.

As can be expected, educators do not have deep knowledge about 3D and adaptivity. Therefore, the used terminology should be easy to understand and resorting to very technical issues and technology such as scripting and programming should not be required. Also the educators should be supported with a user-friendly user interface. Therefore, we have opted for visual modelling languages. In the next section we motivate this choice.

However, it has to be noted that specifying an adaptive 3D VLE is not simple, even if all effort is done to abstract from technical issues. A minimal of knowledge about VR is still needed, as well as being able to specify condition-action rules. Therefore, some training may be necessary for novice users.

To come to the different requirements for the authoring approach, we start by describing a number of scenarios for the authoring of adaptive 3D VLEs that were used to derive the requirements. After that, the actual requirements are given. We differentiate between functional requirements and more usability oriented requirements.

6.1.1 Scenarios for Authoring Adaptive 3D VLE

As mentioned earlier, to enable the identification of the requirements for an authoring approach for adaptive 3D VLEs we have used scenarios. The scenarios define how an example course should be authored. The example is as follow: A teacher (author) likes to teach his students the solar system by means of 3D representations of the sun, the planets, the satellites, and moons along with some text for each of these learning concepts. The scenarios are established based on the common way of authoring a classical e-learning course combined with our experience gained in the GRAPPLE project.

First, the author should establish the course domain by defining the learning concepts required and their semantic relationships. For instance, he can define Sun, Satellite, Moon, Planet, Earth, Mars, Venus, Mercury, etc. as learning concepts. Moreover, semantic relationships such as is-a, composed-of, part-of, etc. can be used to define the domain structure.

The author has also to indicate pedagogical relationships between learning concepts and how the learner knowledge about the different learning concepts will be increased.

To be able to create such a 3D course, the author will also need to have 3D models representing the Sun, the planets, the satellites, and the moons to be included inside the 3D VLE. Note that designing such 3D models is beyond the scope of the adaptive authoring. However, the author needs to associate these 3D resources to their corresponding learning concepts. For instance, he should associate the 3D representation(s) of earth with the learning concept Earth. Note that a learning concept may have different 3D representations for the same learning concept. It is also important to keep in mind that other resources like audio and textual resources need to be mapped to their related learning concepts so that the learner will be provided with textual, audio, and 3D representations resources according to their suitability and user profile.

Next, the author creates the main storyline of the course. As for any course, he can divide the course into different topics. Every topic will deal with some learning concepts from the domain model. Linking the topics together gives the main storyline.

Furthermore, the author specifies the required adaptations. In a first scenario, the author wants to express that the 3D representation used for a learning concept will depend on the user's knowledge level about that concept. For example, the author wants to define an adaptation rule to change the 3D resource for Earth with another 3D resource having more details when Earth has already been studied.

Another authoring scenario can be expressed as follow: The author would like to provide a 3D virtual solar system that does not only display adaptively the 3D objects which represent the Sun and the planets, but also their behaviours. Moreover, the author wants that the learners are able to navigate through the 3D VLE, interact with the 3D objects Venus, Earth, and Mercury, view their rotation around the Sun and at the same time be able to read some textual information about these learning concepts. However, providing all this at the same time may be overwhelming for some learners. To avoid that, the author wants to specify some adaptation techniques to enable or disable user interaction and the 3D objects' behaviours. Furthermore, he may want to specify when the learner can experience the 3D models or read textual information about related learning concepts.

6.1.2 Functional and Usability Requirements

Based on the scenarios mentioned in the previous section and the observations made about the target users (i.e. authors), we derived a list of general requirements (functional as well as usability requirements) for our authoring of adaptive 3D VLEs. Note that different requirements were already identified in other work (also in GRAPPLE). We indicate this where relevant.

Functional requirements

1. It must be possible to specify the learning concepts to be considered and the conceptual relationships between those concepts.

   Motivation: This approach is used in most authoring tools (in the context of adaptive hypermedia e.g., (Cristea & De Mooij, 2003a; De Bra et al., 2003; Dagger et al., 2004; De Bra et al., 2010), in the context of educational games e.g., (Hodhod et al., 2009; Kickmeier-Rust & Albert, 2007; Torrente et al., 2008)) as well as in GRAPPLE, as it allows to better structure the course. Also, in adaptive learning systems, the knowledge of the learner is expressed in term of the different learning concepts to study (De Troyer et al., 2009).
2. It must be possible to differentiate between learning concepts and the actual (3D) learning resources that will be used for a learning concept.

   Motivation: Different 3D learning resources may exist for the same learning concept. The context of use may determine which one is best to use. This was already realized in our work in the context of GRAPPLE and is considered as an important requirement (De Troyer et al., 2009).

3. The learning concepts should be mapped with their related 3D learning resources.
   Motivation: as 3D resources are used in the 3D VLE, it must be possible to trace the learning resources they belong to, as knowledge is expressed in terms of learning concepts. This has also been explored in (Carro et al., 2002; Estalayo et al., 2004) and GRAPPLE.
4. It should be possible to express pedagogical relationships between the different learning concepts.
   Motivation: pedagogical relationships are important to structure the learning material in a course. Furthermore, it defines how the user's knowledge needs to be increased. This is also realized in adaptive hypermedia (De Bra et al., 2010; Dagger et al., 2005; Cristea & De Mooij, 2003b) and educational games (Begg et al., 2005; Mills & Dalgarno, 2007), as well as in GRAPPLE.
5. It should be possible to specify how the knowledge of the learner will increase for the different learning concepts.
   Motivation: To allow adapting the 3D VLE to the knowledge of the learner, it must be possible to keep track of the learner's knowledge progress. This was also realized in (Chittaro & Ranon, 2007b) and in GRAPPLE.
6. It should be possible to use the learning progress (i.e. increase of knowledge) of students for adapting the 3D VLE.
   Motivation: this is one of the principles of adaptive learning systems (see also (Göbel et al., 2009; Yun et al., 2009; Charles et al., 2005; Missura & Gaertner, 2009; Hodhod et al., 2009) and GRAPPLE).
7. It should be possible to use the activities done by the learner in the 3D VLE for adapting the 3D VLE.
   Motivation: this will allow adjusting the 3D VLE to the learner 3D skills as well to issues related to his motivation and attention (see also section 5.2.4). This was also realized in (Chittaro & Ranon, 2007b; Lepouras & Vassilakis, 2006; Togelius et al., 2007; Magerko, 2006) and in GRAPPLE.
8. It should be possible to specify adaptions at the level of the actual 3D resources.
   Motivation: Most adaptations on classical material can be specified on the level of the learning concept and are independent of the actual resource used for the learning concept, e.g., hiding a piece or text or grey a link can be specified independent of the actual text or the actual link. This is different for VR/3D VLEs (De Troyer et al., 2008,2009). For instance, as object's behaviours are usually strongly connected to an actual 3D object, it is not always possible to replace certain behaviour by another. Similarly, it is also not always possible to replace one 3D object in 3D VE by any other 3D object, as this object's properties may not fit into the 3D VLE under consideration. This was already recognized in GRAPPLE.

9. It should be possible to specify storylines for courses.
   Motivation: Usually, educators have a certain storyline in mind that they want the learner to follow. A typical example is the need to specify narrative aspects in most of the education games (Hulpuș et al., 2010; Göbel et al., 2009; Thomas & Young, 2007; Hodhod et al., 2009). This will enable authors to define the sequence of concepts and learning topics (i.e., learning path) that the learners need to follow inside the 3D VLE (Ewais & De Troyer, 2013a). See section 5.1.1 for more details.
10. It should be possible to specify adaptations for the storyline of a course.
    Motivation: As we are aiming for an adaptive 3D VLE, it may also be necessary to adapt the main storyline of the course to the learner (Ewais & De Troyer, 2013a).

Usability requirements

11. Educators should be able to provide some textual hints and notifications to the learners whenever it is needed. For instance, authors can define a text message to notify the learner that an adaptation action will happen.
    Motivation: From a usability perspective, it is important to keep the user always informed about the state of the system (Nielsen, 1993). Therefore, it is also important to inform learners about what adaptations will be applied in the 3D VLE. This was also derived from the evaluation conducted for GRAPPLE (Ewais & De Troyer, 2013b) as the participants preferred to receive textual notations when an adaptation action happens.
12. The authoring approach should bridge the gap between educators, adaptive 3D VLE technology and authoring terminologies.
    Motivation: Educators are usually not schooled in programming or in 3D/VR. Therefore, the authoring approach should shield up the educators from any technical details and technological implementation. This is also recognized in the context of StoryTec (Göbel et al., 2009), WEEV (Marchiori et al., 2012), and GRAPPLE (De Troyer et al., 2009).
13. Experienced educators should be able to see and/or specify detailed technical 3D information where relevant.
    Motivation: If educators are willing and capable of doing more advanced adaptation they should not be limited in doing so. The authoring approach should provide authors expressive means to create the different models needed for composing the adaptive 3D VLE. This is also realized in the context of AHA! (De Bra et al., 2003) and GRAPPLE (De Bra et al., 2010) where advanced authors are provided with more advanced functionalities to increase the controllability of the authoring process.

14. Authors should be supported in creating adaptation in a consistent way.
    Motivation: Consistency is an important aspect of usability. Different adaptations for the same or a similar purpose should be avoided, as this will be confusing for the learner. Therefore, authors should be able to have a consistent adaptation style for the complete adaptation process. For instance, this could be used to avoid having different specifications for bounding boxes that are used for the same purpose. This is discussed further in next section (section 6.2).

The functional requirements are considered in further sections. Requirements 1, 2 and 3 are to be supported in the authoring of the domain model. We will not elaborate on authoring the domain model as this has already been considered in many other works (see e.g., (Brusilovsky, 2003; De Bra et al., 2003)). Furthermore, providing the pedagogical model language will satisfy requirements 4 and 5. In addition, requirements 6 to 10 will be supported while authoring the adaptation model. More particularly, requirements 6, 7, and 8 are supported by providing the adaptive topic model language, and requirements 9 and 10 by the adaptive storyline model language.

Usability requirements (requirements 11-15) will be satisfied in the proposed authoring tool. More details are discussed in Chapter 9.

Before we start elaborating on the visual languages which are proposed to help authors in defining adaptive 3D VLE, we elaborate a bit more on requirement 15. To satisfy this requirement, adaptation theme specifications are introduced. They are discussed in the next section.

6.2 Adaptation Theme Specifications

Adaptation themes are used to provide consistent specifications for the adaptation process. We have opted to allow providing adaptation theme for different target groups. By default, adaptation themes are used for three target user groups: Beginners, Intermediate, and Advanced users. However, authors are still able to define theme specifications for other targeted users.

Adaptation theme specifications allow customizing colours, size, texture mapping, audio play, etc. For instance, when the author uses spotlight adaptation type for beginner learner, he can specify a red colour for its displayed ray. Alternatively, for advanced learners, the author may adopt the green colour for the spotlight ray. The same can be applicable for other adaptation types related to other targeted groups.

In general, adaptation themes can be defined by customizing specifications in terms of different representations for adaptation types. Beginner authors could use predefined adaptation themes, while authors advanced in 3D modelling are able to adjust theme specifications.

More technical details related to adaptation theme will be given in section 9.5.4.

6.3 Visual Languages for Authoring Adaptive 3D VLEs

In order to support authors in the creation of adaptive 3D VLEs, a set of visual languages is proposed. In general, a graphical or visual language is defined as "set of graphical symbols, a set of compositional rules for how to form valid visual sentences, and definitions of their meanings " (Moody, 2009).

The rationale behind using visual languages is derived from the observation that graphical specifications are, in general, easier for the communication with non-technical people. Visual languages make it easy to convey information, as many people can think and remember things in term of pictures (Boshernitsan & Downes, 2004). Furthermore, visual languages can provide appropriate abstractions that make the specifications easier and less time consuming (Moody, 2009). However, visual languages should be defined with care to be usable and effective.

Usability issues to consider for visual/graphical languages are discussed in (Figl et al., 2010; Moody, 2009). To this end, Moody proposed nine principles for *cognitive effectiveness* related to the design of a visual language (Moody, 2009). Usually, cognitive effectiveness is defined as accuracy, ease and speed with which a representation can be processed by a human mind (Larkin & Simon, 1987). One of the nine principles proposed by Moody is the *semantic transparency*, which refers to an easy association between the notations and their corresponding concepts.

Furthermore, *complexity* is another issue to be considered when designing a visual/graphical language. For instance, providing only basic graphs can run the risk of being too complex to be understood (Lindley, 2005). Complexity management refers to providing a visual notation to represent information without overloading the human mind (Moody, 2009). Cognitive load is determined by the number of elements that should be considered at once (Kirschner, 2002; Kalyuga, 2011). Miller explained that the possible number of elements to be considered simultaneously is approximately seven elements plus or minus two elements (Miller, 1956). According to Moody in (Moody, 2009), a possible solution for reducing the cognitive load is modularization, which is used to divide the complex domain into smaller parts or models. Another important solution is hierarchal structuring which is used to provide different level of details to make complex problems understandable (Moody, 2009).

There are several successful visual languages used in different contexts. One of the best-known conceptual modelling languages is the Unified Modelling language (UML) used in the context of software engineering (Fowler

& Scott, 2000). UML is used by system analysts to create an object oriented design of a software application by means of a set of notations without being restricted to a specific programming language (Bruegge & Dutoit, 2000). Other conceptual modelling languages, for instance used in the context of database design are Entity Relationship (ER) (Chen, 1976) and Object Role Modelling (ORM) (Halpin & Morgan, 2008). UML is more expressive than ER and ORM as it can also deal with modelling behaviour and modelling interaction. As an adaptive 3D VLE is a piece of software, we could consider using UML for the authoring process. However, UML is very much oriented towards computer skilled users. Therefore, we consider UML as such, not appropriate for an educator-oriented approach. In comparison to UML, which is a general purpose modelling language, our proposed modelling languages can be considered as *domain specific languages* (Van Deursen et al., 2000), specially conceived for the domain of 3D VLEs .

In particular, we have defined three visual modelling languages: the *pedagogical model language* to construct the pedagogical model and that will allow specifying pedagogical relationships between learning concepts; the *adaptive storyline model language,* which is used to specify the general storyline which learners need to follow, as well as the adaptation aspects for the storyline (i.e. to create the adaptive storyline model); and the *adaptive topic model language* is defined to specify the required adaptations for the different topics in the storyline (i.e. to create the adaptive topic model).

6.4 Pedagogical Model Language

The *Pedagogical Model Language* (PML) is used to define a pedagogical structure for the learning concepts by means of pedagogical relationships (*requirement 4*). It is also used to express how the knowledge of the learner (or in general the user model data) should be updated based on these pedagogical relationships and the activities performed in the 3D VLE (*requirement 5*).

One of the proposed guidelines for designing a domain specific language is to "*Compose existing languages where possible.*" and "*Adopt existing notations domain experts use*" (Karsai et al., 2009). Therefore, we based our proposed PML on the modelling language used in GRAPPLE to define the conceptual adaptation model using pedagogical relationship types. In more detail, the concept of PRTs is based on the definition of the pedagogical model in the context of GRAPPLE (Albert et al., 2009; De Bra et al., 2010) in the sense that learning concepts are connected to each other using predefined conceptual relationship types (which are called PRTs).

The PML will use the learning concepts defined in the Domain Model. In PML, learning concepts are represented visually as rectangles (see Figure 6.1 A). They can be connected through labelled arrows (see Figure 6.1 B), which represent the pedagogical relationships. Different *Pedagogical Relationship Types (PRTs*) are defined and given a specific colour. For example, the PRT *pre-*

requisite is given a green colour. The reason of using a colour for each PRT is to make it easy for authors to figure out the relationships between learning concepts visually rather than reading the name of every PRT.

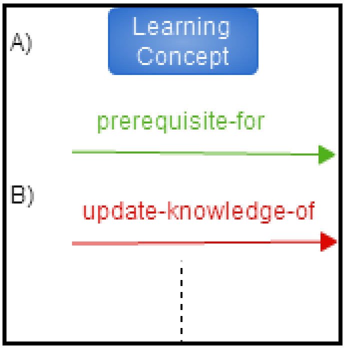


Figure 6.1: PML elements; A) learning concept, B) pedagogical relationship

An experienced author can define his own pedagogical relationship types, but the most commonly used ones are already predefined. When new PRTs are defined a unique name should be specified as it is used as a reference for the individual pedagogical relationships.

Next sub section will present the predefined pedagogical relationship types.

6.4.1 Pedagogical Relationship Types

A typical example of a Pedagogical Relationship Type (PRT) is the *prerequisite*. The goal of this PRT is to define when a learning concept X is a prerequisite for another learning concept Y meaning that the learner needs to study concept X before he can start learning concept Y.

In general, PRTs are limited to binary relationships that connect two learning concepts: source learning concept and target learning concept. The predefined PRTs are:

- *X prerequisite-for Y*: this PRT is used to connect two concepts X and Y, where X is the prerequisite of Y.
- *X co-requisite for Y*: this PRT is used to indicate that the learning concept X must be followed at the same time as learning concept Y.
- *X defines Y*: this PRT is used to connect two concepts X and Y, where X is one of the concepts used in the definition of Y.
- *X illustrates Y*: this PRT is used to express that a concept X illustrates the learning concept Y.
- *X interesting-for Y*: this PRT is used to indicate that concept X is an interesting concept for learning concept Y. For example, if the learner

is learning about earth, it might be interesting to also learn about its moon. So, the concepts moon and earth could be connected through the interesting-for PRT.

- *X inhibits Y*: this PRT is used to indicate that concept X should not be accessed (learned) after having accessed concept Y.
- *X updates-knowledge-of Y*: this is used to indicate that the user's knowledge of concept Y is updated after accessing (learning) concept X. For instance, learning about Mars can update the knowledge about Earth as both of them are related to the Inners solar system.

An example of the use of PML is depicted in Figure 6.2. The learning concept *Sun* is a *prerequisite for* some other learning concepts: *Mercury, Venus, Mars,* and *Earth*. This means that *Mercury, Venus, Mars,* and *Earth* should be learned after getting the required knowledge about *Sun*. Furthermore, when the learner learns about Mercury, his knowledge about Venus will also increase. This is specified by having the *update-knowledge-of* pedagogical relationship between Mercury and Venus. Likewise, this relationship (*update-knowledge-of* PRTs) is also applied to other concepts: Venus, Mars, and Earth.

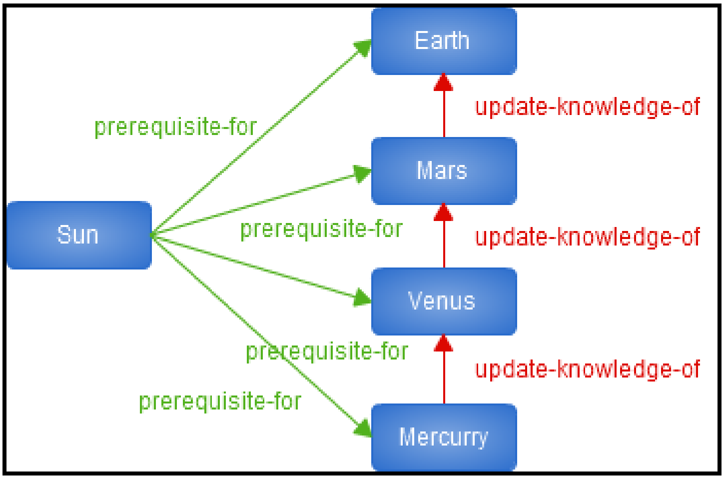


Figure 6.2: Pedagogical Model for a couple of learning concepts related to the Solar System Course.

6.4.2 Pedagogical Update Rules

To express how the knowledge of the learner (or in general the user model data) should be updated based on the pedagogical relationships; PRTs are associated with *Pedagogical Update Rules* (*PURs*), which are condition-action rules. PURs are responsible for defining how the user model will be updated. More concretely, it will define when, how and which user model attributes should be updated. All the PURs are triggered on accessing their source learning concept. Recall that, there are always two learning concepts involved in a PRT, one being the source and the other being the target. For instance, *Sun* is the source learning concept and *Mercury* is the target learning concept for *prerequisite-for* PRT as displayed in the previous figure.

The action part of a PUR defines the updating mechanism that should be applied. For instance, the *update-knowledge-of* PRT's rule will update the knowledge level of the target learning concept. Another example is the *prerequisite-for* PUR, which could be used for determining the suitability of the target learning concepts for the learner.

Note that PURs need to be defined (by the author) for newly defined PRTs. Nevertheless, existing PRTs are already associated with predefined PURs so that novice authors can use them directly. However, advanced authors may update them to fine-tune them to their requirements. Note that supporting experienced and inexperienced authors in the definition of the PRTs was also supported in context of GRAPPLE (Albert et al., 2009).

The general format for the PURs is as follows:

IF <user_model_condition>
Then <user_model_update_actions>

The formal specification for PUR is defined in EBNF (Garshol, 2003) as follow:

<user_model_condition> ::= <learningConcept_name> "." <userModel_attribute> <relational operator> <value>.

<learningConcept_name> ::= <character> { <character> }.
<character > ::= 'A' | 'B' | 'C'| 'D' | 'E' | 'F' | 'G' | 'H' | 'I' | 'J' | 'K' | 'L' | 'M' | 'N' | 'O' | 'P' | 'Q' | 'R' | 'S' | 'T' | 'U' | 'V' | 'W' | 'X' | 'Y' | 'Z' | 'a' | 'b' j 'c' | 'd' | 'e' | 'f' | 'g' | 'h' | 'i' | 'j' | 'k' | 'l' | 'm' | 'n' | 'o' | 'p' | 'q' | 'r' | 's' | 't' | 'u' | 'v' | 'w' | 'x' | 'y' | 'z'|' '| '-'| '_' .
<userModel_attribute> ::= 'knowledge' | 'suitability' | 'interest' | 'visited'.
<relational operator> ::= 'greater-than' | 'less-than' | 'is' | 'is-not'.
<value> ::= <level> | <boolean>.
<level> ::= <digit> [<digit>] .
<digit> ::= "0" | "1" | "..." | "9" .
<boolean> ::= 'TRUE'| 'FALSE'.

<user_model_update_actions> ::= <learningConcept_name> "." <userModel_attribute> 'becomes' <value>.
<learningConcept_name> ::= <character> { <character> }.
<userModel_attribute> ::= 'knowledge' | 'suitability' | 'interest' | 'visited'.
<value> ::= < +UPDATELEVEL> | <boolean>.
<+UPDATELEVEL > ::= <digit> [<digit>] .
<digit> ::= '0' | '1' | '...' | '9' .

<boolean> ::= 'TRUE'| 'FALSE'.

Learning concept name refers to learning concepts from the domain model. For the sake of simplicity, we refer to learning concepts in our further PUR examples by characters like X, Y, etc.

User model attributes are also needed in a PUR to specify how they should be updated. The proposed attributes, which are used in the context of the pedagogical model, are taken from previous work; particularly, AHA! in (De Bra et al., 2002) and GRAPPLE in (Albert et al., 2009). We suppose that the following user model attributes exist (or attributes in the user model can be mapped on these attributes) (see table 6.1): the user *knowledge* attribute represents the knowledge level (range 0 - 100) of a specific learning concept; the *suitability* attribute represents the appropriateness (true – false) of a specific concept to be learned; *visited* is used to represent the number of times that the learner accessed (0 – many times) a learning concept, the *interest* attribute is used to indicate the interest level (0 - 100) of a specific learning concept for a learner, and *preference* is used to keep track of the learning preferences for a specific learning concept) of a learner.

User Model Attribute	Type	Value	Description
knowledge	Numerical	0-100	This attribute is needed to represent the user's knowledge about a specific learning concept.
interest	Boolean	True – False	This attribute is used to indicate how a learning concept might fall in the user's interest.
preference	Boolean	True – False	This attribute is needed to know the learning concepts that the learner prefers to learn.
suitability	Numerical	0-100	This attribute determines when the learner is able to learn about this topic
visited	Numerical	0 – X	This attribute is used to represent the number of times that the learner accessed a learning concept.

Table 6.1: List of user model attributes

Next, relational operator refers to *greater-than, less-than, is,* and *is-not* which are used to compare user models attributes with specific values. Such values are represented as either LEVEL when it is related to numbers or TRUE and FALSE when it is related to boolean conditions.

The updating user model mechanism (action part of the PUR) is the second part of the PUR. This part has a similar syntax as the previous part except from the <relational operator> that is replaced by "becomes", to allow assigning new values to the user model attributes.

The following PURs are associated to the predefined PRTs as default rules. LEVEL and UPDATELEVEL are knowledge levels to be specified when using the rule (or the default value can be used):

- *X prerequisite*-for Y: The PUR will make concept Y suitable when the knowledge level of concept X has reached a certain level:
  IF (X.knowledge greater-than LEVEL)
  Then (Y.suitability becomes TRUE)

- *X co-requisite for Y*: This PUR is used to indicate that the learning concept X must be followed at the same time as learning concept Y.
  IF (X. suitability is TRUE)
  Then (Y.suitability becomes TRUE)

- *X defines Y*: This PUR will update the knowledge level of concept Y when the knowledge level of concept X has reached a certain level:
  IF (X.knowledge greater-than LEVEL)
  Then (Y.knowledge becomes +UPDATELEVEL)

- *Y interesting-for X:* This PUR will update the interest attribute for concept Y when interest attribute value is TRUE for X:
  IF (X.interest is TRUE)
  Then (Y.interest becomes TRUE)

- *X illustrates Y*: This PUR will make concept Y suitable when the learner accessed the learning concept X:
  IF (X.visited is LEVEL)
  Then (Y.suitability becomes TRUE)

- *X Inhibits Y*: this PUR will set the suitably attribute of Y to false if the learner accessed the learning concept X.
  IF (X.visited is LEVEL)
  Then (Y. suitability becomes FALSE)

- *X updates-knowledge-of Y*: This RUP is used to indicate that the user knowledge of the learning concept Y is updated upon accessing the learning concept X.
  IF (X.visited greater-than LEVEL)
  Then (Y.knowledge becomes +UPDATELEVEL)

6.5 Adaptation Model Languages

As mentioned earlier, the adaptation model is composed of two types of models, the adaptive storyline model and the adaptive topic model. The adaptive storyline model enables authors to define how the learner should proceed in his learning and acquiring knowledge by navigating inside the 3D VLE and by interacting with the 3D models (*requirement 9*). Furthermore, the author can define adaptation rules to deliver the learning path adaptively (*requirement 10*). The adaptive topic model allows authors to define adaptation behaviour of the 3D objects' representation, interaction, behaviour, and navigation techniques (*requirement 8*). Furthermore, the author can define adaptations depending on the user model and the learner activities inside the 3D VLE (*requirement 6* and *requirement 7*).

The principles of the adaptive storyline model and the adaptive topic model have already been introduced in section 5.2.5. Here we introduce the visual languages defined for specifying these models. The overall storyline is specified using the *Adaptive StoryLine Language* or *ASLL*, and the individual topics by means of the *Adaptive Topic Language* or *ATL*. In the next sections, we will introduce those languages.

Figure 6.3 depicts the relationship between these two languages. At the top level, the adaptive storyline model is used to specify the storyline for the course. At the next level, the content of each individual topic is modelled. Details will be presented in further sub sections.

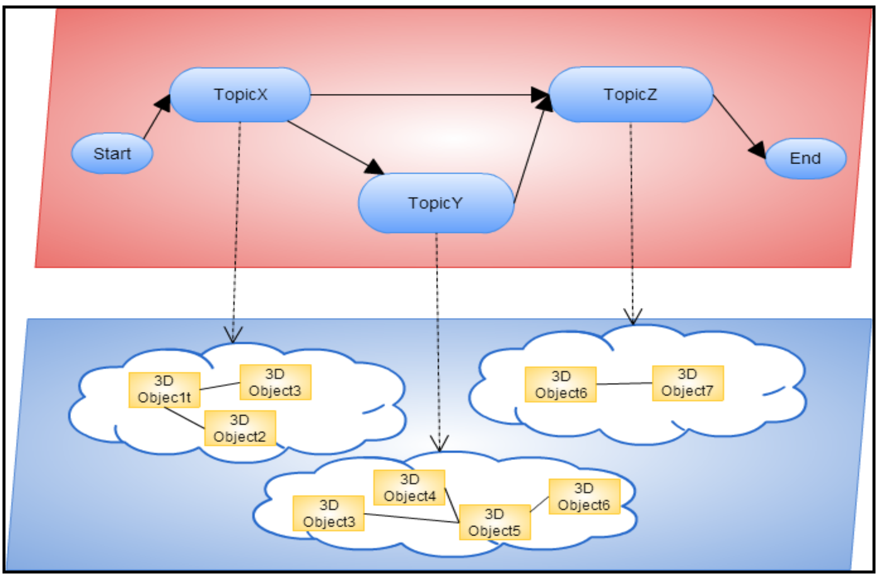


Figure 6.3: Two-level specification of the Adaptation Model

6.5.1 Adaptive StoryLine Language (ASLL)

As mentioned earlier, the concept of an adaptive storyline is inspired by the work done in the context of adaptive 3D games Like (Lindley, 2005) and the StoryTec authoring tool in the 80 Days project (Göbel et al., 2009) which allows the author to create a story for video games.

One of the important guidelines considered in ASLL is to keep it simple so the understandability will be increased (Paige, Ostroff & Brooke, 2000; Karsai et al., 2009; Bobkowska, 2003). Another guideline considered is to avoid unnecessary generality and limit the language elements (Karsai et al., 2009) by decomposing the definition of adaptation to two levels (i.e., adaptation at the storyline level and adaptation at the topic level).

Further subsections will describe firstly, the adopted symbols' representation and their semantic meanings. Secondly, the structure of the language and how the related adaptation rules are formulated are discussed. Finally, an example is presented.

6.5.1.1 ASLL Visual Symbols

Before we start elaborating on this modelling language, it is important to mention that ASLL is similar to a state diagram in UML: it has circles representing the initial state (*start*) and final state (*end*), symbols denoting the states (here topics) and arrows denoting transition (storyline adaptation rule that specify how to make the transition for topic to the other).

ASLL's graphical symbols are shown in Figure 6.4. Figure 6.4(A) shows the notation for the initial state (start) of the storyline, an ellipse named *Start*.

Figure 6.4(B) shows the core building block of a storyline (a topic). A topic is represented by a filled ellipse containing the name of the topic. This symbol represents an abstraction for the actual specification of a topic, which is given by ATL, and which will be explained further on. Remember that a topic covers all learning concepts that are related to the same topic.

To specify adaptivity in the storyline, *storyline adaptation rules* are used. In Figure 6.4(C), the notation used for an adaptation rule is shown. Authors are able to give a meaningful name for each storyline adaptation rule in order to increase the readability of the model language.

A storyline adaptation rule is a condition-action rule; each part has its own visual representation. The condition part, which is used to specify when the learner should move to the next topic, (see figure 6.4(D)) is represented by a hexagon containing the condition (here represented by the letter C) along with an arrow pointing to the action part of the rule. The action part, which is used to specify what kind of adaptation strategies (defined in section 4.3) should be applied to the learning concepts involved in the next topic, is represented as a pentagon shape (see Figure 6.4(E)). The pentagon shape may contain the name of the adaptation strategy that needs to be applied to the proceeding topic's learning concepts.

Let us illustrate a storyline adaptation rule between two topics TopicX and TopicY, as follow: Once the learner has the required knowledge about TopicX and the TopicY is suitable for him (condition part), the adaptation strategy 'TourGuide' should be applied to TopicY and the learner will be directed to TopicY.

The end of the storyline is represented by a double ellipse named *End* (see Figure 6.4 F).

It is worth to mention that, both start and end symbols are considered as special nodes, which may include text messages to be presented to the learner. Such text messages are used to notify the user that he will start a new topic or he reached the end the current topic. These specifications are defined by the author in the adaptive topic model (see section 6.5.2).

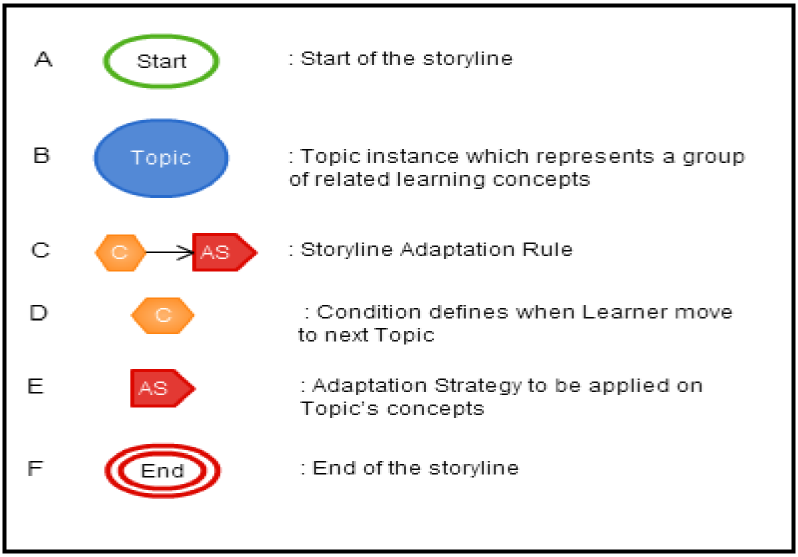


Figure 6.4: ASLL elements. A) start of a storyline; B) Topic in a storyline; C) topic adaptation rule; D) condition; E) adaptation state; F) end of a storyline.

6.5.1.2 Language Structure

Now, we present the ASLL's structure which is presented in EBNF as follow:

<adaptive_storyline_model> ::= <start> <storyline_flow> <end>.

The *adaptive storyline model* entry is defined by three entries: *start* entry which is used to state the beginning of the storyline, storyline flow which includes topics and the associated storyline adaptation rules, and end entry to state the end of the storyline.

```
<start> ::= ‘Start’ [<notification_message>].
<end> ::= ‘End’ [<notification_message>].
<notification_message> ::= <text_message> [<audio_message>]
<text_message>::=<string>
<string> ::= <character> { <character> }.
<character > ::= 'A' | 'B' | 'C'| 'D' | 'E' | 'F' | 'G' | 'H' | 'I' | 'J' | 'K' | 'L' | 'M' | 'N' | 'O'
| 'P' | 'Q' | 'R' | 'S' | 'T' | 'U' | 'V' | 'W' | 'X' | 'Y' | 'Z' | 'a' | 'b' j 'c' | 'd' | 'e' | 'f' | 'g' |
'h' | 'i' | 'j' | 'k' | 'l' | 'm' | 'n' | 'o' | 'p' | 'q' | 'r' | 's' | 't' | 'u' | 'v' | 'w' | 'x' | 'y' | 'z' | ‘
‘ | ‘-‘| ‘_’ .
<audio_message>::= <url>.
<url>::= ‘https://’, <urlchar>.
<urlchar>::= <digit> | <character> | <special_character>
<special_character>::=  ‘-‘ | ‘_’ | ‘.’ | ‘~’ | ‘!’ | ‘*’ | ‘'’ | ‘(‘ | ‘)’ | ‘;’ | ‘:’ | ‘@’ | ‘&’ | ‘=’ |
‘+’ | ‘$’ | ‘,’ | ‘/’ | ‘?’ | ‘%’ | ‘#’ | ‘[‘ | ‘]’.
```

Both *start* and *end* entries may include *text messages* which are defined by the author to notify learner that he begins a topic or he reached the end of the current topic. Furthermore, an audio message can also be provided along with the text message.

```
<storyline_flow> ::= <source_topic> <storyline_adaptation_rule>
<target_topic> {‘,’ <source_topic> {<storyline_adaptation_rule>
<target_topic>}} .
<source_topic> ::= <topic>.
<target_topic> ::=  <topic>.
<topic> ::=  <learning_concept> {‘,’<learning_concept>}.
<learning_concept> ::= <learningConcept_name>.
<learningConcept_name> ::= <character> { <character> }.
```

A *storyline flow* entry consists of a *source topic* (current topic). Each *source topic* entry has at least one *storyline adaptation rule*. The *storyline adaptation rule* connects the *source topic* with the *target topic* entry. Both of source and target topics consist of at least one learning concept.

We now present the storyline adaptation rules.

```
<storyline_adaptation_rule> ::= [<storyline_adaptation_rule_name>] ‘IF’
<source_topic_condition> ‘AND’ <target_topic_condition> ‘THEN’
<adaptation_strategy_action>
```

A *storyline adaptation rule* has an optional name and two parts: the condition part, which is an *IF* statement that is used to determine when the rule will be applied, and the action part, which is a *THEN* statement which is used to specify the adaptation strategy that should be applied to the target topic. Their syntax is given as follows:

<storyline_adaptation_rule_name> ::= <character> { <character> }.
<source_topic_condition> ::= <topic_name> '.' <topic_attribute> <relational operator> <value>.
<target_topic_condition> ::= <topic_name> '.' <topic_attribute> <relational operator> <value>.
<topic_name> ::= <character> { <character> }.
<topic_attribute> ::= 'suitability' | 'completed' | 'knowledge' | 'difficulty'.
<relational operator> ::= 'is greater-than' | 'greater-than or equal' | 'less-than' | 'less-than or equal' | 'is' | 'is-not'.
<value> ::= <level> | <boolean>.
<level> ::= <digit> [<digit>] .
<digit> ::= "0" | "1" | "..." | "9".
<boolean> ::= 'True'| 'False'.

A condition in the *IF-statement* is composed of two sub-conditions. The first one is related to the source topic, the second condition is related to the target topic. Both conditions are formulated by specifying the topic's name, one of the associated attributes, a relational operator and values to be compared with. The list of topic attributes is presented in Table 6.2. Such attributes are used to compare them with specific values. To achieve that, relational operators are used. A relational operator can be one of the following operators; 'greater-than', 'greater-than or equal', 'less-than', 'less than or equal', 'is' (equality) or 'is-not'. Finally, a value can be a numerical value or a Boolean value depending on the type of topic attribute that is selected.

The adaptation strategy action in the THEN part, used to specify which adaptation strategy will be applied to the target topic, is defined as follows:

<adaptation_strategy_action> ::= 'APPLY' <adaptation_strategy> 'TO' <target_topic_name>.
<adaptation_strategy> ::= 'TourGuide' | 'NavigationWithRestriction' |, ...
... stands for the rest of adaptation strategies presented in section 4.2.

Topic Attribute	Type	Description
suitability	Boolean	This attribute determines when the learner is able to learn about this topic. The suitability of the topic is by default defined as the suitability of all learning concepts in the topic. However, the author is still able to redefine the suitability of a topic (using the authoring tool).
completed	Boolean	This attribute is used to know if the user has already finished the topic or not. The author may use this attribute to assure that the learner

		will not move to a new topic until he completed the current topic.
knowledge	Numerical	This attribute is related to the user's knowledge about the topic (as opposed to his knowledge about the different leaning concepts in a topic). There is, of course, a relationship between the knowledge of the individual learning concepts and the topic's knowledge level. By default, each learning concept will contribute equally to the knowledge of the topic. For instance, if a topic has four learning concepts, then the knowledge level of the topic will be increased by 25% every time the user master one of the related learning concepts. As this rule may not always be applicable, we allow the author to adapt this (using the authoring tool).
difficulty	Numerical	This attribute defines the difficulty level of a topic for the user.

Table 6.2: List of topic attributes that can be used to formulate the condition part of the storyline adaptation rule

6.5.1.3 ASLL Example

The following is an example related to an adaptive storyline for a course about the solar system. The example is depicted in Figure 6.5.

As a first step in the storyline, all learners need to start learning about stars. One way to notify a learner to start learning about stars is to display a text message or/and play an audio file. Such specifications are defined in the start node of the storyline. Moreover, let us assume that the author also wants that all learners should be directed towards the stars by applying the *TourGuide* adaptation strategy. This provides a tour through the 3D learning objects related to the topic named "*Learn about Stars*". As can be expected, the purpose of this step is to provide the learners an overview about essential facts concerning the stars in the solar system like their visual representations, physical locations in the virtual space, etc. before they start experiencing the virtual world. However, the adaptation flow inside the topic itself is specified in the adaptive topic model for the *Learn About Stars* topic (see next section).

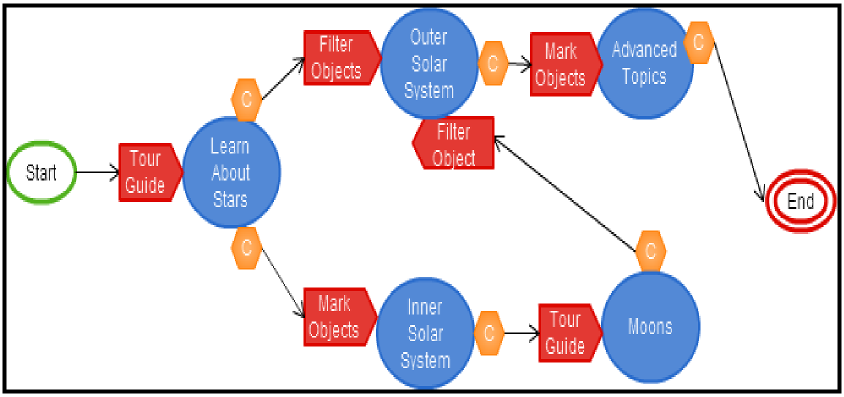


Figure 6.5: Adaptive Storyline for a 3D VLE Solar System Course

After mastering the first topic, the learner will be directed to the next topic, which is either the *Outer Solar System* topic or the *Inner Solar System* topic. Which one should be taken is given by means of the two *storyline adaptation rules* attached to the *Learn About Stars* topic.

In general, (an) adaptation rule(s) that is (are) attached to a topic is (are) triggered when the condition of the rule is satisfied. Let us assume that the author wants to defined the first storyline adaptation rule as follows: if the learner's knowledge about stars is above a specified value and the suitability of the topic *Inner Solar System* is true, then the *markObjects* adaptation strategy should be applied to the 3D objects related the *Inner Solar System* topic. As mentioned earlier, this adaptation strategy is basically used to highlight 3D virtual objects so that the learner will be able to identify them clearly. This rule is specified as follows:

> IF 'Learn About Stars'.knowledge greater than 90 and 'Inner Solar System'.suitability is TRUE
> THEN APPLY 'markobject' TO 'Inner Solar System'

The other storyline adaptation rule should be defined as follows: If the learner knowledge about stars is above a specific value and the suitability of the topic *Outer Solar System* is true, then the *filterObjects* adaptation strategy to display only the 3D learning objects related to this topic should be used. This rule is specified as follows:

> IF 'Learn About Stars'.knowledge greater than 90 and 'Outer Solar System'.suitability is TRUE
> THEN APPLY 'filterObject' TO 'Outer Solar System'

Normally, the conditions of different storyline adaptation rules associated with the same source topic should not be satisfied at the same time. However, if such case happens then it is the adaptive engine's task to select and apply one of the triggered rules.

Now, let us assume that the learner has been directed to *Inner Solar System* topic. In the same way as in the previous step, the learner needs to master the inner solar system topic by experiencing different related 3D virtual object. Then, a storyline adaptation rule is defined to direct learner toward the *Moons* topic in which he will be able to learn about the different moons in the solar system. The condition of this rule is related to the current learner's knowledge of *Inner Solar System* and the suitability of the *Moons* topic. A satisfied condition will apply the adaptation strategy *TourGuide* to the 3D learning objects related to the *Moons* topic.

After that, once the learner satisfies the condition of the rule associated with the topic *Moons*, the learner will be able to proceed towards the 3D learning objects related to *Outer Solar System*. The author has used the *filterObjects* adaptation strategy to provide the 3D virtual objects in the *Outer Solar System* topic.

Finally, once the rule associated with the *Outer Solar System* topic is triggered (e.g., which could satisfy a condition like the learner's knowledge about the *Outer Solar System* topic is high enough and the suitability of the *Advanced Topics* topic is true), an adaptation strategy will be applied to direct the learner's attention towards the *Advanced Topics* topic by highlighting the 3D learning objects to be learned. Here, the selected adaptation strategy is *markObjects*. As mentioned earlier, it is used to highlight the 3D learning objects to be learned. This is the last topic; when the last adaptation rule is triggered, the learner will be directed to the end of the storyline. Of course, the author can associate a text message to be displayed to learners at the end of the storyline.

The next section presents the Adaptation Topic Language (ASL), which allows designing the adaptivity in a topic and both start and end of the storyline.

6.5.2 Adaptive Topic Language (ATL)

Using ATL, the author can specify the adaptivity within a single topic. The work here is inspired by the Conceptual Adaptation Model presented in (Hendrix et al., 2008), which has been proposed in the context of adaptive hypermedia authoring tool, particularly in GRAPPLE. This conceptual adaptation model is used for authoring adaptations to contents and navigation by means of a graphical and user-friendly interface.

Based on the proposed design guidelines in (Karsai et al., 2009), we also followed the same approach in GRAPPLE and support the authors with a visual language that allows them to connect learning concepts related to the topic with adaptation rules that apply adaptation types (defined in section 4.3) to different learning concepts.

Next to the DSML guideline that advices to compose existing languages where possible, another important guideline followed is "*reflect only the necessary domain concepts*" (Karsai et al., 2009), meaning that the language should only use the domain concepts and their essential properties that are contributing to the authoring task rather than showing all the domain concepts.

In addition, another important guideline is "*use the same style everywhere*" (Karsai et al., 2009). For instance, in both PML and ATL, the learning concept is represented by a rectangle. Moreover, learning concepts are connected by arrows that involve the rules to be evaluated (this is done in all our visual languages). Moreover, the condition notation is the same in both ASLL and ATL. Using the same visual notation to represent the same concepts can decrease the cognitive load and increase the languages understandability. This aspect is referred as consistency (Paige et al., 2000).

Similar to the previous section, the next subsections will describe the adopted visual representation and their meanings. After that, the structure of the ATL and how the adaptation rules are formulated are discussed. Finally, an example using ATL is presented.

6.5.2.1 Visual Symbols

ATL's graphical symbols are shown in Figure 6.6. Firstly, a set of learning concepts is composing a topic. A learning concept is represented as a rectangle including its name (see Figure 6.6(A)). In addition, learning concepts are connected with other learning concepts through so-called *topic adaptation rules,* which consists of four symbols (Figure 6.6(B)). As the topic adaptation rule is an *event-condition-action* mechanism, we opt to have different visual notations for the different parts, i.e., a visual notation for the event part, a visual notation for the condition part, and a visual notation for the action part. Moreover, there are the source and target learning concepts that are required to show the flow of adaptation from one learning concept to another.
Next, we describe the four visual notations related to the topic adaptation rule.

The first visual notation is the *VR event* notation (mandatory). A VR event is graphically represented as a diamond (see Figure 6.6(C)). It is used to represent when the condition part needs to be evaluated, i.e. when the VR event occurs for the source learning concept. A list of possible VR events is mentioned in next section.

The second visual notation is the *condition* notation (mandatory). It is represented as a hexagon titled with the letter *C* (see Figure 6.6(D)). In addition, the condition symbol can be annotated with the actual condition to make the graphical model more readable and to give more information in the graphical model itself. The condition must be satisfied in order to apply the adaptation type.

The third visual notation is the *adaptation type* notation (mandatory). It has two variants. The first notation is an arc (Figure 6.6(E)) and the second one is an arrow (Figure 6.6(F)). The arc is used to specify adaptation types to be applied to the same learning concept (both source and target learning concept refer to the same learning concept). Such notation can for instance be used to require a new 3D resource (for displaying the learning concept). An arrow is used to represent an adaption type applied to a learning concept (the target) that is different from the source learning concept. The arcs or arrows are labelled with the selected adaptation type. For instance, when the *hide* adaptation type is used, the arrow/arc is labelled *hide*. Therefore, the author is able to know which adaptation types will be applied.

The fourth visual notation is for the *notification message* (optional). It is represented as a cloud symbol (Figure 6.6(G)). It is used to provide textual information along with an auditory or visual notification (optional) to the learner when the adaptation rule is activated. The notification message is optional, which means it is not necessary to have a message for all adaptations. Presenting feedback or hints to the learner may affect the learning outcomes in different ways (Razzaq & Heffernan, 2010). However, we believe that it should be left to the authors when and how such notification messages should be provided. Note that a hint mechanism was already proposed in the context of ELEKTRA project, which supports adaptation also in the content of the learning hints based on the individual skill level of the learner (Koidl et al., 2010).

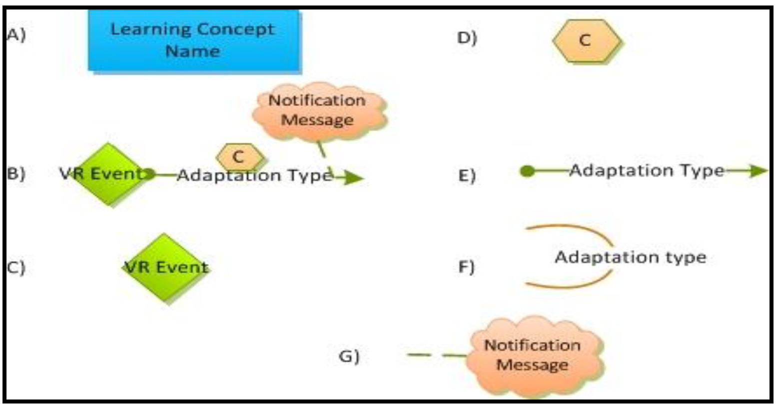


Figure 6.6: ASL elements: A) learning concept, B) Adaptation type to be applied on a 3D learning concept, C) Adaptation type to be applied to a 3D learning Concept, D) VR event E) Text Message.

Two examples of how to combine the previous symbols to formulate an adaptation rule are given in figure 6.7. The first adaptation rule (see Figure 6.7(A)) is used between Sun and Earth. The VR event *close to* is used to trigger the associated adaptation rule. The condition symbol is associated with the condition. The adaptation type to be applied to the target learning concept

(Earth) is *Display*. The second adaptation rule (see Figure 6.7(B)) is used on a single learning concept (Earth). The rule has a *touch* VR event and uses the *disable_interaction* adaptation type to disable user interaction with the Earth object if the learner has already interacted with it many times. In both cases, a notification message is provided.

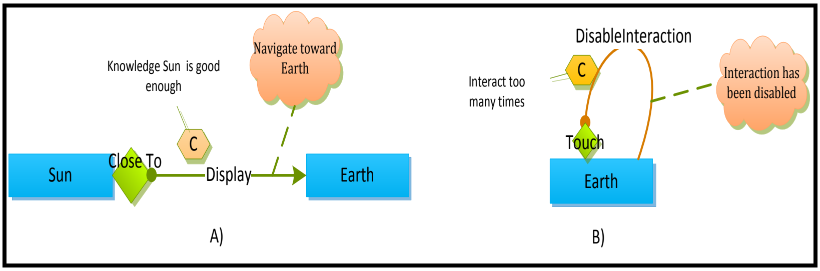


Figure 6.7: Examples of adaptation rules: A) adaptation rule between two learning concepts; Sun and Earth, B) adaptation rule on a single learning concept.

The next section, the structure of ATL is presented using EBNF. After that, a simple example illustrating the use of ATL to define adaptations in the *Inner Solar System* topic is presented.

6.5.2.2 Language Structure

Turning to the structure of ATL, the following EBNF specifications define the language structure:

<adaptive_topic_model> ::= <source_learning_concept> {<topic_adaptation_rule> <target_learning_concept>} {',' <source_learning_concept> {<topic_adaptation_rule> <target_learning_concept >} }

An *adaptive topic model* entry is defined by three parts: *source learning concept* part, *topic adaptation rule* part, and *target learning concept* part.

<source_learning_concept> ::= <learningConcept_name>.
<target_learning_concept> ::= <learningConcept_name>.
<learningConcept_name> ::= <character> { <character> }.

The two learning concepts are connected to each other by an adaptation rule. So there are a source learning concept and a target learning concept. A source learning concepts can be connected with a target learning concept by at least one *topic adaptation rule*.

<topic_adaptation_rule>::= 'ON ' <active_vrevent> 'DO' 'IF' <condition> 'THEN' 'APPLY' <adaptation_action> [<notification-_message>].

A topic adaptation rule has three parts: the *event* part, which is used to determine which VR activities will trigger the topic adaptation rule; the *condition* part, which is formulated from conditions based on user model attributes (user profile and 3D activities); and the *action* part, which is used to determine the adaptation action that should be performed. Figure 6.8 depicts the three parts of the topic adaptation rule.

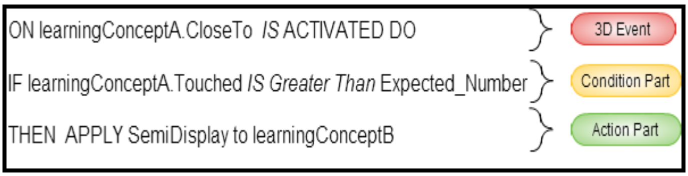


Figure 6.8: An example of topic adaptation rule.

<active_vrevent>::= <learningConcept_name> '.' <vrevent> 'IS ACTIVATED'.
<vrevent>::= 'Timer' | 'CloseTo' | 'Collison ' | 'StringEntered ' | 'Touch' | 'KeyPressed'.

The event part is specified by specifying the VR event that should trigger the rule associated with the source learning concept. The list of VR events is shown in Table 6.3.

<condition> ::= <learning_concept_name> '.' (<user_profile_attribute> | <3Dactivity_attribute>) <relational operator> <value>.

<user_profile_attribute> ::= 'knowledge' | 'suitability' | 'interest' | 'visited'.
<3Dactivity_attribute> ::= 'timer' | 'closeTo' | 'collison ' | 'stringEntered ' | 'touch' | 'keyPressed'.
<relational operator> ::= 'greater-than' | 'greater-than or equal' | 'less-than' | 'less-than or equal' | 'is' | 'is-not'.
<value> ::= <level> | <boolean>.
<level> ::= <digit> [<digit>] .
<digit> ::= "0" | "1" | "..." | "9".
<boolean> ::= 'True'| 'False'.

A condition of the topic adaptation rule is formulated as a condition on the user profiles attributes (Table 6.1) or the 3D activity attributes (see Table 6.3) related to the source learning concept.

<adaptation_action> ::= <adaptation_type> 'TO' <learning_concept_name>.
<adaptation_type> ::='hide' | 'semiHide' | ....
.... represents the full list presented in section 4.3.

The adaptation action in the rule can be one of the adaptation types defined in section 4.3. This adaptation action will be applied to the 3D object which corresponds to the target learning concept.

<notification_message> ::= <text_message> [<audio_message>].
<text_message>::=<string>.
<string> ::= <character> { <character> }.
<audio_message>::= <url>.
<url>::= ‘https://’, <urlchar>.
<urlchar>::= <digit> | <character> | <special_character>
<special_character>::= ‘-‘ | ‘_’ | ‘.’ | ‘~’ | ‘!’ | ‘*’ | ‘’’ | ‘(‘ | ‘)’ | ‘;’ | ‘:’ | ‘@’ | ‘&’ | ‘=’ | ‘+’ | ‘$’ | ‘,’ | ‘/’ | ‘?’ | ‘%’ | ‘#’ | ‘[‘ | ‘]’.

A notification message includes a *text message* that will be displayed to the learner to inform him about the applied adaptation type. However, an audio message can also be provided along with the text message.

Event	Type	Description
Timer	Numerical	Generates an event after a predefined time. For instance, when a course author likes to deactivate a 3D object behaviour after it has been activated for a specific period of time.
CloseTo	Boolean	Generates an event when the user enters, exits or moves within a region (defined by a box) around a 3D object.
Collision	Boolean	This event is triggered when an object collision is detected.
StringEntered	Numerical	Generates an event when the user entered a (predefined) string.
Touch	Numerical	Detects when the user touches (e.g., by clicking with the mouse) a learning object.
KeyPressed	Numerical	Generates an event when the user presses a key from the keyboard.

Table 6.3: List of VR event that are recorded in the user model (3D activity history model).

6.5.2.3 ATL Example

A simple example of using ATL is given in Figure 6.9. Back to the example given in previous section, one of the topics that could be followed by the learner is the *Inner Solar System* topic. This topic contains all the learning concepts representing the inner solar system such as *Sun, Mercury, Venus, Mars* and *Earth*. As it was specified in section 6.5.1.3, an adaptation strategy (*markObjects*) is already used to highlight all related 3D objects.

In this model, it is specified how the adaptation should be performed for these learning concepts within this topic. .

To inform the learner where to start, a timer VR event is used along with adaptation type to semi-hide the Sun and a text message will be displayed to the learner to ask him to navigate toward Sun and to start interacting with it. After that, if the user touches the Sun (captured by a VR event), then the condition parts of two associated adaptation rules, which are connected to Venus and respectively Mercury, will be checked. If one of the conditions becomes true the corresponding action part (adaptation type) will be executed. Suppose that the condition of the adaptation rule between the Sun and Mercury becomes true (the user's knowledge about Mercury is not good enough and he already interacted with Sun a couple of times) then an annotation with be added to Mercury's 3D object using the *displayAnnotation* adaptation type. Furthermore, a text message will be shown to the learner asking him to navigate toward the annotated object (Mercury). However, if the condition part of the other adaptation rule (between Sun and Venus) is true then the *displayAnnotation* adaptation type will be applied to Venus's 3D object.

Now, let us consider that the triggered rule asked the learner to navigate toward Mercury. Once the learner arrived at Mercury's 3D object, he will be able to interact with it and to view its rotation around its axis and around the Sun. To stop Mercury's rotation around the Sun, the author has used an adaptation *disablebehavior-rotationArroundSun* to disable the behaviour combined with a timer VR event and a condition about the learner's knowledge. This adaptation rule is used as rotation behaviour may distract learner from the other learning tasks. To inform the learner about the fact that the rotation was disabled, the author can use a notification message to be displayed when the adaptation type is triggered. Similarly, the *disableInteraction-mouseClicks* adaptation rule is used to disable interaction when the learner has already interacted enough with Mercury, and a text message will be displayed to inform him that interaction with Mercury is disabled.

If the learner interacted with Mercury's 3D object a couple of times and he has the required level of knowledge about Venus, then the annotation text related to Venus will be displayed (according to the adaptation rule between Mercury and Venus). Similar to Mercury, the author has specified an adaptation rule to disable Venus' rotation around its axis after a predefined time (by the adaptation *DisableBehaviour-rotationArroundItself)*. When the student's knowledge about Venus is above the specified threshold and the learner was *close to* Venus different times then Mars will also be annotated by applying the annotating adaptation type. Likewise, when the student's knowledge about Mars reaches a certain level, a text message will be presented to the student to let him advance towards Earth and it will be also annotated.

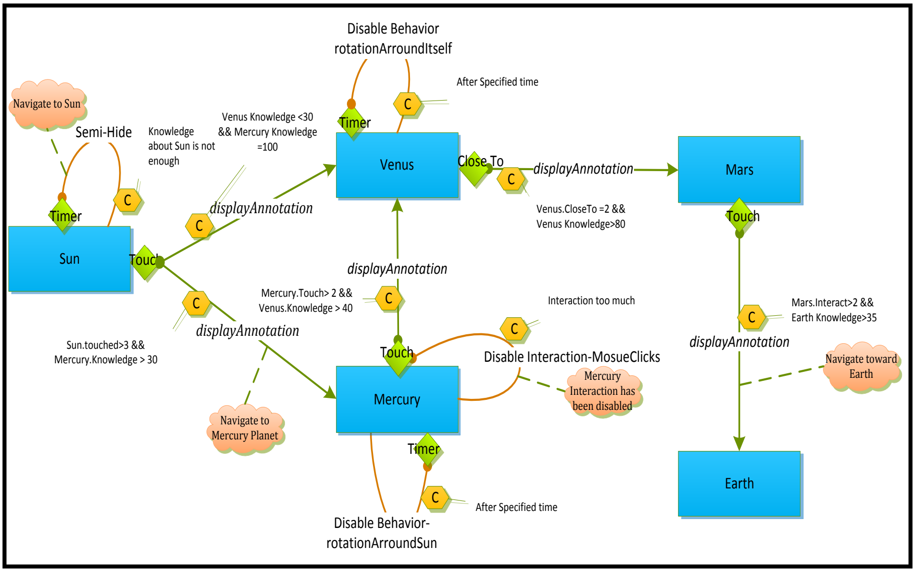


Figure 6.9: Adaptation Topic Model for Inner Solar System Topic.

6.6 Summary

This chapter has formulated important functional and usability requirements to be taken into account for authoring adaptive 3D VLE. Next, the concept of adaptation theme is presented to satisfy one of the usability requirements, i.e. allowing creating adaptations in a consistent way.

This chapter has also described one of the major contributions of this thesis: the three visual languages defined. The purpose of the three languages is to give the authors the possibility to design *adaptive* 3D virtual *learning* environments. These languages are the pedagogical model language (PML), the adaptive storyline language (ASLL), and the adaptive topic language (ATL). The chapter described the visual notations and the syntax of these three languages. The languages are represented as graph-like diagrams.

The pedagogical model language (PML) enables authors to define pedagogical aspects. PML allows connecting learning concepts to each other via so-called *pedagogical relationship types* (PRTs). PRTs, such as *prerequisite*, *updates-knowledge-of* are used to define how the learning concepts are related and how the learning process knowledge in the user model should be updated. For that, a *pedagogical update rule* is associated with a PRT to specify which user model attributes needs to be updated and how.

With regard to the adaptive storyline language (ASLL), the authors are able to define the storyline (or learning path) that the learner needs to follow. The language consists of a *start* and *end* element which are used to notify when the storyline starts and when it ends respectively. Furthermore, it consists of *topics,* which include group of related learning concepts. Each topic is connected with other topics using *storyline adaptation rules,* which are used to define when the learner will be able to move from one topic to another adaptively. These rules are using a condition-action mechanism.

The adaptive topic language (ATL) enables the authors to define the actual adaptation behaviour for the 3D materials associated to the learning concepts related to each topic. This is done by connecting the learning concepts via *topic adaptation rules*. Such a rule is using an event-condition-action mechanism. Furthermore, the rule can also be associated with a notification message to inform the learner about what happened or what he has to do next.

A note to make is that the User Model is, to some extent, considered as a black box. We supposed that there exist some services that support the adaptive engine with getting the required attributes values.

Chapter 7: Authoring Adaptive 3D VLE - Demonstration

In this chapter, we illustrate the proposed authoring approach and more in particular its modelling languages. We demonstrate the languages through a course about the solar system for first year undergraduate students. This course tries to convey the solar system using 3D models represents the different bodies relate to the 3D solar system.

This chapter is structured as follow: Section 7.1 elaborates on the course content. Section 7.2 describes the different models created during authoring. Section 7.3 shortly describes the course as implemented using the GRAPPLE adaptive engine. Section 7.4 summarises the chapter.

7.1 Course Description

Based on the presented examples in section 6.4.1, section 6.5.1.3, and section 6.5.2.3, a full example of authoring a solar system course as an adaptive 3D VLE will be explained further. In particular, the demonstration presented here shows how the author can create an adaptive 3D solar system course using the Pedagogical Model Language, the Adaptive StoryLine Language, and the Adaptive Topic Language.

We see two options to present a course about 3D solar system. The first option, we will call the *Hypermedia 3D Solar System* course. In this option, the course is a combination of textual learning material about the solar system, presented as regular hypermedia, and a 3D VLE of the solar system where the 3D models are presented. The textual material is displayed in an area of web page (e.g., a frame) where the learner can navigate through the textual material using standard navigation, i.e. menus and hyperlinks. The 3D VE is displayed in another area of the same web page where the user is able to navigate through the 3D virtual environment. The second option, we will call the *Pure 3D Solar System* course and uses only a 3D environment with 3D models to teach students different aspects related to the solar system. However, the author is still able to include textual information and audio messages inside the 3D virtual environment.

Concerning the first option (the *Hypermedia 3D Solar System* course), authoring adaptation to the text material can be done using authoring tools such as AHA!, or GRAPPLE. Concerning designing adaptation inside the 3D VLE (the 3D VE part of the *Hypermedia 3D Solar System* course or the *Pure 3D Solar System* course), the author can use our proposed authoring approach.

Further sections will focus on providing adaption to the 3D VLE. An example that integrates both hypermedia and 3D models in the same web page will be provided in section 7.3. This example is presented to show the possibility of the integration of both hypermedia and 3D models in the same course.

Note that the 3D VLE itself has to be designed using tools such as X3D Editor[56], BS Editor[57], SwirlX3D Editor[58], or Flux Studio[59]

We want to structure the 3D solar system course into five topics: *Learn About Stars*, *Inner Solar System*, *Outer Solar System, Moons,* and other small-body populations including asteroids, comets, dwarf, and satellite. (*Advanced Topics*). Each topic includes a number of learning concepts as follows: *Learn About Stars* consists of the *Sun* as it is the biggest star in our solar system; *Inner Solar System* includes the planets *Mercury, Venus, Earth,* and *Mars*; *Outer Solar System* consists of the planets *Jupiter, Saturn, Uranus,* and *Neptune*; *Moons* topic includes *Moon* (moon of Earth), *Ganymede* (moon of Jupiter), *Callisto* (moon of Jupiter), *Io* (moon of Jupiter), *Europa* (moon of Jupiter), *Rhea* (moon of Saturn), *Titan* (moon of Saturn), *Iapetus* (moon of Saturn), *Dione* (moon of Saturn), *Tethys* (moon of Saturn), *Enceladus* (moon of Saturn), *Mimas* (moon of Saturn), *Hyperion* (moon of Saturn), *Phoebe* (moon of Saturn), *Janus* (moon of Saturn), *Epimetheus* (moon of Saturn), *Prometheus* (moon of Saturn), *Titania* (moon of Uranus), *Oberon* (moon of Uranus), *Umbriel* (moon of Uranus), *Ariel* (moon of Uranus), *Miranda* (moon of Uranus), *Triton* (moon of Neptune) and *Proteus* (moon of Neptune); *Advanced Topics* includes *Halley's Comet, Ceres dwarf,* and *Vesta asteroid.*

The main goal of the 3D VLE is to allow learners to (1) interact with the solar system bodies, (2) view the 3D solar system with different properties (which is not possible in a physical setting), (3) navigate through the 3D VLE to get a feeling about the physical location and distance between the different solar system bodies.

The authoring scenario that we used here is based on the one presented in section 6.1.1, and used to establish our authoring approach. The authoring scenario is as follows:

[56] www.web3d.org/products/detail/x3d-edit/
[57] http://www.bitmanagement.de/products/bs_editor.en.html
[58] http://www.pinecoast.com/swirl3d.htm
[59] http://mediamachines.wordpress.com/flux-player-and-flux-studio/

The author first creates the pedagogical relations between the different learning concepts involved in the topics. For instance, he wants to define the prerequisite relationship between *Sun* and the different planets.

Next, the author can define a specific storyline for the course. As explained, a course is divided into different topics, and each topic deals with some learning concepts from the domain model. Linking the topics together gives the main storyline, which is provided adaptively according to the learner knowledge about a topic and a topic's suitability. The author also may want to define the criteria on which the learner's knowledge about a topic and the topic suitability can be updated.

As mentioned in section 6.1.1, different 3D resources can be associated to a learning concept. For instance, the author may need to associate different representations of earth with the learning concept *Earth*. For instance, the different 3D resources of *Earth* may have different level of details, which can be discovered by the learner step by step. Each resource can be displayed adaptively depending on the learner activities inside the 3D VLE.

The author would like to provide a 3D solar system that shows the actual behaviour of the different celestial bodies, such as the rotation of the planets around the sun. Moreover, the author wants that the learners are able to navigate through the 3D VLE, interact with the 3D objects representing the solar system bodies.

The next section details how the adaptive 3D solar system course has been designed using the Pedagogical Model Language, the Adaptive StoryLine Language, and the Adaptive Topic Language.

7.2 Authoring the 3D Solar System Course

The three visual languages are intended to provide an easy-to-use and understandable representation of the most complex elements of the authoring of the adaptive 3D VLE. The pedagogical base for the adaptive 3D VLE is represented visually to define educational dependencies between learning concepts. The storyline representation provides a description of the course's learning path, while the topic descriptions focus on user interaction and adaptation of learning concepts.

7.2.1 The Pedagogical Model

Creating the Pedagogical Model is achieved by using the Pedagogical Model Language (PML), which presents the learning concepts as blue rectangular shapes and the pedagogical relationship types as directed arrows having different colours. Figure 7.1 gives the Pedagogical Model for the 3D solar system course.

According to the authoring scenario explained in previous section, the learning concept *Sun* is a *prerequisite for* the planets (learning concepts):

Mercury, Venus, Mars, *Earth*, *Jupiter, Saturn, Uranus* and *Neptune* (see Figure 7.1). This means all planets should be learned after obtaining the required knowledge about *Sun*. This constraint will be achieved by using the default associated pedagogical update rule that turns the suitability of the different planet learning concepts to TRUE once the learner has a good knowledge about the learning concept *Sun.*

Furthermore, when the learner learns about *Mercury*, his knowledge about *Venus* will also be increased. This is specified by putting the *update-knowledge-of* pedagogical relationship between *Mercury* and *Venus*. Likewise, this relationship (*update-knowledge-of*) is also applied to other concepts: *Venus, Mars, Earth*, etc. This PRT has a default associated pedagogical update rule that updates the learner knowledge after a specific number of visits to the learning concept, which is applicable for our purpose.

Furthermore, planets such as *Earth*, *Jupiter, Saturn, Uranus,* and *Neptune* have at least one moon. Therefore, an *interesting-for* pedagogical relationship type can be used between planet and their corresponding moon learning concept to switch the interest to TRUE after visiting the corresponding planets (as it is defined in the associated pedagogical update rule).

Finally, *Proteus* moon (Neptune's moon) has a *prerequisite for* relationship with *Halley's Comet*, *Ceres dwarf,* and *Vesta asteroid*. Therefore, the learner will not be able to learn about them till he gets the required knowledge about *Proteus*.

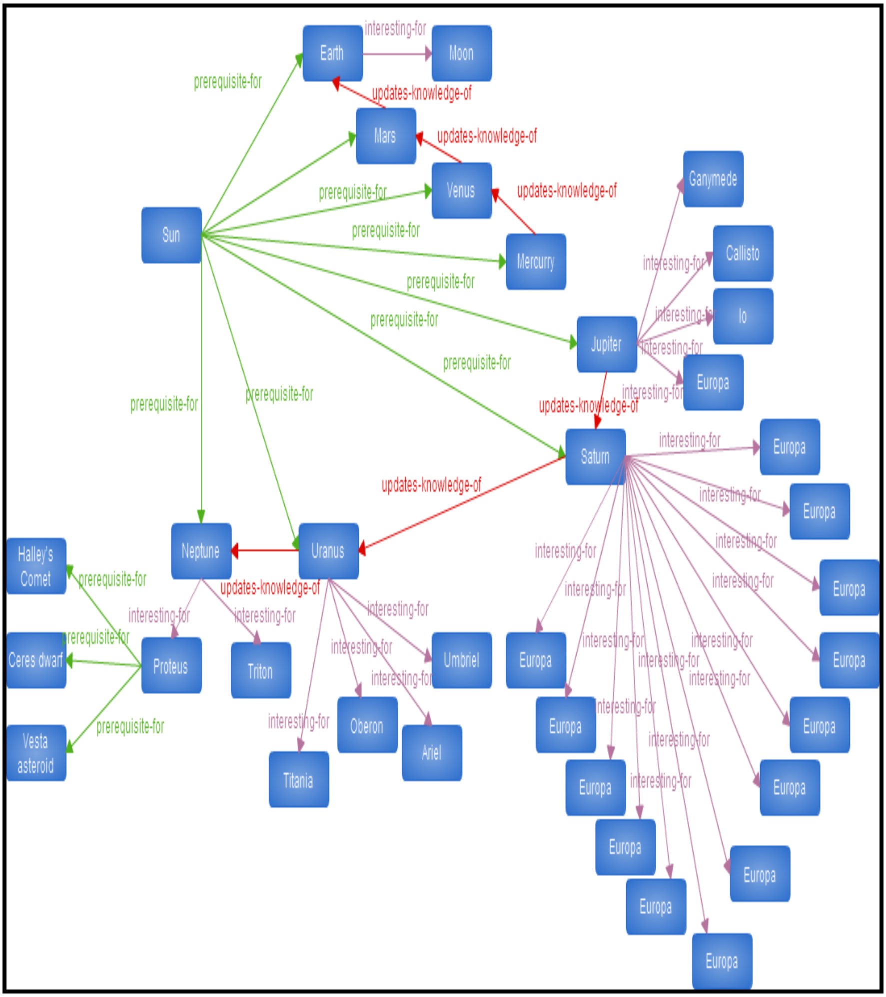


Figure 7.1: Pedagogical Model for 3D solar system.

7.2.2 The Adaptive Storyline Model

To create the Adaptive Storyline Model for the adaptive 3D solar system course, the Adaptive StoryLine Language is provided to allow for the definition of the topics which compose the adaptive storyline. As already described, the topics are represented as blue filled ellipse. The topics can be linked to each other by the storyline adaptation rules that control how the learner can move from one topic to another. A storyline adaptation rule is composed of a condition represented as an orange hexagon shape and an adaptation strategy represented as a red pentagon shape.

The author can also define an adaptation theme to be used throughout the 3D course. These specifications can be defined using the authoring tool (see section 9.5.4).

Below, we will explain the use of *Adaptive StoryLine Language* to create a storyline for the 3D solar system course.

The author specifies a text message to be displayed to inform the learner that he should start learning about the topic *Learn about Stars* topic which includes the learning concept *Sun.* Such specification is defined in the start node of the storyline (see Figure 7.2). Moreover, let us assume that the author also wants that the learner should be directed towards the *Sun* by applying the *TourGuide* adaptation strategy. This is used to provide information about the *Sun* like visual representations and physical location in the virtual space.

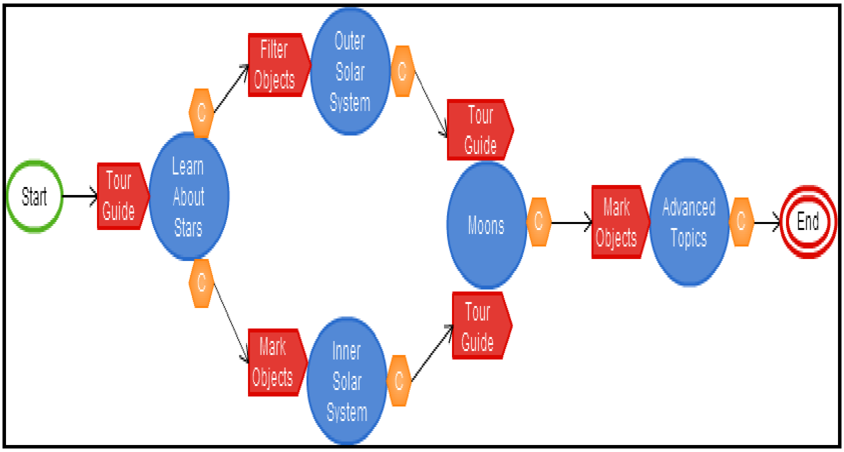


Figure 7.2: Adaptive storyline for a 3D VLE Solar System Course.

After that, the author allows the learner to move from the *Learn about Stars* topic to another topic. This can be the *Outer Solar System* topic or the *Inner Solar System* topic. To direct the learner to the appropriate topic, the author creates two storyline adaptation rules, both attached to the *Learn About Stars* topic. The first rule specifies that if the learner's knowledge about the *Learn About Stars* topic is above a specific value and the difficulty level of the *Outer Solar System* topic is less than a specific value, then the learner is (only) allowed to move to the *Inner Solar System* topic and the *markObjects* adaptation strategy should be applied to the 3D objects related to the *Inner Solar System.* The second rule specifies that if the learners' knowledge about *Learn About Stars* is above a specific value and the suitability of the *Outer Solar System* topic is true, then the learner is allowed to move to the *Outer Solar System* topic and the *filterObjects* adaptation strategy is applied to display only the 3D objects related to the *Outer Solar System*.

Next, after mastering the *Inner Solar System* topic or the *Outer Solar System* topic, the author wants to direct the learner to the *Moons* topic. The *Moons* topic consists of all moons related to *Earth, Jupiter, Saturn, Uranus,* and *Neptune*. The author does this by creating storyline adaptation rules that he associates with the previous topics (*Inner Solar System* topic and the *Outer Solar System*). The condition of the rule associated with *Inner Solar System* topic checks if the learner's current knowledge level about the *Inner Solar System* topic is sufficient and the suitability of the *Moons* topic is true. Then, the *TourGuide* adaptation strategy will be applied to the learning concepts in the *Moons* topic. The other storyline adaptation rule, which is associated with the *Outer Solar System* topic, evaluates whether the current topic (*Outer Solar System*) is completed and whether the suitability of the *Moons* topic is true. Then, the *TourGuide* adaptation strategy will be applied to learning concepts related to the *Moons* topic.

Finally, the last rule associated with the *Moons* topic (e.g., which could test a condition like whether the suitability of the *Advanced Topics* topic is true), will control whether the learner can progress to the last topic, the *Advanced Topics* topic. If so, an adaptation strategy (*markObjects*) will be applied to direct the learner's attention towards the *Advanced Topics* learning concepts by highlighting *Halley's comets, Ceres dwarf*, and *Vesta asteroid*. Since this is the last topic, after finishing this topic, the learner will be directed to the end of the storyline. The author can associate a text message to be displayed to the learners to conclude the course.

7.2.3 The Adaptive Topic Model

Specifying the topics is achieved by using the *Adaptive Topic Language;* this language allows defining the adaptive behaviour concerning the learning concepts related to a specific topic.

We present the following Adaptive Topic Models:

<u>*Learn About Stars*:</u>
As specified in the storyline (see previous section), the first topic is *Learn About Stars*. To inform the learner where to start, the author uses a *timer* VR event to apply the *mark* adaptation type to the *Sun* (see Figure 7.3). Also a text message will be displayed to the learner to ask him to navigate toward *Sun* and to start interacting with it.

Another adaptation rule is used to display an annotation once the learner comes close to the *Sun*. This is specified in an adaptation rule that is triggered by the *CloseTo* VR event.

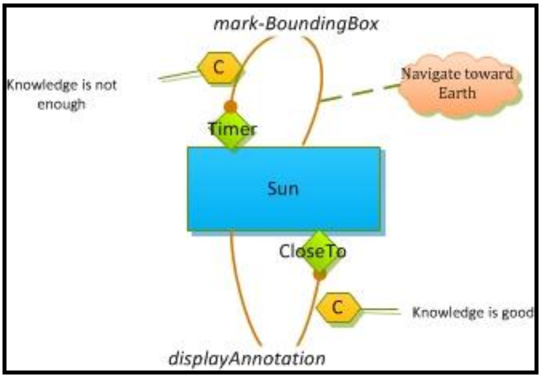


Figure 7.3: Applying adaptation types to the Sun.

As there is only one learning concept in this topic, the learner's knowledge about the *Learn About Stars* topic is increased by visiting the Sun (see section 9.5.9).

<u>*Inner Solar System*</u>:
The Adaptive Topic Model for the *Inner Solar System* is given in Figure 7.4. The following adaptations are defined for the learning concept *Mercury*.

A first adaptation rule uses the *timer* VR event for triggering a rule to annotate *Mercury* after the associated condition is satisfied. The condition is defined as 'If the user's knowledge about *Mercury* is not good enough and he already interacted with *Sun* a couple of times'. The action part uses the *displayAnnotation* adaptation type to display the annotation.

Another adaptation rule defined for *Mercury* is as follows. It uses the *Touch* VR event to capture the learner's interaction with *Mercury.* The condition part is used to evaluate the learner's knowledge about *Mercury*. The action part is used to enable *enablebehaviour-rotationAroundSun,* which starts the rotate of *Mercury* around the *Sun.* Also, the author has associated a text message with the rule to inform the learner that *Mercury* has started its rotation around the *Sun*.

The author also defined a third adaptation rule also triggered by the *Touch* VR event. The condition part of the rule is related to the number of user interactions with *Mercury*. The action part is the *displayAnnotation* adaptation type to display an annotation to guide the learner towards the next learning concept (*Venus*).

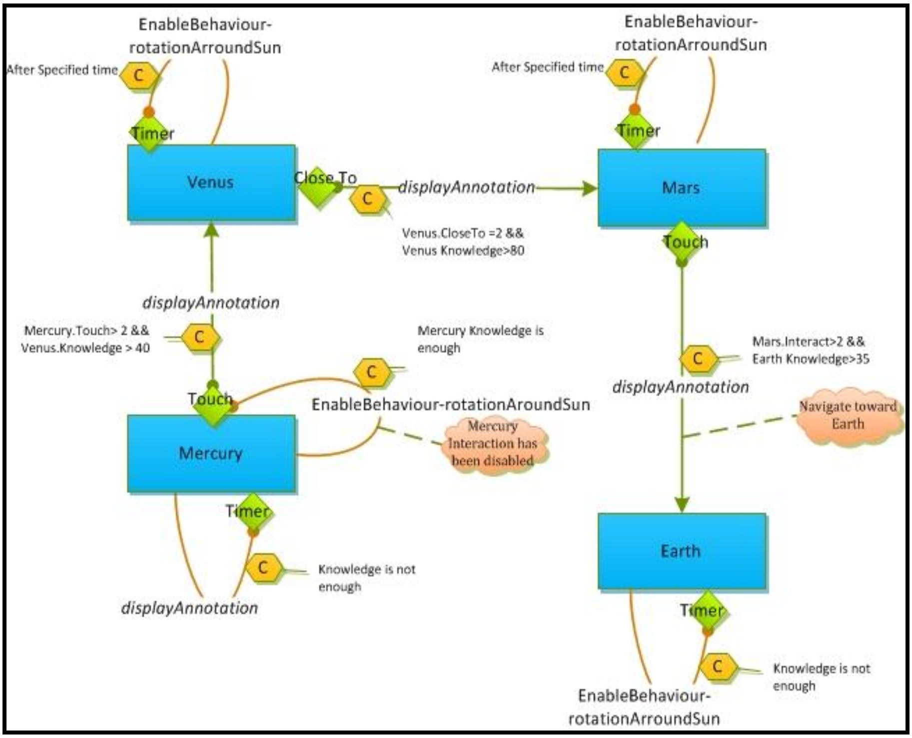


Figure 7.4: Adaptation types applied to planets related to Inner Solar System

The author also has defined similar adaptation rules to the other three planets (*Venus, Mars,* and *Earth)* (see Figure 7.4)

Outer Solar System:
The Adaptive Topic Model for the *Outer Solar System* is given in Figure 7.5. The following adaptations are defined for the learning concepts.

For *Jupiter*, the author has specified an adaptation to enable the rotation of his planet around the sun when the learner already has a good knowledge about *Sun*. For this, the author has used the *Touch* VR event to trigger the start of the rotation around the *Sun* using the *enablebehaviour-rotationAroundSun* adaptation type.

Another adaptation rule is used to semi-hide Jupiter if the learner has already visited it a couple of times. It might not be desirable to have *Jupiter* disappearing completely from the overall 3D VLE, so the author uses the *semiHide* adaptation type to keep *Jupiter* visible but with a transparent representation. Furthermore, a text message will be displayed to inform the learner that *Jupiter* is semi-hided because he has already a good knowledge about it.

A third adaptation rule is used to direct the learner to the next learning concept which is *Saturn.* The rule is triggered by the *Touch* VR event. The action is the annotation of Saturn (specified by means of the *displayAnnotation* adaptation type).

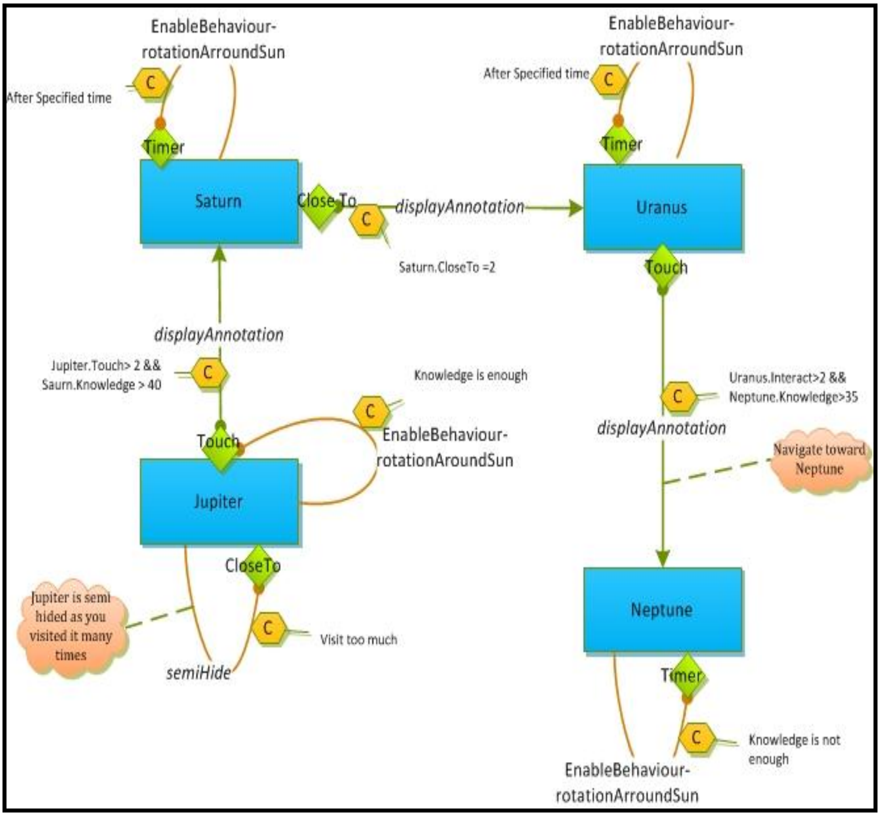


Figure 7.5: Apply adaptation types to outer solar system's learning concepts.

Similar adaptation rules are applied to the rest of the planets of the outer solar system, which are *Saturn, Uranus*, and *Neptune*.

<u>*Moons:*</u>

A snapshot of the Adaptive Topic Model for the *Moons* topic is given in Figure 7.6. The following adaptations are defined for the learning concepts.

For the *Moon,* the author has defined an adaptation rule with a *timer* VR event along with a condition about the learner's knowledge about the *Earth* planet. The action part will enable Moon's rotation around Earth by using the *enablebehaviour-rotationAroundEarth* adaptation type. Finally, a text message will be displayed to the learner to ask him to interact with *Moon*.

Another adaptation rule has been defined to deal with the learner's interaction with the *Moon*. The rule is triggered with a *touch* VR event; the condition is about the learner's interaction with *Moon;* the action part will annotate the next object, which is *Ganymede* (a moon of Jupiter) and display a text message to the learner to ask him to navigate toward *Ganymede.*

Furthermore, the author has created an adaptation rule to enable rotation around Jupiter and to display a text message to the learner to start interacting with *Ganymede*. Moreover, the author has defined another adaptation rule to annotate the next moon, called *Callisto* (also moon of Jupiter). The author has applied the same adaptation rules defined for *Callisto* than those defined for the previous moons. Similar adaptation rules can be defined for the rest of the moons: *Io, Europa*, *Rhea*, *Titan*, *Iapetus*, *Dione*, *Tethys*, *Enceladus*, *Mimas*, *Hyperion*, *Phoebe*, *Janus*, *Epimetheus*, *Prometheus*, *Titania*, *Oberon*, *Umbriel*, *Ariel*, *Miranda*, *Triton,* and *Proteus*.

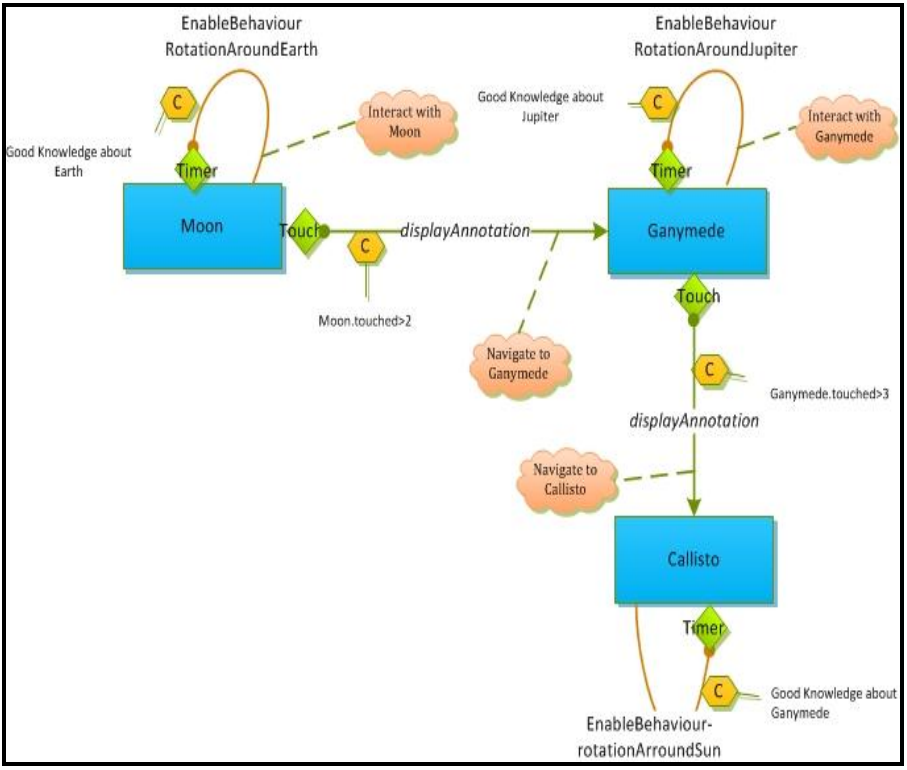


Figure 7.6: Snapshot of the Adaptive Topic Model for Moons including three moons.

<u>*Advanced Topics:*</u>
The author has defined the last topic (*Advanced Topics*) as follows:

For *Halley's Comet*, the author has defined an adaptation rule which enables its orbiting. This rule is specified by using a *timer* VR event, a condition about the learner's knowledge about *Halley's Comet*, and an action which enables this behaviour by using the *enablebehaviour-rotationAroundSun* adaptation type.

For *Ceres dwarf,* an adaptation rule with the *touch* VR event along with a condition about the number of visits, and a *displayAnnotation* adaptation type is defined*.* When this adaptation rule is triggered, a text message will be displayed to the learner to ask him to navigate toward *Ceres dwarf*.

Similar adaptation rules can also be used for *Ceres dwarf* and *Vesta asteroid*.

7.3 The 3D Solar System Course

To validate our authoring approach, the example adaptive course authored using the three modelling languages, has been implemented. For this implementation, the GRAPPLE adaptive learning engine has been used. As there is no code generation process (yet) from our visual authoring languages for the GRAPPLE engine, the course has been defined manually into the GRAPPLE format.

The 3D course allows the students to work from any location using a computer with Internet access.

Furthermore, to investigate the issues related to the combination of different types of content, this course contains text and images explaining the solar system (a hypermedia part), as well as a 3D VLE of the solar system were the sun, planets, comets, dwarfs, etc. can be experienced inside the virtual environment. All the 3D virtual objects are visible. During the learner's navigation and interactions with the 3D objects, the defined adaptive behaviour for the 3D VLE will be initiated.

Also the hypermedia part of the course is adaptive. This part of the course is presented with a classical menu in the left panel of the top frame of the browser window. If the learner clicks on the Sun-link, a textual explanation of *Sun* will be displayed in the text frame. It has been specified (using the GRAPPLE format) 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, he will be directed to the 3D VLE part (bottom frame) where a message will be displayed about the topic that he should consider inside the 3D VLE.

As it has been specified in the Adaptive Storyline Model, the first topic is about *Sun.* Therefore, the learner will be directed to a 3D model representing the *Sun* and, as specified in the Adaptive Topic Model, the 3D model will be marked by a bounding box and annotated with text. Figure 7.7 shows a screenshot for the *Sun* with the applied adaptation types (annotation and marking).

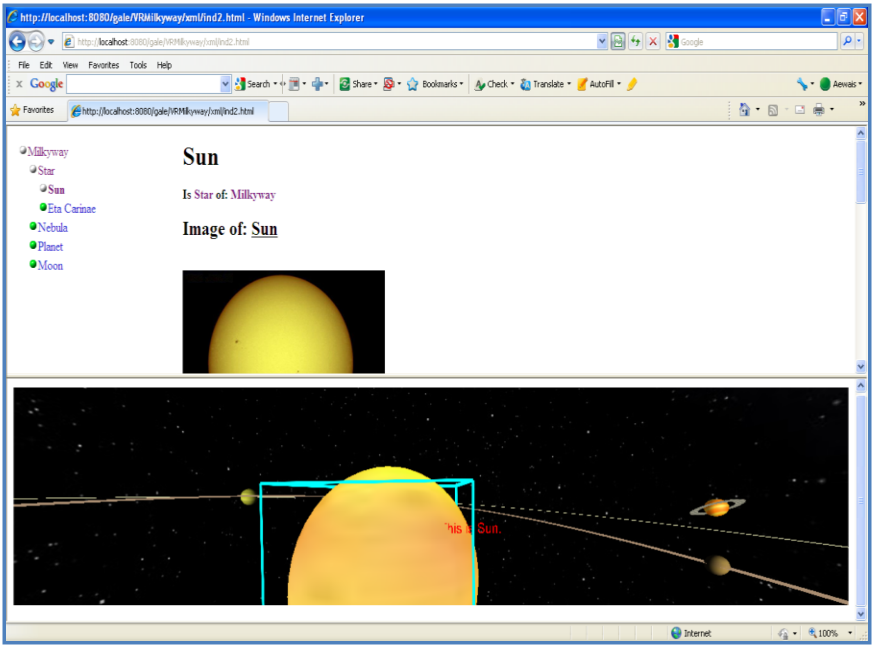


Figure 7.7: Sun is annotated and highlighted.

After acquiring a good knowledge about the topic (*Learn About Stars*), the learner is directed to a new topic which is, in this case, the *Inner Solar System* by displaying a text message about the new topic and marking the corresponding learning concepts, which are *Mercury, Venus, Mars,* and *Earth*, by bounding boxes.

Also, the *Mercury* planet will be annotated. Furthermore, the learner will be able to see the 3D object representing *Mercury,* which rotates around the *Sun* and around its axis. A text message will be shown to inform the learner about *Mercury's* rotation around the *Sun* (see Figure 7.8). The learner can proceed to the rest of the planets that are related to *Inner Solar System* and adaptation rules will be triggered to annotate them and start their rotation around the *Sun.*

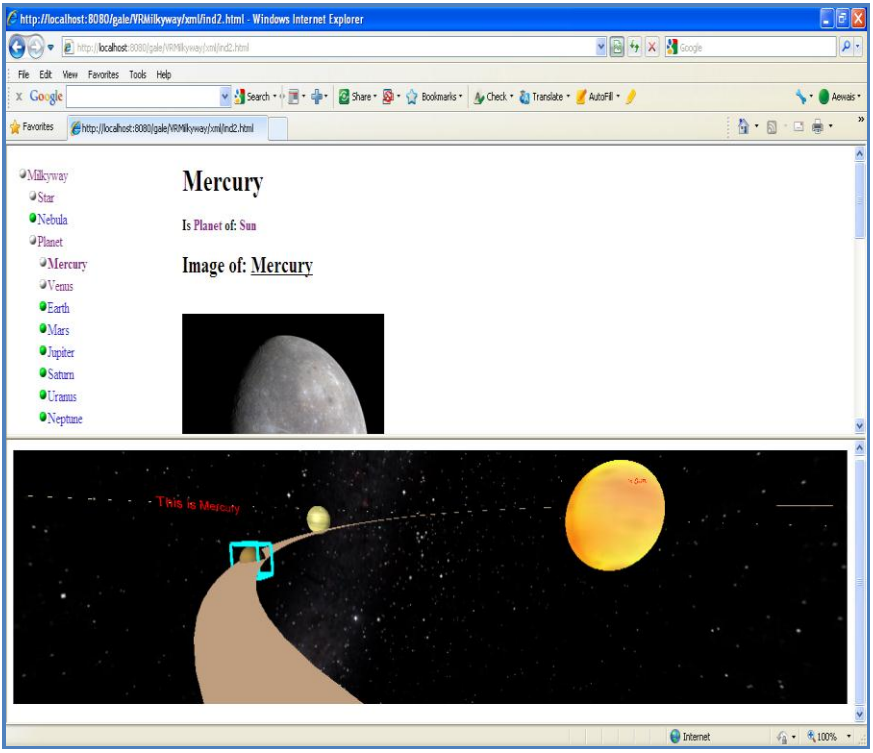


Figure 7.8: Annotating Mercury and start its rotation around Sun.

Later on, the learner may also be directed to the *Outer Solar System* topic, the *Moons* topic and the *Advanced Topics* as specified in the Adaptive Storyline Model.

7.4 Discussion

In this chapter, we have illustrated the use of the visual modelling languages by means of an example. The presented example is used to showcase their capabilities to specify an adaptive 3D VLE. The example is a web-based course for teaching undergraduate students the *Solar System*.

The 3D VLE course is divided into a number of topics, each having their own learning concepts. Particularly, the course is divided into *Learn About Stars*, *Inner Solar System*, *Outer Solar System,* and *Advanced Topics*.

The Adaptive StoryLine Language is used to define the (adaptive) learning sequence of the topics, i.e. the 3D VLE storyline.

The Adaptive Topic Language enables the specification of the 3D activities that can be performed to the 3D objects and the possible way to unfold the adaptation behaviour according to the learner inputs and interactions.

The course has also (partially) been implemented using the GRAPPLE adaptive engine.

The three modelling languages allow specifying the adaptation inside a 3D VLE. However, note that these visual languages do not guarantee that any educator can design and produce a good 3D VLE in terms of educational contents and adaptation mechanisms. It is still up to the educators to choose the educational 3D resources, compose them in the best way, and provide the required adaptation types and strategies.

Chapter 8: Evaluation

The ultimate goal of this dissertation is to give a conceptual approach for designing adaptive 3D VLE. This is achieved by providing a conceptual framework and three modelling languages: the Pedagogical Modelling Language, the Adaptive StoryLine Language, and the Adaptive Topic language. We conducted an evaluation of the proposed modelling languages to evaluate the usability, effectiveness, and expressiveness of the proposed modelling languages, as well as the workload perception.

The purpose of the evaluation was to examine if the proposed visual languages (Pedagogical Model language, Adaptive StoryLine Language, and Adaptive Topic Language) satisfy the requirements for our visual languages, such as the ability to define the pedagogical aspects of the 3D VLE, to define the storyline that the learner should follow, and to define the adaptation flow of the 3D objects inside each topic, as well as the evaluation of the graphical notations used. The next subsections present the settings of the evaluations and the results, followed by a discussion about findings, recommendations and possible improvements.

In section 8.1, the evaluation methodology is presented, which consists of a quantitative and a qualitative evaluation. Section 8.2 deals with the quantitative evaluation, while section 8.3 describes the qualitative evaluation. In section 8.4, a discussion about the results and the validity of the results is provided, and in section 8.5, a summary of this chapter is presented.

8.1 Evaluation Methodology

In general, it is good practice to use a combination of qualitative and quantitative methods to evaluate software and software approaches. Each approach has its own merits and purpose (Mandinach E., 2005). For instance, if there is a need to have reliable and objective results and outcomes, then quantitative methods are a good choice. Qualitative methods are more subjective but may reveal important information on the product, software, or approach evaluated.

For the evaluation of our visual languages, we have also followed such a hybrid approach, i.e., a combination of qualitative and quantitative methods.

First, we conducted a quantitative evaluation to gather quantitative results. To find explanations for different issues raised in the quantitative evaluation, we also performed a qualitative evaluation.

The quantitative evaluation is used to validate different usability and acceptability aspects. The usability aspects are *suitability for the task*, *self-descriptiveness,* and *conformity with user expectations,* and based on the ISONORM 9241/110-S Evaluation Questionnaire (ISONORM). The acceptability aspects are *perceived ease of use, perceived usefulness,* and *attitude,* which are mainly based on the Subjective Impression Questionnaire (SIQ). More details will be given in section 8.2.

The qualitative evaluation is used to perform a more deep analysis of the results of quantitative evaluation. This qualitative evaluation consisted of interviews with people from the educational domain, as well as from the visual modelling domain. The set of questions used, in this phase, was based on the Cognitive Dimension Framework (Blackwell et al., 2001, Green et al., 1996). The interview setup, the questions used, and the results will be discussed in section 8.3.

Important to notice is that we targeted to do a true end-user evaluation. An end-user evaluation requires the participation of end-users of the product under evaluation. End-users of our modelling languages are persons engaged in teaching activities, such as teaching assistants and lecturers, and (interesting in) using 3D and Virtual Reality (VR) for their courses. When designing the visual languages, we tried to shield as much as possible technical details, however some general background knowledge of 3D/VR is needed to understand some aspects of the languages like the adaptation strategies and adaptation types. We also expect that the visual languages require some learning time; we cannot expect a person, even not a 3D/VR-expert, to be able to create an adaptive 3D VLE using the proposed visual languages without any training in the matter of authoring adaptive 3D VLEs. Combining all these required characteristics for the participants resulted in a difficult situation, as very few persons satisfied these requirements at that moment in time. Therefore, in order to find enough participants, we had to relax our requirements. More details will be provided in section 8.2.2.

8.2 Quantitative Evaluation (Task Scenario and Questionnaire)

This section presents the quantitative evaluation performed. The goal, set-up, the task scenario, the questionnaire, and the principles for the data analysis are discussed in the different subsections (section 8.2.1 to 8.2.5). Next, in section 8.2.6, the results of the evaluation are described in detail. General information collected about the participants is given in section 8.2.6.1. Usability and acceptance results are presented in section 8.2.6.2 and section 8.2.6.3.

Qualitative feedback (also gathered in this quantitative evaluation) is presented in section 8.2.6.4, and workload perception in section 8.2.6.5.

8.2.1 Goal

The goal was to evaluate the visual modelling languages by performing a user session to measure both usability and acceptability aspects of the visual languages.

8.2.2 Set-up

The evaluation was conducted with a group of fourteen volunteers with heterogeneous background who were asked to perform a task scenario. Afterwards they had to fill in a questionnaire.

As we were also looking for critical feedback from the viewpoint of usability and user subjective satisfaction, we asked PhD students from our university, the Vrije Universiteit Brussel, and instructors from the Arab American University-Jenin (the home university of the author) to participate. In particular, four PhD candidates and researchers from the Vrije Universiteit Brussels and ten instructors from the Arab American University-Jenin were involved in the evaluation. All of them are from the Computer Science department.

Participants were informed that the evaluation result would give evidence of the quality of the provided modelling languages and would be used to reveal both advantages and disadvantages. An important aspect in this respect is the possibility of having biased answers. Although, the participants were recruited from the department of the current and home university of the author, we tried to avoid biased results as much as possible, in the following way. All participants were aware of the fact that both negative and positive results would be considered for further improvements of the proposed visual languages. Besides the fact that each participant did the evaluation individually and the results were treated anonymously, they were also informed that there were no right or wrong answers and that it is was not an evaluation of the participants themselves. Moreover, we also paid attention to the formulation of the questions, as this is also a way to avoid that participants might be encouraged to give more favourable answers. In more details, it is important to use questions such as “what do you think of …?” rather than “Why do you like this …?” as the latter question might encourage the participants to give favourable answer (Lazar, Feng, & Hochheiser, 2010, p.196). In addition, the additional qualitative evaluation conducted (see section 8.3) (with a different group of participants), can also be used to judge about the validity of the results. If the results of both evaluations are more or less consistent, then this is an indication that the results are valid.

The conducted evaluation was divided in three steps. The first step introduced the participants to the different notations of each visual language;

an example created using the visual languages; a list of Pedagogical Relationship Types (PRT) and their default associated updating rules; and a selection of Adaptation Strategies and Adaptation Types. In this way, a participant would know the goal of each PRT, Adaptation Type, and Adaptation Strategy.

Successively, in the second step, the participants had to design an adaptive 3D VLE about the Solar System using the three visual languages, i.e. perform a task scenario. We will call this the authoring exercise. More detail on the authoring exercise is presented in section 8.2.3.

Finally, in the third step, the participants were asked to fill in a questionnaire to obtain feedback about their experience with the visual languages. The survey design and the principles of the data analysis are presented in section 8.2.4 and section 8.2.5 respectively.

8.2.3 The Authoring Exercise

Each participant was given a printed document (in English), containing an explanation and figures about the *Pedagogical Model Language* (PML), the *Adaptive StoryLine language* (ASLL), and the *Adaptive Topic language* (ATL), as well as the description of the tasks to be performed. The document is presented in Appendix A.

The focus of the evaluation was on verifying whether the participants were able to specify an adaptive 3D VLE using our visual modelling languages, as well as on aspects of usability and acceptance. Because of the fact that authoring an adaptive 3D VLE is not a trivial task that can be learned in a short time, we decided to give the participants a scenario with a detailed oral guide (spoken instruction) on how to carry out the task of authoring an adaptive 3D VLE using the visual languages.

Furthermore, the participants performed their authoring task by using regular paper and pen. As it was not our purpose to evaluate an authoring tool, but rather the level of expressiveness of the visual notations of the proposed languages and the actions needed to create an adaptive 3D VLE using the visual languages, the use of pen and paper is acceptable. This approach also avoids spending a lot of resources on the development of a software tool before we received any feedback on the proposed languages. In a later stage, the conducted evaluations can be repeated using a drawing tool or an authoring tool and it can be compared with the results obtained here.

However, we also admit that using paper-pencil rather than a drawing or authoring tool also has some limitations. For instance, a software tool could guide the correct use of the languages; this cannot be achieved with pen and paper. To solve such issues, the interviewer was responsible for explaining clearly the constraints in the visual languages and for guiding and helping the participants with syntactical issues during the authoring exercise whenever required.

The stages that the participants had to follow included:

Designing the Pedagogical Model:

a) Create a Pedagogical Model for the solar system using the visual notations of the Pedagogical Model Language. Every participant could create his own Pedagogical Model.
b) Define updating rules for the Pedagogical Relationship Types. Note that this step was not required. The participant could keep the default associated updating rules.

Designing the Adaptive Storyline Model:

a) Create the Adaptive Storyline Model to be followed by the learners, using the visual notation provided by the Adaptive Storyline Language.
b) Define the storyline adaptation rules between the different topics.
c) Associate learning concepts to every single topic.

Designing the Adaptive Topic Model:

a) Create an Adaptive Topic Model for each topic by using the visual notations of the Adaptive Topic Language.
b) Define the adaptation rules required between the learning concepts.

8.2.4 Questionnaire Design

After performing the required authoring tasks, the participants had to fill in an online questionnaire. In (Slaughter, Harper & Norman, 1994), it is mentioned that using an online questionnaire can encourage participants to write and comments on open-ended formulated questions. Therefore, we used an online questionnaire instead of a paper-based questionnaire.

The questionnaire was designed to gain feedback from the participants on how they perceived the authoring approach and the visual languages. This would help us to identify usability problems and to assess whether the advantages outweigh the disadvantages.

The questionnaire was composed of six categories: Demographic Information, Authoring Adaptive 3D VLE (A3D-VLE), ISONORM 9241/110-S Evaluation Questionnaire (ISONORM), Subjective Impression Questionnaire (SIQ), Qualitative Feedback (QF), and Workload Perception (WP). The questionnaire is given in Appendix A.4. Further on is an explanation of each category.

The demographic information category is used to gather information about the participant, like age, gender, background about 3D/VR, experience in using graphical model languages, etc.

The questions related to the Authoring Adaptive 3D VLE category aims at evaluating the degree of easiness, clearness, and understanding of the authoring process. Furthermore, this questionnaire is used to detect the participants' perceptions and opinions related to applying adaptation using the visual languages.

The ISONORM 9241/110-S evaluation questionnaire (Prumber J., 1999) was used to gather the participant opinions concerning the different usability aspects of the visual languages. Furthermore, this questionnaire is used to detect strong and weak aspects related to the three visual languages and to obtain concrete suggestions for improvements. All questions in the ISONORM questionnaire were negatively formulated and they were related to four aspects of usability: *suitability for the task*, *self-descriptiveness, conformity with user expectations,* and *suitability for learning*.

Considering the different aspects related to the user acceptability according to the Technology Acceptance Model (Davis, F., 1989), a Subjective Impression Questionnaire was used to collect data on the *perceived ease of use, perceived usefulness,* and *attitude*. (Marie-louise et al., 2009) mentioned that such questionnaire can also be used in the context of e-learning applications.

Feedback and subjects' recommendations have been gathered by the questions in the Qualitative Feedback category. The questions are based on three aspects: *appreciation feedback, depreciation feedback,* and *recommendations*.

To evaluate the cognitive load, the questions used are mainly based on the NASA Task Load Index (NASA TLX) (Hart & Staveland, 1988). These questions are grouped into the Workload Perception category and include questions related to *mental demand*, the *effort* required to accomplish the task, and the *frustration* that subjects could have experienced in creating 3D VLE using the proposed visual languages.

The questions related to Authoring Adaptive 3D VLE, ISONORM 9241/110-S, Subjective Impression, and Workload Perception categories were closed questions using a Likert scale (Oppenheim, 1992) from 1 (strongly disagree) to 5 (strongly agree).

8.2.5 Data Analysis Principles

In the online questionnaire all questions were mandatory. We used positive as well as negative formulated questions (except for the ISONORM 9241/110-S evaluation that only contained negative formulated questions) and both types of questions used a scale from 1 (Strongly Disagree) to 5 (Strongly Agree). For example:

a) “Using the modelling languages is a good idea” is a positive formulated question.
b) “Using the modelling languages is unpleasant” is a negative formulated question.

A positive response on a positive question is a good result whereas a negative response on a negative question is also a good result. For example:

c) For the positive question “Using the modelling languages is a good idea”, 5 is a good result and 1 is a bad one;
d) For the negative question “Using the modelling languages is unpleasant”, 1 is a good result and 5 is a bad one.

We have interpreted the overall average score values as explained below and summarized in Table 8.1 and Table 8.2.

For positively formulated questions:

- Average values between 3.5 and 5 are considered as a good to very good or perfect evaluation’.
- Average values between 2.5 and 3.5 are considered as a medium/neutral to good evaluation.
- Average values below 2.5 indicate a rather poor evaluation and suggest that improvements are needed.

Average Value	Average of 3.5 to 5	Average of 2.5 to 3.5	Average of 1 to 2.5
Interpretation	Very Good to Perfect (Positive)	Medium / Neutral	Poor / Need for action (Negative)

Table 8.1: VR Usability/Acceptance Positive Scale

The reverse is used for negatively formulated questions: the less the average the more positive the evaluation is.

Average Value	Average of 1 to 2.5	Average of 2.5 to 3.5	Average of 3.5 to 5
Interpretation	Perfect to Good (Positive)	Medium/ Neutral	Poor / Need for action (Negative)

Table 8.2: VR Usability/Acceptance Negative Scale

8.2.6 Evaluation Results

In this section we will present the results of the questions related to each category of the questionnaire.

8.2.6.1 Demographic Information

All participants were from the domain of Computer Science and the average age was 32 (youngest was 26, eldest 36). Although the participants were from the Computer Science domain, the demographic data indicated that very few participants (5 out of 14 participants) were familiar with VR/3D like video games, Virtual Reality, or 3D Virtual Environments. However, all participants were using the computer on daily basis. The majority of the participants were males (11 participants) while only 3 participants were females.

One area of interest is the experience with authoring courses in general. Concerning this aspect, only 2 participants reported to be inexperienced with authoring standard courses, while 12 participants had experience in authoring courses.

Another important demographic variable was the experience in authoring 3D Virtual Environments for courses. The result showed that most of the participants (11 out of 14 participants) had limited experience in authoring 3D virtual environments or videogames in the context of e-learning. On the other hand, all participants were familiar with graphical modelling languages like UML or ORM (to be expected as the participants were computer scientists).

8.2.6.2 Authoring Adaptive 3D VLE and Usability Issues

As mentioned earlier, usability issues related to authoring an adaptive 3D VLE using the proposed visual languages were evaluated mainly in two questionnaires: the authoring adaptive 3D VLE questionnaire (A3DVLE) and the ISONORM 9241/110-S questionnaire. The A3DVLE questionnaire focused in particular on the adaptation aspects in the adaptive storyline language and adaptive topic language. Clearness and understanding of the graphical notations and symbols were also considered in this questionnaire. As mentioned earlier, the ISONORM 9241/110-S questionnaire included questions related to suitability for task, self-descriptiveness, suitability for learning aspects, and conformity with user expectations.

According to the evaluation results (see Figure 8.1), the usability of the visual languages is evaluated as ‘Medium/Neutral’ to ‘Good to Perfect’. The bars diagram in Figure 8.1 presents the results concerning both ISONORM and A3DVLE questionnaires.

There were 8 positive formulated questions related to authoring adaptive 3D VLE (A3DVLE questionnaire) that reported good to perfect while 3 questions (positive formulated) of the same questionnaire received a neutral rating. Concerning the negatively formulated questions in the A3DVLE questionnaire, 4 questions were rated as good to perfect and 2 questions as neutral. 1 question was rated as poor. 5 questions related to ISONORM 9241/110-S questionnaire were rated as good and 3 questions were rated as neutral. We now provide more details.

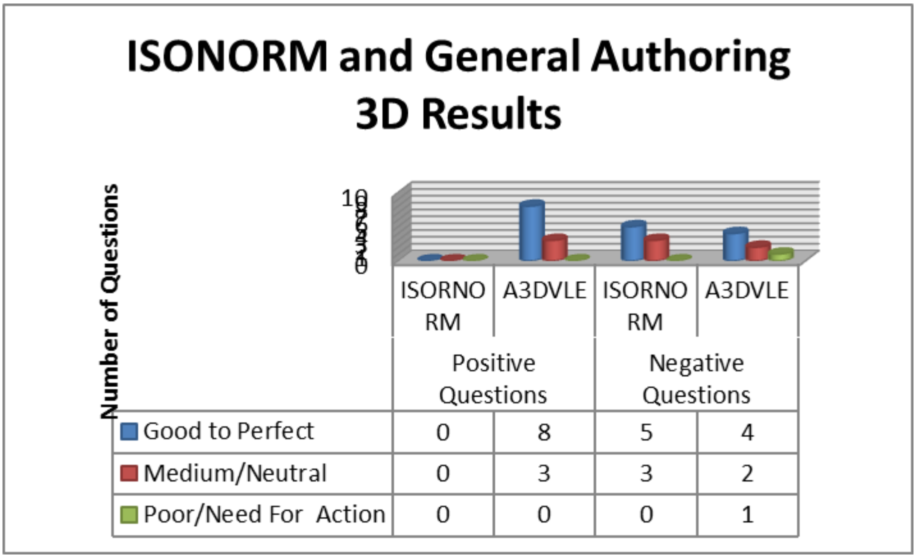


Figure 8.1: ISONORM and Authoring 3D VLE (A3DVLE) Questionnaire Results

In general, all questions related to the *suitability for the task* aspect were rated as good. Among the questions/statements that were related to the suitability for the task aspect and were rated as good, the statement "*The proposed modelling languages allow creating the specified scenarios*" was rated good by 11 participants (out of 14). Furthermore, 12 participants (out of 14) considered the visual languages to appropriately meet the demands of the authoring adaptive 3D VLE. Additionally, 9 participants (out of 14) gave a good feedback to the question "*The proposed modelling languages complicate the authoring task due to a complex process*" and the question "*The modelling languages force the user to follow an unnecessarily rigid sequence of steps*" was rated as good by 11 participants (out of 14).

Turning to conformity with user expectations, 9 participants (out of 14) agreed that the proposed visual languages are designed according to consistent principles.

With regard to the usefulness and easiness of the visual language, the results were good and neutral depending on the individual visual language. For instance, 8 participants (out of 14) indicated that defining the adaptive storyline model using its corresponding visual language was very easy. And being able to define the adaptation inside a topic using the adaptive topic language was considered as "very useful and easy" by 9 participants (out of 14). Furthermore, 10 participants (out of 14) gave a good score concerning the statement "*Being able to add 3D Adaptation Types to 3D learning concepts in a topic is very useful.*". Moreover, 12 participants (out of 14) considered applying adaptation strategies to a topic as useful. However, the answers (8 out of 14 participants) on the question related to the easiness of defining the pedagogical model tip the scales in favour of a neutral feedback. In addition, 10 participants rated "*Using the Adaptive Topic Language to specify the flow of a topic was very easy*." as neutral too.

Concerning the questions related to the *self-descriptiveness* aspect, participants (12 out of 14) indicated that the graphical notations (symbols) used in the three modelling languages were clear. Moreover, 10 participants out of 14 rated the statement: "*The language/terminology used in explanation is understandable*" as good too.

Furthermore, for the *suitability for learning* aspect the results vary depending on the modelling language. In general, 10 participants (out of 14) considered the used graphical notations in the three visual languages as not difficult to learn. Furthermore, 8 participants (out of 14) gave a good result about the required time to learn the visual languages. The statement "*Using the Pedagogical Model Language to specify the pedagogical relationships between learning concepts is difficult to learn.*" was rated with a good by 8 participants (out of 14). In addition, 10 participants considered the adaptive storyline language and adaptive topic language as not difficult to learn.

However, despite the fact that the three modelling languages were rated as good, 7 participants gave a neutral score concerning understanding the goal of the pedagogical model. Moreover, concerning the adaptation strategies, some participants (9 out of 14) gave a neutral rating on the question about the difficulties to understand them. And 8 out of 14 participants gave neutral to the difficulty of understand the adaptation types. Furthermore, 10 participants gave a neutral score to whether the three visual languages require remembering too many issues or not. Although many participants agreed that there was no unnecessary input or effort, some of them (6 participants out of 14) spent quite some time on understanding the adaptation types and strategies and defining the course structure before they could start with designing the adaptive 3D VLE using the three visual languages.

Finally, a poor score (10 out of 14 participants) was obtained for the question "*The defined Pedagogical Relationship Types are difficult to understand*".

It may be useful to mention that, in average, participants spent around 40 minutes to complete the required tasks using the three visual languages (best time was 25 minutes and worst one was 90 minutes).

Overall, from the above results, related to the suitability for the task, conformity with user expectations, usefulness and easiness, self-descriptive and suitability for learning aspects, we can conclude that the proposed visual notations of the modelling language allowed the participants to design the adaptive 3D VLE solar system without being 3D/VR experts. The overall positive feedback on the usability questions gives an indication that the required functionality is supported by the proposed modelling languages. However, we have to take into consideration that the participants had a good knowledge of modelling. In addition, most of the participants considered the proposed modelling languages rather intuitive.

8.2.6.3 Acceptability

As mentioned in section 8.2.4, acceptability was evaluated with the Subjective Impression Questionnaire. The aspects *perceived ease of use, attitude,* and *perceived usefulness* are covered by this questionnaire.

These aspects scored in average good (see Figure 8.2). In particular, 4 questions related to perceived ease of use were rated as good while the other questions (3 questions) were rated as neutral. Concerning the two questions related to the attitude aspect, 1 question was rated as good and the other was rated as neutral. Finally, perceived usefulness questions (2 questions) were rated as good.

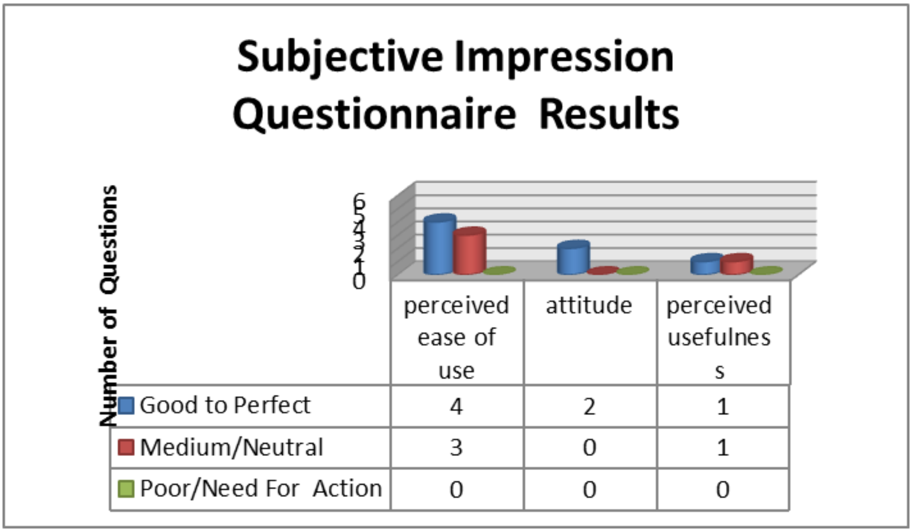


Figure 8.2: Subjective Impression Questionnaire Feedback.

Starting with *perceived ease of use* questions, some participants (9 out of 14) indicated that both the pedagogical model language and the adaptive topic languages were easy to use. Furthermore, 8 participants credited the easiness of remembering the notation and their use in the modelling languages. 10 participants believed that they could become skilful at using the modelling languages. However, some neutral feedback was also given to three questions. Some participants (8 out of 14) rated the "ease to use" of adaptive storyline language as neutral. Furthermore, another neutral feedback was given by 9 participants (out of 14) to the statement "*Using the modelling languages make* *it easy to author adaptive 3D courses.*". Also considering the visual languages as easy to understand was rated with a neutral feedback by 8 participants (out of 14).

There is a notable good evaluation concerning the users' attitude questions. This is based on the fact that 13 (out of 14) participants rated the statement: "*Using the modelling languages is a good idea*" as good to perfect. Moreover, 12 (out of 14) participants gave a good rating about considering the modelling languages as pleasant to use.

Although, *perceived ease of use* and the *attitude* aspects were rated as good, one of the questions related to *perceived usefulness* was rated as neutral and the other one was rated as good. In particular, 7 participants out of 14 gave a neutral rating for "*easy to use the three visual languages to author adaptive 3D VLE*". However, only 3 (out of 14) disagreed with considering the three visual languages as useful in authoring the adaptive 3D VLE. Having two different results for the two questions related to perceived usefulness could be affected by the fact that the participants were using paper and pen to perform their authoring tasks. Consequently, they had to look up the syntax of each language in the provided documents.

Overall, the results concerning the acceptability were good. The participants also reported a good degree of satisfaction on the authoring process of adaptive 3D VLE using the Pedagogical Model Language, the Adaptive Storyline Language, and the Adaptive Topic Language.

8.2.6.4 Qualitative Feedback

In this section we present the results of the qualitative feedback (open questions). The results are divided into three categories: *appreciation feedback*, *depreciation feedback*, and *recommendations*.

Appreciation

In responding the question "*What did you like best about the authoring approach in general and its languages?*', most of the participants (11 out of 14) noted that they liked the fact that there are three modelling languages for creating the whole course of adaptive 3D VLE. Further on, ease of use and consistency was mentioned (7 times). Others (5 participants) considered the availability of the adaptation strategies that could be applied to all 3D objects related to a topic as very useful.

Concerning the question "*Was the Adaptive Storyline Language expressive enough to specify the overall storyline of the adaptive 3D course? (Please describe why 'Yes it was' or 'No it was not')*", answers revealed that connecting topics with adaptation rules helped to define the flow of the overall storyline. In particular, 8 participants liked the adaptation rules between topics and the fact that they could choose which adaptation strategies to apply to the 3D objects inside the topic.

Concerning the question "*Was the Adaptive Topic Language expressive enough to specify the details of each topic? (Please describe why 'Yes it was' or 'No it was not')*", 4 participants highly appreciated that they could specify adaptation types for different 3D virtual objects. Furthermore, being able to define when the adaptation type should be triggered by means of a VR event helped the author to obtain a general overview of when the adaptations will take place. In addition, 7 participants gave a credit to the use of hints and notification messages for students when an adaptation action would happen.

Depreciation

In responding to the question "*What did you like least about the authoring approach in general and its languages?*", answers revealed some flaws which are summarized into the following categories:

- Some (3 participants) would have preferred to use a software tool that enables them to drag and drop the different visual notations for every single modelling language.
- Using different adaptation types to the same object was confusing for 4 participants.
- One complaint about designing the storyline was reported. In particular, the participants would have preferred to have the adaptive storyline without having to add learning concepts for each topic. A suggestion was to define the learning concepts when the author creates the adaptive topic model.
- Sometimes it was confusing for the participants to distinguish between the adaptive storyline and adaptive topic. This was mentioned by 6 participants.

The answers on the question "*Was the Pedagogical Model language expressive enough to specify the pedagogical aspects for the adaptive 3D courses? (Please describe why 'Yes it was' or 'No it was not')*" provided some useful insight into a problem related to authoring the Pedagogical Model. For instance, 6 participants needed quite some time to know when the Pedagogical Relationship Types (PRTs) should be used and to understand their purpose. Others (7 participants) suggested using the same colour for all the PRTs instead of using different colours for different PRTs.

Recommendations

Answers related to the question "*What should be improved and how?*" were mostly related to the need for a supporting tool. The use of the modelling languages within an authoring tool could support the authors with different help mechanisms like tool tips, messages, and tutorials. Furthermore, participants mentioned that it would be easier to detect errors in the models using the visual languages with immediate feedback. Another issue related to a supporting tool is that user model attributes like knowledge, interest, and visited could be displayed while the author is defining the updating rules related to the pedagogical relationship type.

Another recommendation had to do with the Pedagogical Relationship Types, which are recommended to be un-coloured. This remark is already mentioned in previous section.

8.2.6.5 Workload Perception

Participants were also requested to give feedback on the workload perception. The answers concerning questions related to mental demand, the effort required to accomplish the task, and the frustration they could have experienced in using the proposed modelling languages, were neutral. However, 9 participants indicated that the task was not physically demanding and that they were mostly successful in accomplishing the authoring tasks using the visual languages.

In addition, some frustrations were reported. For instance, a frustration reported was related to the user model's attributes that have to be used to define the adaptation rules in the adaptive storyline model and topic model. As already mentioned, some participants would have preferred to have a list of the possible user model attributes.

Another frustration was reported about the consistency between the Pedagogical Model and the Adaptive Storyline Model. Some participants (9 out of 14) were wondering whether it is author's role to make sure that the order of learning concepts in both models (Pedagogical Model and Storyline Model) should be consistent. This is indeed a fair question. However, providing an authoring tool that checks the consistency of both models can provide a solution for this issue. This is already realized in a number of authoring tools in the context of adaptive hypermedia such as AHA! (De Bra, Smits, & Stash, 2006) and GRAPPLE (Hendrix et al., 2008; Cristea et al., 2009) by using algorithms to check language grammar, adaptation rules, cycle detection, etc.

8.3 Qualitative Evaluation (Interviews)

This section starts with presenting the goal of this evaluation (section 8.3.1). Then (section 8.3.2), an explanation about the steps that were performed during this evaluation is given. After that, the design of the interview is discussed (section 8.3.3). Finally, the gathered results are discussed (section 8.3.4).

8.3.1 Goal

As mentioned earlier, to seek for more justification for the results obtain in the quantitative evaluation, as well as to obtain extra critical feedback it was decided to use an additional evaluation method, more particularly a qualitative method. We opted for interviews based on questions from the cognitive dimension rather than repeating the questions used in the previous evaluation (where the questions were mainly based on ISONORM 9241/110-S Evaluation Questionnaire (ISONORM) and Subjective Impression Questionnaire (SIQ)). Interviews would allow us to ask for more explanations if needed. The use of a different set of questions could provide us a different angle of incidence.

8.3.2 Set-up

Participants were informed that the evaluation's results would give evidence of the quality of the provided modelling languages and would be used to reveal both advantages and disadvantages. Furthermore, they were informed that there were no right or wrong answers. Moreover, the interviews were done individually.

The conducted evaluation was divided into four steps:

The first step introduced the participant to the research context, goal of the evaluation, and the proposed visual languages.

After that, in the second step, the participant was asked to investigate the provided examples of a pedagogical model, an adaptive storyline model, and an adaptive topic model, which were designed using our visual languages. The examples are the same as the ones used in the previous evaluation (Figure 2 in section A.2.1.1, Figure 4 in section A.2.2, and Figure 6 in section A.2.3).

Successively, in the third step, the participant was asked to do a small exercise. In particular, the participant was asked to create a small pedagogical model containing three learning concepts using three different PRTs. After that, the participant was asked to create an adaptive storyline composed of three topics and applying three different adaptation strategies. Finally, he or she was asked to create an adaptive topic model composed of three learning concepts connected through adaptation rules. The adaptation rules had to be defined using three different VR events, conditions, and adaptation types.

Finally, in the last step, the participant was asked a number of open-ended questions (see Appendix B) to obtain feedback about their experience and opinion related to the visual languages. Note that the interviewer explained, when it was required, why a question was asked and how it was related to the overall interview and to the previous asked questions. Furthermore, the participant was informed that the conducted interview would be recorded for the sake of analysing the answers later on. Besides audio recording, some notes were also taken during the interview to keep track of important comments and answers given.

8.3.3 Interview Design

The questions for the interview are divided into two major groups; one group of questions aimed to gain more critical feedback from the participants (who were of from different domains) on how they perceived the authoring approach and the use of the visual languages, and one group of questions aimed to identify different advantages and disadvantages.

Among the different available techniques to conduct an interview, we chose the structured interview technique. According to (Lazar et al., 2010), a structured interview is used to obtain more accurate results.

As basis for our questions, we have adopted for the Cognitive Dimension Framework (Blackwell et al., 2001, Green & Petre, 1996). Cognitive theories are considered as the centre of the usability evaluation of different modelling languages (Figl et al., 2009) and also of visual modelling languages. In general, the cognitive dimension framework evaluates the usability of visual languages using fourteen dimensions which are *Visibility, Viscosity, Diffuseness, Hard Mental Operations, Error Proneness, Closeness of Mapping, Role Expressiveness, Hidden Dependencies, Progressive Evaluation, Provisionality, Premature Commitment, Consistency, Secondary Notation* and *Abstraction Management.* More information about these dimensions can be found in (Green & Petre, 1996; Blackwell & Green, 2000). We selected this framework as it is specifically developed for visual languages.

As mentioned earlier, in general, the Cognitive Dimension Framework uses 14 dimensions to evaluate the different cognitive aspects related to visual notations. However, in this evaluation, we mainly wanted to focus on those aspects that were identified in the previous evaluation as possible problem areas or required further investigation. Therefore, we selected the following dimensions: *Consistency*, *Role Expressiveness*, *Closeness of Mapping*, *and Hard Mental Operations*.

The following is a description of each selected cognitive dimension and the justification for selecting it.

- Consistency:

  <u>Description:</u> Consistency is defined as: similar semantics are expressed in similar syntactic form (Bobkowska, 2003). Furthermore, consistency is also related to the fact that user who has learned part of the language can infer the rest (Figl at al., 2009). Here it is important to mention that consistency of a visual language should not be confused with the consistency of the model created using the visual language. The latter is related to the reliability aspect as mentioned in (Paige et al.,, 2000).

  <u>Justification</u>: The consistency dimension is selected because it has been noticed that participants in the previous evaluation phase gave some contradictory scores in this respect. Although good results were reported

concerning the easiness of remembering the visual notations and their use and the good feedback was obtained related to the statement "*The proposed modelling languages (Pedagogical Model Language, Adaptive Storyline Language and Adaptive Topic language) are easy to use.*" (section 8.2.6.2), a neutral feedback was obtained for the statement related to the ease to use aspect of the three visual languages to author adaptive 3D VLE in section 8.2.6.3.

- Role Expressiveness

  Description: This dimension is related to how obvious the role of each component of the notation is in the solution as a whole (Green & Petre, 1996).

  Justification: This dimension is considered as the answers to the question "*Was the Pedagogical Model language expressive enough to specify the pedagogical aspects for the adaptive 3D courses? (Please describe why 'Yes it was' or 'No it was not')*" have reported some difficulties in knowing which PRT to used. Another reported issue was related to the colouring of the PRTs (see section 8.2.6.4).

- Closeness of Mapping

  Description: This dimension is related to how closely the notation corresponds to the problem domain (Bobkowska, 2003).

  Justification: This dimension is selected because of a contradictory result related to adaptive storyline language. A neutral feedback was obtained to the ease to use of the adaptive storyline (see section 8.2.6.3), while a good feedback was obtained for the ease of use related to using the adaptive storyline language to specify the main flow of the course (see section 8.2.6.2). Therefore, there was a need to investigate further whether there are some parts of the adaptive storyline language describe things in a strange way, and if this affected the ease of use.

- Hard Mental Operations

  Description: This dimension refers to the degree to which "*the user needs to resort to fingers or pencilled annotation to keep track of what's happening*" (Green & Petre, 1996).

  Justification: This dimension is investigated because of the results obtained in the previous evaluation concerning mental demand and frustration (which were neutral).

Further investigation was concentrated on the usability aspects *ease to use* and *usefulness*. The justification for considering these aspects is related to the following statements:

- Possibility of using adaptation strategies in the context of adaptive storyline language and the use of the adaptation types in the context of adaptive topic language.

- Possibility of associating learning concepts to each topic in the adaptive storyline language.
- Both ease to use and usefulness of the three visual languages.
- The influence of (not) using an authoring tool to evaluate the visual languages.

Finally, it is important to mention that the different cognitive dimensions are considered for the three visual languages rather than only considering the cognitive dimension for the visual language that was rated with neutral or poor.

The formulated questions were open-ended questions and for the sake of emphasis, all questions were asked even if they were already answered during the interview course.

8.3.4 Evaluation Results

8.3.4.1 Participants

The interview evaluation was conducted by considering a group of four people with heterogeneous background. The participants are identified as follow:

- **PGVM** (**P**articipant with **G**ame **V**isual **M**odelling background): this participant had a background in computer science and a good background in modelling languages, especially in designing visual languages for creating adaptive educational videogames.
- **PVM** (**P**articipant with **V**isual **M**odelling background): this participant also had a background in computer science and a good background in modelling in general.
- **PC** (**P**articipant with **C**hemical background): this participant had a background in chemistry.
- **PEG** (**P**articipant with **E**lectronic En**G**ineering background): this participant had a background in electronic engineering

We selected people with different background in modelling to see if this would have a major influence on their opinion. Also computer scientists, as well as non-computer scientists have been considered to see how this would influence the results.

8.3.4.2 Results

The four participants were able to interpret the diagrams made with the different visual languages; they were able to explain what each diagram expressed. In addition, the participants were able to do the assigned task of creating a small model with each language.

Next, we provide the feedbacks received for each language individually. We only mention the ones most important.

Pedagogical Model Language

All participants understood the goal of the pedagogical model and could explain why educators need to define such a model. Furthermore, they understood the purpose of each PRT and could explain their meaning. They were able to recall mainly four PRTs, which are *prerequisite-for, co-requisite for, interesting-for,* and *updates-knowledge-of.* The rest of the PRTs like *defines, illustrates,* and *Inhibits* could not be recalled. However, according to 3 out of the 4 participants (PGVM, PVM and PC) providing a list of PRTs in an authoring tool would make it easier to explore all available PRTs and understand their purposes.

Considering the Role-Expressiveness, one of the participants (EGVM) was confused because of using different colours for different PRTs. Note that this remark was also raised in the previous evaluation. A suggestion was given to leave the colouring mechanism as an option for the author so that he can either use a unified colour for all PRTs or he can customize each PRT's colour using an authoring tool according to his own needs.

Turning to the Closeness of Mapping, three participants agreed that there are no parts of the visual language that were strange or perceived as unnatural for describing the pedagogical model. The other participant (PGVM) again complained about the use of different colours for different PRTs.

With regard Hard Mental Operations, all participants considered the visual notations as easy to remember. Furthermore, a participant appreciated the fact that there was no need to follow a specific flow in this language, as learning concepts can be inserted in any place (in the drawing canvas) and connected with each other via PRTs.

On the question "*What are positive/negative aspects of PML? How could it be improved?* ", two participants (PGVM and PVM) mentioned the need of using collapse and expand functionality to see/hide all learning concepts that are connected with a learning concept via a specific PRT. According to the two participants, such technique can make a large pedagogical model more readable.

Adaptive Storyline Language

All participants were able to describe the main goal of the adaptive storyline language. For instance, one of the participants (PVM) explain it as it defines the learning sequence from high level point of view by specifying tow aspects: First, it defines the different topics that need to be covered in the learning path. Second, it defines the rules that should be satisfied before starting a new topic. Another participant (PC) explained it as "*defining chapters that need to be covered in a course and then you define the sequence of the chapters and inside each chapter you define what needs to be studied*."

The participants were also able to understand the meaning of the adaptation strategies. The most common recalled adaptation strategies were *MarkObject, FilterObjects, TourGuide, InteractionAtMost,* and *BehaviourAtMost.* However, two participants (PGVM and PVM) mentioned that with the support of an authoring tool, it would be easy to review all the available adaptation strategies and to know the purpose of each adaptation strategy.

In responding the question "*What do you think about the need to add learning concepts to a topic using the Adaptive Storyline Language?"* (asked because there was some questioning about this in the previous evaluation), the four participants agreed that it is useful and important to add the learning concepts to the related topics in the stage of defining the learning path.

Answers related to the question "*Do you understand how the mechanism of the storyline adaptation rules is working?*" were positive. However, two participants (PGVM and PVM) proposed to represent the condition using a numerical value instead of the C letter. The numerical value should be used to represent the learner's knowledge level on which the condition is formulated.

Turning to the Role-Expressiveness, the four participants considered the different parts of the visual language do contribute to the goal of the language. One of the participants (PEG) would have preferred a training session before starting to use the adaptive storyline language.

Considering Closeness of Mapping and Hard Mental Operation, the four participants agreed that there are no strange parts in the adaptive storyline model and the visual notations are easy to remember. They also appreciated the fact that the adaptive storyline language enabled them to define the start and the end of the learning path, and to divide the learning path into different topics connected with adaptation rules. Important to mention is that one of the participants (PC) reported that in the beginning he was confused about using circles and rectangles. However, after realizing that circles are used to represent topics and rectangles are used to represent learning concepts then it was clear for him.

With regard to the ease-to-use aspect, the four participants were positive. However, one participate (PGVM) mentioned that there is a need for a tool to check the correctness of the structure (syntax) of the designed adaptive storyline model.

In respond to the question "*What are positive/negative aspects of ASLL? How could it be improved?* ", a remark was raised by two participants (PGVM and PVM) who mentioned that there is a need for a consistency mechanism between the pedagogical model and adaptive storyline model. Such a mechanism could be supported by an authoring tool and could enable authors to base the adaptive storyline model on a particular part of the pedagogical model.

Finally, more feedback was asked by the question "*Is there anything else you would like to discuss about ASLL?*". A participant (PEG) suggested applying a time limit for each topic so that the storyline could be considered as time-based storyline, to indicate that the learner needs to learn a topic within a determined time to avoid him spending too much time in learning a specific topic. As mentioned earlier, two participants (PGVM and PVM) mentioned the need for an authoring tool that would enable the educators to select a topic and then show the parts of the pedagogical model that corresponds to the selected topic. According to the two participants, this could increase the consistency between the two models. Another suggestion (proposed by PGVM) was about distinguishing between the different adaptation strategies. This could be achieved by using different visual notations for different categories of adaptation strategies. In more details, the participant suggested to have a specific pentagon shape with dotted end for all the adaptation strategies that are applied to a group of 3D objects and another pentagon shape with dotted edges for adaptation strategies that are related to navigation like *TourGuide* and *FreeNavigationWithSuggestion*, or using specific icons (centred in the pentagon shape) for representing each category of the adaptation strategies. With this mechanism, the category and the purpose of each adaptation strategy could be clearer for authors.

Adaptive Topic Language

The four participants could clearly explain the goal of this language. Furthermore, the participants understood the meaning of the adaptation types and the mechanism of the adaptation rule. Two participants (PVM and PEG) described an adaptation rule as a "*trigger-question-adaptation*" mechanism. Moreover, in response to the question "*What do you think about the possibilities of realising adaptations for a topic by using adaptation types and topic adaptation rules? Would you do it differently?*", the four participants appreciated the adaptation rule mechanism and they mentioned that they would not do it in different way.

Turning to the Role-Expressiveness, three participants (PGVM, PVM and PEG) were able to explain the provided adaptive topic model example. The rule structure including "*VRevent - Condition - Adaptation Type*" was clear for the three participants. However, the other participant (PC) found it difficult to interpret the adaptation rule triggered via a timer VR event.

In relation to Closeness of Mapping, two participants (PGVM and PVM) raised some issues related to the way that the condition is represented. One of the participants (PVM) said the condition should be displayed using a numerical value instead of the C letter inside the condition symbol. This is a similar remark than the one raised in the adaptive storyline language. Another participant (PGVM) suggested displaying adaptation type on top of the arrow and the condition at the bottom of the arrow (without using special symbol for the condition). And to avoid having a cluttered model, a collapse/expand mechanism could be used to show/hide adaptation type and condition at the same time.

For Hard Mental Operation, two participants (PGVM and PEG) reported that it is not easy to remember the sequence of the visual notation without having a training session or to have a supporting authoring tool. A participant (PGVM) suggested using more intuitive icons for the events, such as a clock icon in the timer VR event and a hand icon in the touch VR event. This could make it easier to recognize the trigger type that fires the adaptation rule. Finally, other participants (PVM and PC) mentioned that there is a similarity between the adaptive storyline language and the adaptive topic language as there are conditions and actions to be specified, which could be exploited to make it easier to learn the adaptive topic language when the user is already familiar with the adaptive storyline language.

Concerning the answers of the question "*Do you find the visual language easy to use? Why (not)? Are there some parts that you put in just because there were in the example? If so, which ones?*", a participant (PVM) mentioned the need for a guiding mechanism for the process of designing such a model.

In responding to the question "*Is the ATL small enough, i.e. are there any language features that do not contribute to the purpose of the language?*", three participants (PGVM, PC, and PEG) considered the language as not small because textual information needs to attached to the condition symbol. However, two of the participants (PVM and PC) reported that the language described in a natural way what they want to define when the rule should be triggered, i.e. what is the condition to be evaluated and what is the visual effect that should happen to the 3D object.

Almost all the answers related to the questions *"What are positive/negative aspects of the Adaptive Topic Language? How could it be improved?"* and "*Is there anything else you would like to discuss about ATL?"* were covered in the previous questions. For instance, answers were related to the suggestion to represent the condition differently. Also to remove the C letter from the condition symbol and replace it with numerical value.

General Aspects

It is interesting to note that the four participants answered the question related to the Consistency in a positive way. In more detail, the participants were able to recognize that a circle is used to refer to the topic (in the adaptive storyline language) whereas a rectangle (in both pedagogical model language and adaptive topic language) is used to refer to the learning concepts. Furthermore, the condition symbol also has the same visual notation in both the adaptive storyline language and adaptive topic language. The only remark raised (by PGVM) related to consistency was the use of a consistent font format for the different visual notations, and consistent colour for learning concepts symbol.

In responding to the question "*Is it possible to design a large course using the three visual languages? Please explain why yes or no?*", all participants agrees that it would probably be possible to use the three visual languages to design an adaptive 3D VLE. For instance, a participant (PC) said that it would be possible to design a linear or non-linear storyline using the adaptive storyline language. However, the two main points mentioned are (1) the consistency mechanism between the pedagogical model and the adaptive storyline model, and (2) reducing the textual information related to the condition symbol in the adaptive topic model.

Turning to both usefulness and ease-to-use aspects, the four participants found that the visual languages are useful and they considered the three visual languages as easy to use languages. They mentioned different reasons. For instance, a participant (PVM) said that it was easy to use because of being able to define the adaptation rules naturally in the adaptive topic model, to connect the learning concepts with pedagogical relationships using arrows symbols, to define the adaptation rules between different topics using condition and action mechanism rule.

For Hard Mental Operations, most of the answers were also related to previous remarks and feedback. For instance, supporting the authors with a list of PRTs in an authoring tool could help them to create a pedagogical model without knowing all possible PRTs by heart (mentioned by the four participants). Furthermore, supporting authors with a wizard that enables them to specify the different parts of the adaptation rule can be helpful and could reduce the mental demand (mentioned by PGVM and PVM).

At the end, in response to the question "*Do you think that the availability of an authoring tool would have influences your answers and opinions?* " all participants mentioned that the answers would be the same as they evaluated the visual languages rather than the functionality and usability of an authoring tool that would support the use of the visual languages. However, the participants (all of them) were looking forwards to try out the visual languages using an authoring tool. Furthermore, they started to draw some requirements for such an authoring tool.

8.4 Discussions

Overall, the evaluations of the proposed visual languages were quite positive. However, we recall that in the first evaluation all participants were computer scientists, and two of the participants of the second evaluation were computer scientists too, which could have had an impact on the evaluation. Furthermore, we need to be careful with generalising the results, as the number of users in the evaluations (14 in first evaluation and 4 in the second evaluation) was rather small.

In more detail, we can state that in both evaluations usability, acceptance, and cognitive aspects were rated well. In the quantitative evaluation (first evaluation), the usability results were good despite the fact that most of the participants lacked experience with authoring adaptive 3D VLEs and no true learning period was foreseen to let the users practice. Furthermore, acceptance results were good too, despite the fact that it is obvious that some time is required to get acquainted with the visual notations. In the qualitative evaluation (second evaluation), the results of the different cognitive dimensions that were evaluated were positive and there was no notable difference between people with different background.

The first evaluation raised some issues for which we have sought explanations in the second evaluation. The following explanations were found

A. *Ease to use and easy to remember*
As mentioned earlier, some contradictory results concerning the ease of use were reported in the first evaluation. According to the second evaluation, this contradiction could be attributed to the fact that there is a need for training session or authoring tool to support the participants in using the adaptive topic language in particular.

B. *Visual notations*
In the first evaluation, a neutral feedback was given to the mental demand and frustration. Although, the cognitive aspect was rated good in the second evaluation, it was noticed that some of the visual notations (both the adaptation strategies in the adaptive storyline language and VR events in the adaptive topic language) need to be attached or complemented with icons or symbols whose appearance indicates their meaning more clearly and intuitively.

C. *Pedagogical Relationship Types*
Different PRTs are distinguished by means of their name but also by a colour. However, this led to difficulties in understanding the pedagogical model. A number of suggestions have been given in this respect: considering the use of collapse/expand mechanism in the authoring tool and leaving the colouring mechanism up to the authors.

D. *Storyline*
As far as the authoring of an adaptive storyline is concerned, the aim of decomposing the overall storyline into sub-topics was to reduce the complexity of the adaptive storyline model and enhance its ease of comprehension. In fact, this idea was appreciated by most of the participants. Furthermore, it was possible to specify a storyline using the approach currently used because it allows authors to specify adaptations in a specific order.

E. *Consistency between Pedagogical Model and Adaptive Storyline Model*

Turning to the consistency between the models, it was reported by the participants that the authoring of the both the Pedagogical Model and the Adaptive Storyline Model should be simplified and the relationship between both should be exploited automatically. For instance, in principle, a 3D learning object should not be "displayed" (in a course) until all defined pedagogical relationship types of the corresponding learning concept are satisfied. In fact, this can be realized by using advanced algorithms or by using an authoring tool that enforces the consistency between the two models.

F. *Assigning learning concepts in the Adaptive Storyline*
Although there were some remarks related to the usefulness of adding the learning concepts to the topic in the adaptive storyline language, in the first evaluation, this was not confirmed in the second evaluation. However, an authoring tool could be useful to show explicitly and visually the link between the topic and the corresponding pedagogical part.

G. *Assigning adaptation strategies and types*
In the first evaluation, a neutral feedback was given to questions related to the use of adaptation strategies and types. Furthermore, the use of multiple adaptation types to the same 3D object was perceived as confusing. However, these findings were not confirmed in the second evaluation. The results of the second evaluation showed that the use of adaptation strategies and types is useful. A suggestion to solve possible confusions related to adaptation types and strategies was given in *remark B* by discriminating the adaptation strategies depending on their category and to use some more intuitive icons for VR event.

H. *Hint Mechanism*
The possibility to indicate that notification messages should be provided to the learner, using the associated visual notation (in the adaptive topic model) proved fundamental in helping authors to direct learners in their learning process. One of the good results was attributed to the use of the notification mechanism to complement the adaptation action with textual, audio, and/or video messages. This aspect has been confirmed in the second evaluations.

I. *The need for an authoring tool*
The two evaluation results stressed the need for a supporting software tool. This would not only make it easier to use the languages but also could avoid obvious mistakes and allow checking for the models' correctness and consistency (e.g., between the Pedagogical Model and Storyline Model). The fact that the participants had to do the authoring

exercise using paper and pencil could have influenced the results. However, the participants in the second evaluation indicated that the results could not have been different if they were able to design the models using a software tool.

In general, there the results from the second evaluation (qualitative evaluation) phase are consistent with the results of the first evaluation (quantitative evaluation). This is a first indication that the results could be applicable to a broad range of users (Lazar et al., 2010, p.190). The following highlights that the results of both evaluations are consistent and as a result the validity level of the evaluation is good.

- Participants in the first evaluation perceived the visual languages as clear and useful, understandable, easy to learn, easy to remember and pleasant and ease to use. This is also confirmed in the second evaluation by good appreciation for the different dimensions like consistency, role-expressiveness, and closeness of mapping.
- The qualitative feedback (in the first evaluation) provided useful information that can be considered for further improvements. For instance, it would be beneficial to provide users with a list of user model attributes and 3D VLE activity that are required for (re)defining the updating rules in the pedagogical model and defining the adaptation rules in the adaptive topic model respectively. This was also emphasised in the second evaluation.
- The two evaluation results appreciated the adaptation rule mechanisms used.

The effectiveness of our authoring approach can be considered as good, as all participants in the quantitative evaluation were able to define the adaptive 3D VLE in the right way and all participants in the qualitative evaluation were also able to perform the small authoring exercise. It is worth noting that they were able to do this without a training period.

Finally, we should acknowledge that the evaluations presented in this chapter are only for the proposed visual languages. In order to fully validate our approach, additional evaluations should be conducted, preferably when a functional prototype of an authoring tool is available.. Moreover, further evaluations should also consider participants with different background, like experts in VR for validating the advanced features and non-technical users, with and without modelling languages experience, to evaluate the standard functionality. It may also be important to measure the time required for completing the tasks by different categories of users. If the time required to author a course is too long, people may not be prepare to use it in practise.

8.5 Summary

This chapter has described the evaluation of the modelling languages proposed in the thesis to design adaptive 3D VLEs. Two evaluations were performed: one that was mainly quantitative and one qualitative evaluation. Both evaluations are described.

In the first evaluation, 14 participants were involved. They were given a short description of the modelling languages by means of examples. Furthermore, a list of pedagogical relationship types, adaptation strategies, and adaptation types were provided and explained to help the participants to understand the purpose of each concept. After that, the participants were asked to model an adaptive 3D solar system course using the proposed visual languages. This exercise was done with pen and paper. Afterwards, the participants were asked to fill in an online questionnaire. The questionnaire was divided into the subcategories Demographic Information, Adaptive 3D VLE, ISONORM 9241/110-S Evaluation Questionnaire, Subjective Impression Questionnaire, Qualitative Feedback, and Workload Perception. All the results are presented in the second section of this chapter.

In the second evaluation, four participants with different background were involved. The participants were given a short presentation about the visual languages and their purposes. After that they were asked to explain some examples that were drawn using the visual languages. Then, they were asked to perform a small authoring exercise. Finally, a list of open-ended questions was asked. The questions were mainly composed to gather more feedback on issues that are gathered from the first evaluation. They were based on the Cognitive Dimension Framework. All the results are presented in the third section of this chapter.

Analysis of and a discussion about the results of the evaluations and their validity are also presented. Furthermore, a number of directions to improve the modelling languages, derived from the evaluations, are also presented.

The evaluations indicate that the Pedagogical Model Language, the Adaptive Storyline Language, and the Adaptive Topic Language are intuitive and can be used by people without deep knowledge of 3D/VR. The participants were able to perform the authoring process within a fair period of time using the proposed modelling languages. Using the three modelling languages, the participants were able to capture the different aspects required to design an adaptive course with 3D contents. Moreover, participants found the visual notations easy to use. Not surprisingly, the evaluation revealed the need for software support, i.e. an authoring tool.

Chapter 9: The Authoring Tool

The purpose of this chapter is to illustrate how our authoring approach can be supported by an authoring tool. It presents the functional requirements which should be supported by such an authoring tool in order to facilitate authoring adaptivity in 3D VLE. The chapter also presents the different modules in the authoring tool architecture. Using the proposed modules, the author will be able to design adaptive 3D VLE by following a specific authoring process. This is supported with a user-friendly graphical user interface. It should be noted that not all parts of this authoring tool has been implemented. A complete authoring tool has been implemented in the context of the GRAPPLE project, but the authoring tool presented here is an improved version of this earlier tool. The parts that has been changed or added have not been implemented because of time limitations.

For the sake of simplicity, in this chapter we will call the adaptation types and strategies *3D adaptation states*. 3D adaptation states are the instances of both adaptation types and strategies. Furthermore, 3D adaptation states can be customized during the authoring process.

This chapter is structured as follow: section 9.1 presents the functional requirements that must be supported by the authoring tool. In section 9.2, the authoring tool architecture is presented. Furthermore, a description of the different editors composing the authoring tool is given. After that, the authoring process flow and the information exchanging mechanism are discussed in section 9.3 and section 9.4 respectively. The graphical user interface of the authoring tool is discussed in section 9.5. Finally, the chapter is summarized in section 9.6.

9.1 Authoring Tool Requirements

As already discussed in the chapter 6, we cannot expect that the authors of 3D courses (in principle teachers) have deep knowledge about 3D modelling. Therefore, the authoring tool should be as simple as possible from a 3D knowledge level point of view. However, it should not limit the possibilities for developing advanced adaptive 3D VLEs. As a result, we opt for a tool that also support advanced authors to define advanced adaptation inside the 3D VLE.

However, the more complex adaptations require more knowledge about VR/3D modelling. If the author is satisfied with the use of predefined adaptation types and strategies, he will only need a minimum knowledge about 3D modelling.

Based on the authoring approach and the scenarios given in section 6.1.1, we derived a list of general functional requirements for the adaptive 3D VLE authoring tool:

R1: The authoring tool should allow authors to define how the user model should be updated (i.e., defining the pedagogical model). Therefore, the author should be able to do the following:

R1a: to define (customize) new (predefined) Pedagogical Relationship Type (PRT). This functionality is mainly oriented towards advanced authors who are familiar with designing adaptive courses.

R1b: to formulate Pedagogical Update Rules (PUR) associated with a PRT. Also this functionality is mainly oriented towards advanced authors who are familiar with designing adaptive courses.

R1c: to design the pedagogical model related to the created course. This should be suitable for both types of authors (novice and advanced).

R2: The authoring tool should allow authors to define an Adaptive Storyline Model (ASM). To achieve this, the authors (novice as well as advanced) should be able:

R2a: to define the different topics to be covered in the storyline.

R2b: to identify appropriate 3D objects to be used for the learning concepts involved in the learning process, called 3D learning objects.

R2c: to define the adaptation rules between the different topics in the storyline.

R2d: to associate identified 3D learning objects to their corresponding topics.

R3: The authoring tool should allow authors to create the Adaptive Topic Models (ATM) for every defined topic in the storyline. To achieve this, the author (novice as well as advanced) should be able:

R3a: to customize the different adaptation types to be used in the context of the particular topic. Furthermore, the author should be able to preview the customized adaptation types and strategies.

R3b: to assign adaptation types and VR events to the identified 3D learning objects.

R3c: to select the events, which trigger the adaptation rules, from a predefined list of VR events.

R3d: to define the conditions of the adaptation rules in term of actions performed inside the 3D VLE and in terms of user model attributes.

R3e: to define the action part of the adaptation rules by selecting the customized adaptation types and adaptation strategies assigned to the 3D learning objects.

R4: The author (novice as well as advanced) needs to be able to preview the 3D VLE during authoring. Moreover, he should be able to see the actual effect of applied adaptation types and strategies.

R5: The author should be able to use (novice user) or create (advanced user) an adaptation theme, which can be used in the context of adaptive topic model. To achieve this, the author should be able:

R5a: to create/modify adaptation themes for group of learners.
R5b: to add/delete adaptation types to/from the created adaptation types
R5c: to customize adaptation types properties such as text size, box colour, 3D object's behaviour speed, etc.

R6: The author (novice as well as advanced) should be able to know in which stage of the authoring process he is.

9.2 Authoring Tool Architecture

In this section, we present the architecture of the authoring tool. The architecture is mainly based on the GRAPPLE architecture (De Bra et al., 2010). The two main modules in the authoring tool are the *Data Model Environment* and the *Pedagogical & Storyline Environment,* which are shown in Figure 9.1. Below is a description of the two authoring environments.

Before we start elaborating on the different modules that compose the authoring tool, it is important to mention that both the user model and the domain model specifications are not considered in the proposed 3D VLE authoring tool. Authoring these two models is not specific for 3D VLE, so we suppose that other authoring tools can be used to create these models (e.g., the GRAPPLE authoring tool). Therefore, we consider "external" modules that can be used by some existing services. For this purpose, the *Web Services* module is introduced. This module can also be used to access local and/or external 3D repositories, which are used to store 3D materials and resources.

Besides the two main environments, a visualization tool (preview tool) is integrated in the overall authoring tool to allow authors to preview the 3D VLE and to see the actual effects of customized adaptation types and strategies. The motivation is the fact that if a preview tool is not provided then the author needs to specify adaptations blindly, which can be a difficult task. For instance, specifying a navigation path inside a 3D VLE can be much easier if the author can do it while he is viewing the whole 3D VLE. This previewer is used to satisfy requirement *R4*.

As mentioned earlier, the authoring tool architecture is composed of two authoring environments. On the one hand, the *Data Model Environment* is proposed to be able to define models like pedagogical model, adaptive storyline model and adaptive topic model. In other words, it will allow taking care of some preparation steps such as creating pedagogical relationship types, creating adaptation theme, identifying 3D learning materials, customizing adaptation types and strategies. The *Data Model Environment* consists of different editors to prepare resources. This environment supports requirements *R1a, R1b, R2b, R3a* and *R5*. On the other hand, the *Pedagogical & Storyline Environment* facilitates the definition of the actual adaptive 3D course using the proposed visual languages. In other words, this environment provides the author editors that can be used to define the pedagogical model (satisfy requirement *R1c*), design adaptive storyline model (satisfy requirement *R2a, R2c, R2d*) and create an adaptive topic model (satisfy requirement *R3c, R3d, R3e*).

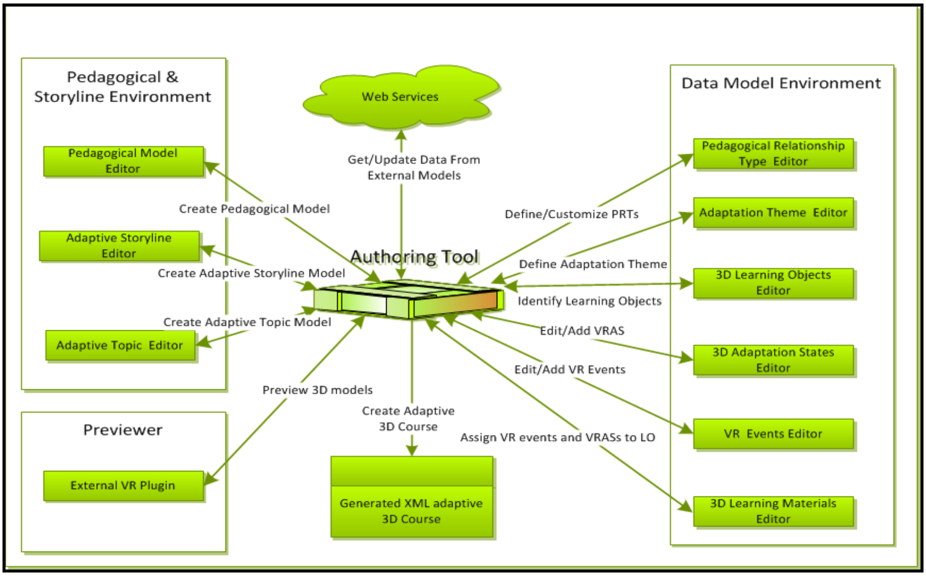


Figure 9.1: Overview of the VR authoring tool Architecture

Next, we will elaborate on the different editors from the two environments.

Firstly, the *Data Model Environment* is composed of six editors, which are mainly used by advanced authors to create and prepare the data models that are used in the other environment (*Pedagogical & Storyline Environment*). These preparation steps are only needed if the provided defaults are not satisfying.

In general, by using separated editors, it is more obvious for the author to know in which stage of the authoring process he is (requirement *R6)*. The editors are*:*

- *Pedagogical Relationship Type Editor* is used to define and customize the PRTs and the PURs. The resulted data model is used in The Pedagogical Model editor (see further on). This editor satisfies requirements *R1a* and *R1b*.
- *Adaptation Theme Editor* is used to create a specific adaptation theme related to a group of learners (i.e., beginners, intermediate or advanced). The resulted data model is used in the Adaptive Topic Model (see further on). This editor satisfies requirements *R5a, R5b, R5c*.
- *3D Learning Object Editor* is needed to identify the 3D resources that will be used/adapted in the 3D VLE. The resulted data model is used in the Adaptive Storyline Model Editor. This editor satisfies requirement *R2b.*
- *3D Adaptation State Editor* is used to create or customize adaptation types and adaptation strategies (called 3D adaptation states) that will be applied to the identified 3D learning objects. The resulted data model is used in the Adaptive Storyline Model Editor and the Adaptive Topic Model Editor. This editor is used to satisfy requirements *R3a.*
- *VR Event Editor* is used to define or customize VR events to be used in the adaptation rules. The resulted data model is used in the Adaptive Topic Model Editor (see further on). This editor satisfies requirement *R3c*.
- *3D Learning Material Editor* is used to associate 3D learning objects with adaptation types, adaptation strategies and VR events. The resulted data model is used in the Adaptive Topic Model Editor (see further on). This editor satisfies requirement *R3b.*

To sum up, the data models (output) defined for the previous editors are used as input data for editors in the *Pedagogical & Storyline Environment*.

Secondly, the *Pedagogical & Storyline Environment* is composed of three editors. As for the previous editors, by using separated editors, it is more obvious for the author to know in which stage of the authoring process he is (requirement *R5)*. These are the editors from the Pedagogical & Storyline Environment:

- *Pedagogical Model Editor* is used to connect the different learning concepts with PRTs and to customize associated PUR (if required). The editor supports the Pedagogical Model Language to perform the previous tasks. This editor satisfies requirement *R1c*.

- *Adaptive Storyline Editor* is used to define the different topics and the adaptation rules between the different topics using the Adaptive Storyline Model Language. Furthermore, the editor allows the author to map topics with their corresponding learning concepts. This editor satisfies requirements *R2a, R2c, R2d*.
- *Adaptive Topic Editor* is used to define the adaptation rules between the different learning concepts. Furthermore, it allows the author to apply adaptation types to learning concepts. This editor satisfies requirements *R3a, R3b, R3c, R3d, R3e*.

When all models are created, the authoring tool can generate the final XML file containing all information for the 3D course and which will be the input for the adaptive engine.

The next section will describe the process flow of the authoring tool and how the different components contribute to the authoring process of an adaptive 3D VLE.

9.3 Process Flow of the Authoring Tool

After discussing the different functional requirements and the authoring tool architecture, we will describe the authoring process flow used to create an adaptive 3D VLE. We will describe the authoring process using the 3D solar system example. The following processes are involved in authoring an adaptive 3D VLE:

Preparatory Processes:
Before one can start to design an adaptive 3D VLE some resources are required. These can be created by the editors provided in the *Data Model Environment* (or defaults provided can be used). Figure 9.2 illustrates the required steps to prepare the different resources. Note that all six preparatory steps are optional. They are only needed for advanced authoring, when defaults are not satisfying. Furthermore, an advanced author can start with any of the five processes without the need to follow a specific order.

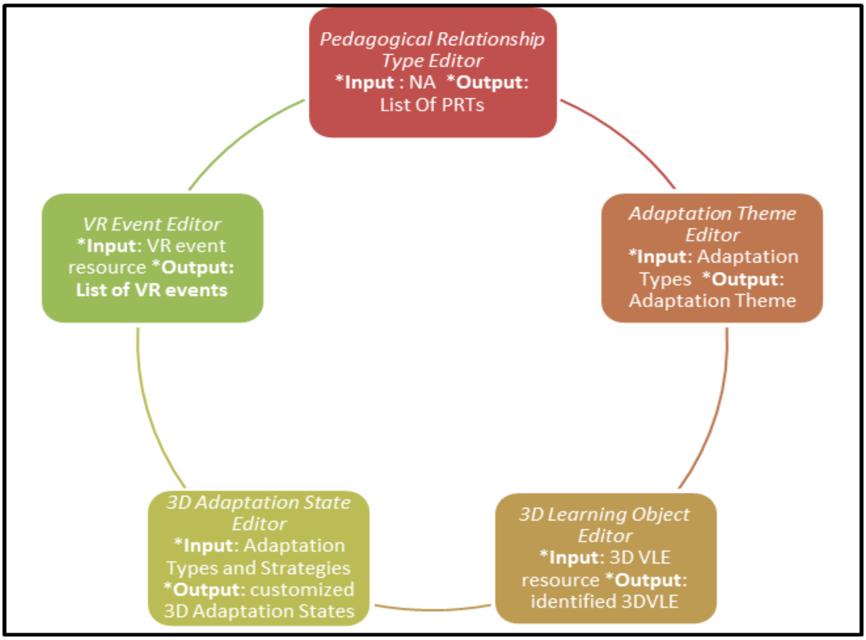


Figure 9.2: Overview of the preparation steps needed to specify an adaptive 3D VLE.

Firstly, an author may start by specifying which 3D objects will be considered in the learning process (using the 3D Learning Object Editor). This step is essential as a 3D VLE may include a large number of 3D objects but not all of them are used for the learning or need adaptation. Therefore, it is necessary to identify the 3D objects in the 3D VLE that are required for the learning process.

For example, the author may have a 3D resource that includes all the planets, moons, stars, galaxies, comets, dwarf, and satellite When the course only concerns stars, planets and their moons, he can identify the actual 3D learning objects such as sun, earth, mercury, etc. in the available 3D resource.

The input of this process is a 3D resource that represents the overall 3D VLE. The output of this process is a *3D Learning Object Data Model* (*3DLODM*) which includes all the identified 3D learning objects. The result of this step is used in 3D Learning Material Editor (see further on). Furthermore, a 3D VLE with identified 3DLO's is stored in a repository, which can be consulted for reusing the resource for other courses.

Secondly, it may be necessary to customize a collection of adaptation types to be used for a specific learner group. This is achieved by using the Adaptation Theme Editor, which enables the author to create a theme. Every theme is specific to a certain group of learners and contains a number of adaptation types which are customized for the group.

For instance, the author may want to create an adaptation theme for beginner learners where the *spotlight* adaptation type has the yellow colour for its ray, and the *displayAnnotation* adaptation type has a red colour for the text.

This process requires adaptation type resources as input. The output is a number of adaptation themes. An *Adaptation Theme Data Model* (*ATDM*) is used in the context of the adaptive topic model editor.

Thirdly, customizing adaptation types and strategies may also be required. This is done by using the 3D Adaptation State Editor. The editor enables authors to specify the different properties of adaptation types and strategies (3D adaptation states) that will be used.

For instance, the author may want to change the colour or the bounding box of the *highlight* adaptation type.

Adaptation types and strategies specifications are the input for this editor. The output is a so-called *3D Adaptation States Data Model* (*3DASDM*). Because 3D adaptation states can be used in different course contexts, the author can store the 3D adaptation states in a specific repository and use them in different courses.

Fourthly, defining or customizing the VR events that will be used in the context of the adaptive topic model may also need to be prepared. In fact, this process requires a good background about 3D technologies, but as mentioned before, this step is option.

For instance, the author may change the time interval related to the earth rotation around the sun using the *timer* VR event or the proximity area related to mercury using in *close to* VR event.

The input used in this process is VR event specifications (as X3D format) and the output of this process is a *VR Event Data Model* (*VREDM*). The output of this process is used in 3D Learning Material Editor.

Fifthly, defining PRTs, which are used in the pedagogical model, can be prepared in this step. To perform this, the Pedagogical Relationship Type Editor is used.

There is no input data for this step; the output is the *PRT Data Model* (*PRTDM*), which is input for the Pedagogical Model Editor.

Sixthly, after defining the 3D learning objects, 3D adaptation states, and VR events, the author will be able to apply 3D adaptation states and VR events to annotated 3D learning objects. He will also be able to perform further modifications to 3D adaptations states or VR events. To achieve this, the 3D Learning Materials Editor is used. See Figure 9.3.

For instance, the author may assign a customized *highlight* (3D adaptation state) and the *close to* (VR event) to Earth (3D learning object).

The input is 3D Learning Object Model Data (3DLODM), 3D Adaptation States Model Data (3DASDM), and VR Events Model Data (VREDM). The result of this process is the *3D Learning Material Data Model* (*3DLMDM*), which is input for the Adaptive Topic Model Editor.

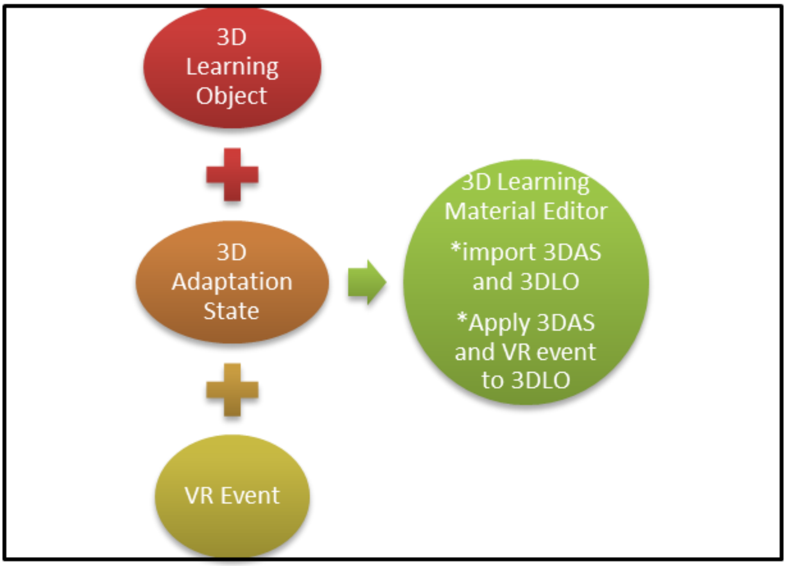


Figure 9.3: Creating 3D Learning Material using 3DLOs and applying both 3DASs and VR events.

Kernel Processes:
After the required resources have been prepared (possibly by an advanced author), the author is able to start defining the three major models, i.e. Pedagogical Model, Adaptive Storyline Model, and Adaptive Topic Model.:

First, the author should start with specifying the pedagogical aspects of the adaptive 3D VLE. He can use the Pedagogical Model Editor for this, which takes the defined Pedagogical Relationship Types (PRTs) as input. He uses this editor to connect learning concepts with PRTs. In this step, for instance, the author may define a prerequisite between the sun and its planets. This is achieved by using the pedagogical model language. Further customizations to PURs (if they are needed) can also be done in this step.

Secondly, the author is able to define the Adaptive Storyline Model. Using the Adaptive Storyline Model Editor, the author first creates topics to be covered during the learning process. After that, he can assign the learning concepts to the different topics. Finally, he defines the storyline adaptation rules between the different topics to define an adaptive learning path for the learner. To define a storyline adaptation rule, the author needs to define the condition on which the adaptation strategy will be applied. The author can use an adaptation strategy from one of the predefined adaptation themes.

Finally, the author defines the adaptation flow inside each topic. In other words, the author needs to connect the different learning concepts in each topic with adaptation rules, which apply predefined adaptation types. Therefore, the 3D Learning Material is the input data model for this step.

To define the adaptation rules, the author needs to the select the VR events that will trigger the rule, define the condition that will be evaluated before applying the adaptation, and select an adaptation type that will be applied to the target learning concept. This finishes the authoring of the Adaptive Topic Model.

As mentioned earlier, after defining the kernel models (i.e. Pedagogical Model, Adaptive Storyline model, and Adaptive Topic Models), the author will be able to generate the XML specifications for the adaptive 3D VLE using the XML generator functionality of the authoring tool. Such an XML file can be interpreted by an adaptive engine to deliver the adaptive 3D VLE to the learners.

9.4 Information Exchange

An Information exchange mechanism is used between the different editors, as they are independent modules. Moreover, for the reason of interoperability, an XML-based format is used for the different data models. The editors are using the Web Service Module for retrieving and storing the files from or to the database on the server the side (see Figure 9.4). This approach is taken from the GRAPPLE project.

All XML model files have the same structure. The following XML code outlines this common template. Each model has a header part, where Meta-data of the model is defined. Meta-data includes a unique identifier of the model, model type (e.g., domain model, user model, pedagogical model, etc.), Author Unique Identifier, authorization for reading and writing, creation and update time, title and description. The model specifications will be included between the tags <body> and </body>.

```
<model>
<header>
       //the header is shared between all models
<modeluid>660e0-76y-41d-a716-44</modeluid>
       //the modeluid is a unique identifier for the model.
<modeltype>DM</modeltype>
       //Types can be DM, PRT, 3D Learning Objects, 3D
learning Material.
```

```
<authoruid>887-e2b-a76-41d-446</authoruid>
       // the author UUID is the unique identifier of the
       author, who created the model.
<authorisation>readwrite</authorisation>
       // authorisation for all other author; the original
       author always has all permissions.
<creationtime>2002-10-10T17:00:00Z</creationtime>
       //the timestamp of creation of the model.
<updatetime>2002-11-10T17:00:00Z</updatetime>
       //the timestamp the model was last updated.
<title>sun-example-dm</title>
       //the title of the model
<description>sun-example-dm</description>
       //the description of the model
</header>
<body>
       //the body contains the actual content of the model.
       For 3D VLE it can be Adaptation States, PRT and ASM.
</body>
</model>
```

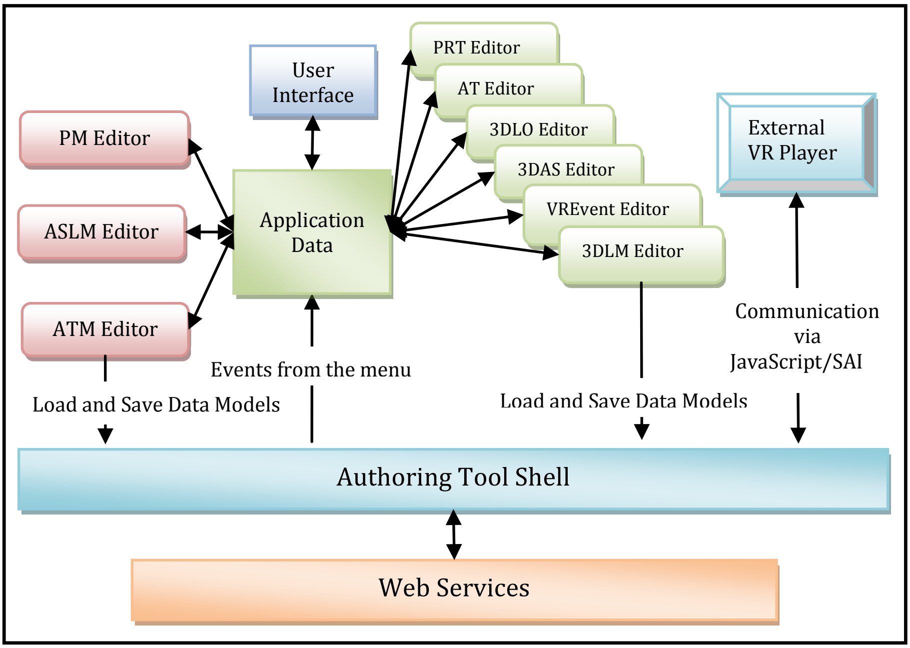


Figure 9.4: Information exchange in the authoring tool (based on GRAPPLE architecture (Kleinermann & De Troyer, 2009))

9.5 Graphical User Interface Prototypes

The previous sections discussed the architecture of the proposed authoring tool, as well as the process flow used for authoring an adaptive 3D VLE and the information exchange between the different editors. We also briefly described the different editors involved. In this section, we present the graphical user interface for the different editors that compose the authoring tool.

Note that the proposed prototype is based on the feedback of the evaluation session done for the authoring tool developed in the context of GRAPPLE (Ewais & De Troyer, 2014, 2013b).

The remainder of this section is structured as follows. In section 9.5.1, a description of the Graphical User Interface (GUI) of the PRT editor is given followed by a description of Pedagogical Model Editor interface in section 9.5.2. After that, the preview tool to display 3D resources is presented in section 9.5.3. Also the GUI of the 3D Learning Object Editor is presented in section 9.5.4. Then the user interfaces to create 3D adaptation states and VR events are discussed in section 9.5.5 and section 9.5.6 respectively. Section 9.5.7 presents the GUI for the 3D Learning Material Editor. Then different interface specifications related to the Adaptive Storyline Editor and the Adaptive Topic Editor are discussed in section 9.5.8 and 9.5.9 respectively.

9.5.1 PRT Editor

The main task of PRT editor is to provide the possibility to create and define pedagogical relationship types (PRTs).

The user interface of the PRT editor is organized using different tabs that allow an author (Domain Expert or Course Author) to manipulate different PRT specifications. The author is able to switch between General tab, User Model tab, Update Rule tab, and Console tab to define the required information. The following are explanations and screenshots related to each tab.

Starting with the *General* tab, which is displayed in Figure 9.5, an author can define the general PRT properties such as Title (or name) which should be unique for every PRT. Another property is the Description that allows providing a full description of the pedagogical meaning and/or the purpose of the PRT. Furthermore, the AuthorID property is required to identify the author who created this PRT. Besides AuthorID property, Created and Modified are auto-generated properties that store the time the PRT is created and modified respectively. Finally, the PRT Colour property specifies the specific colour of the PRT.

Figure 9.5: General tab to enter meta-data for a PRT.

The *User Model* tab is used to specify the variables from the User Model that will be accessed and updated in the associated Pedagogical Update Rule (specified in the Update Rule tab). A list of variables can be given and for each variable, different properties can be defined (see Figure 9.6). Examples of user model variables are knowledge and visited. A user model variable has a value that is defined by a type property, a default value, and a range property. It can also have a location to load the value of the variable if it is coming from elsewhere, e.g., from an external learning management system.

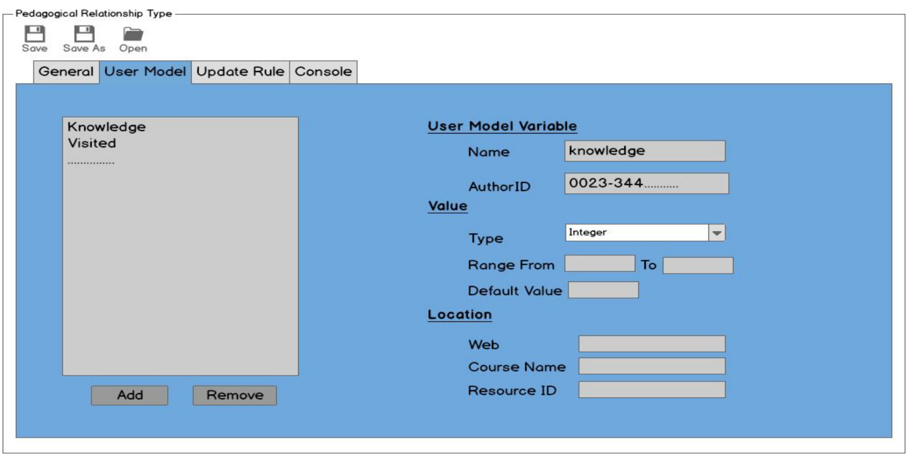


Figure 9.6: User Model Tab to define the required user model attributes.

Once the required user model variables are specified, the *Update Rule* tab is used to compose and define the syntax of the associated Pedagogical Update

Rule. This is achieved by two panels (see Figure 9.7). The author can either use the right panel to enter the rule directly, which assumes that the author has a good knowledge about the language, or the author can use the left panel that helps an author to compose the rule. For that, the author can use the following menu panels:

- Condition Panel: this menu panel displays the terminology available to formulate the condition of the IF-Then statement.
- User Model Variable Panel: this menu panel includes the user model variables that could be accessed and updated.
- Operators Panel: this menu panel shows the operators that can be used in the update rule. For instance, "greater than" is represented as ">", "greater than or equal" is represented as ">=", etc.

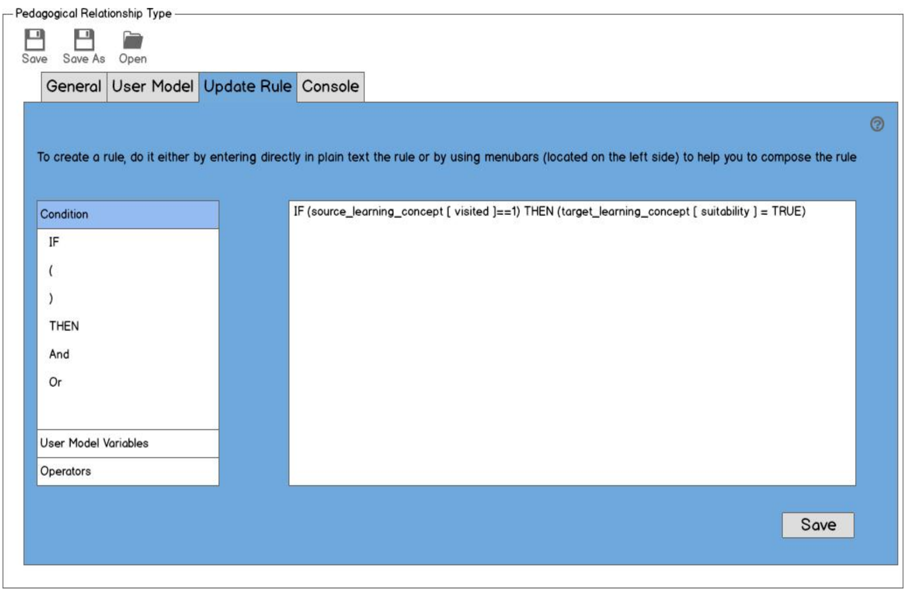


Figure 9.7: Update rule tab can be used to define the associated pedagogical update rule.

The *Console* tab is mainly used by the advanced authors to customize a PRT using XML code (see Figure 9.8). We will not elaborate on this. See XML schema for PRT given in (http://www.ewais.be/Phd/PRT). The code will be interpreted by the adaptive engine to process the defined update rule.

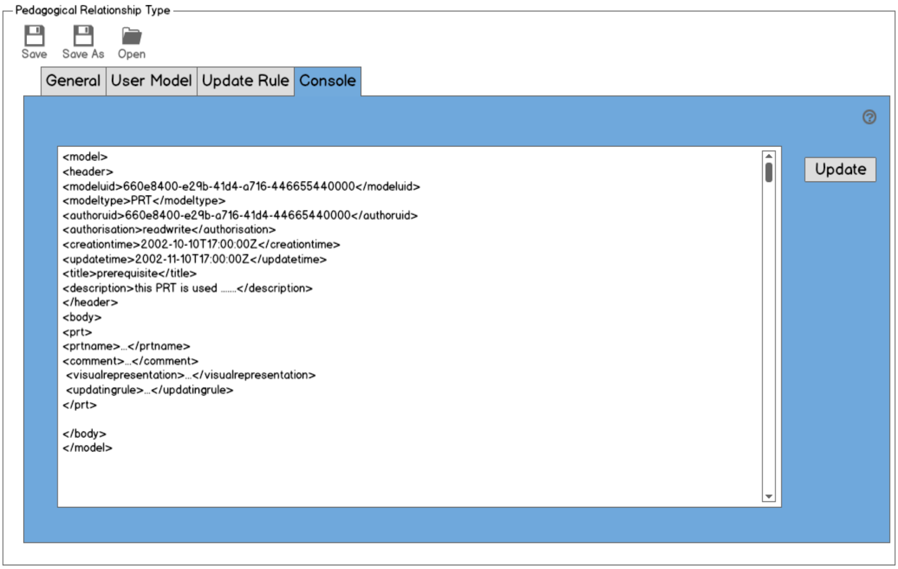


Figure 9.8: Console tab to customize the PRT code.

The output of PRT Editor is a set of pedagogical relationship types that can be used in the Pedagogical Model Editor. The next section will explain this Editor.

9.5.2 Pedagogical Model Tool

The main task of the Pedagogical Model Editor is to enable authors to define the PRTs between the different learning concepts.

The user interface of the Pedagogical Model Editor is divided into two tabs (see Figure 9.9). The first tab (*PM visual Environment*) is the main one and is used to create a Pedagogical Model visually; the second tab is the *Console* tab, which is used to display the pedagogical model in XML format (for the advanced author).

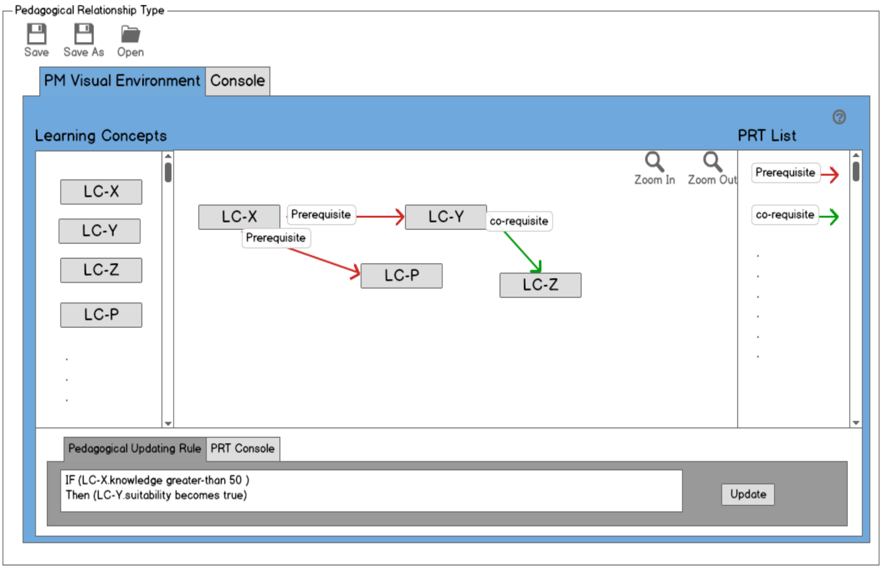


Figure 9.9: Main screen for creating a Pedagogical Model using the visual pedagogical model language.

The window for the first tab is split into four panels as follow:

- The left panel is used to list all learning concepts (obtained from the Domain Model) that are related to the course. Furthermore, all learning concepts are represented as rectangles that can be dragged and dropped to the middle panel (i.e. canvas).
- The right panel is used to display all PRTs (created with PRT Editor (see section 9.5.1)). Moreover, all PRTs are represented as arrows that can be dragged and dropped to the middle panel. These arrows are used to connect learning concepts. Together, they form a certain pedagogical structure for the course.
- The middle contains the canvas used to visually create the Pedagogical Model. It also enables authors to zoom in or zoom out to be able to view parts or the overall pedagogical structure.
- The fourth panel (at the bottom) is used to edit the PUR associated with the selected PRT. The author is able to modify or use the default PUR and save changes to the PUR. The screen for editing a PUR is shown in Figure 9.10. This panel is intended to make it easy for the author to modify the associated PUR by using simple graphical components like combo boxes and input text fields. On the one hand, combo boxes are used to either select user model attributes like knowledge or suitability, and to select operators like “is greater than” or “is equal to”. On the other hand, an input text field is used to enter the values that will be assigned to the selected user model variables.

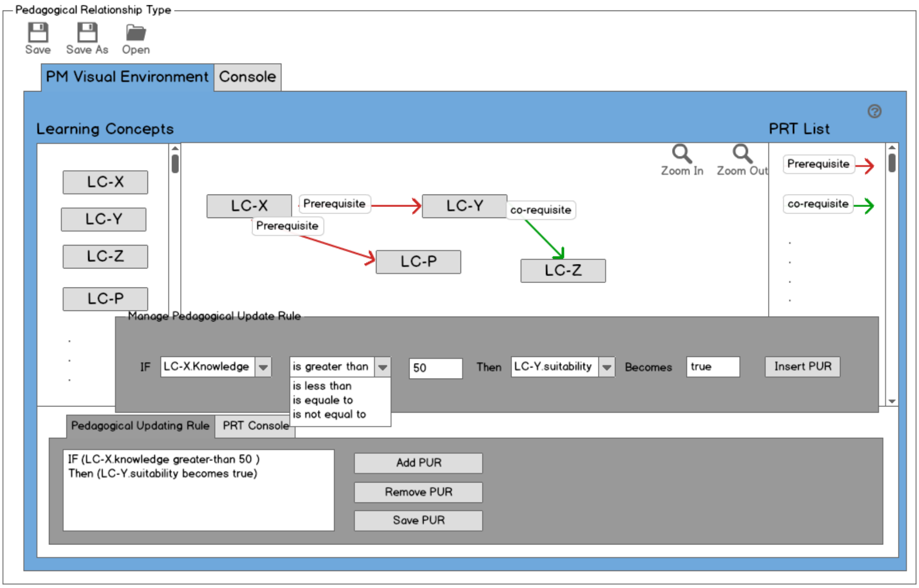


Figure 9.10: Manage pedagogical update rule screen to define criteria to update user model attributes.

9.5.3 Previewing 3D VLE Resources

The adaptive 3D VLE authoring tool allows the author to preview VR/3D resources using a previewer. The previewer can be launched from the main screen of the authoring tool. It is also integrated in the VR Event Editor, the 3D Adaptation State Editor, and the 3D Learning Material Editor to provide a visualization of the 3D models. Figure 9.11 shows a screenshot of the previewer. The previewer can be realized by using an existing 3D/VR player (like Vivaty, Flux, ...).

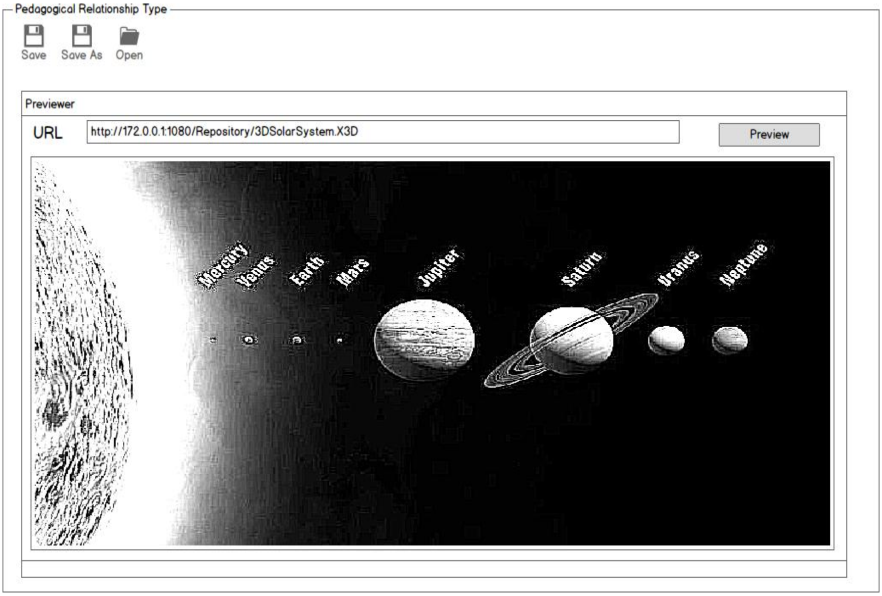


Figure 9.11: Previewing a VR resource.

9.5.4 Adaptation Theme Editor

The goal of this editor is to enable the author to create adaptation themes by customizing a group of adaptation types.

One of the important advantages of this editor is that it helps the author to have consistent specifications for the different adaptation types that are used for a group of users. Therefore, the author does not need to customize every single adaptation type used in the context of adaptive topic model. However, the author is still able to perform modifications to adaptation types individually by using the 3D Adaptation State Editor (see section 9.5.8).

We believe that classifying learners into groups can best be done in the context of creating the user model. Therefore, we suppose that the authoring tool can get a list of learner groups from a user model application using a certain web service.

The user interface of the editor contains an adaptation theme tab, which is organized into three panels, a panel for the preview tool, a Meta-Info tab, and a Console tab, which contains the XML specifications for the adaptation theme. The interface of this editor is depicted in Figure 9.12.

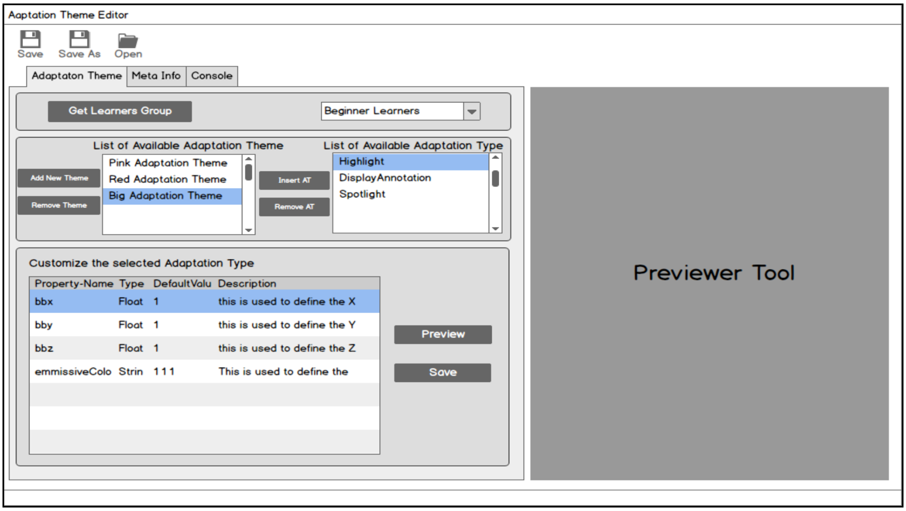


Figure 9.12: Identifying 3D objects.

The first panel in the adaptation theme tab is used to select a learner group. To retrieve the available learner groups, the *Get Learners Group* button is available. Next, the user will be able to select one of the available learner groups. The second panel enables the author to either select an adaptation theme already associated with the selected learner group, or to add a new adaptation theme to the selected learner group (see Figure 9.13). After that, the author is able to select/add a pre-existing adaptation type to associate them with the defined adaptation theme using the *insert AT* button (see Figure 9.14).

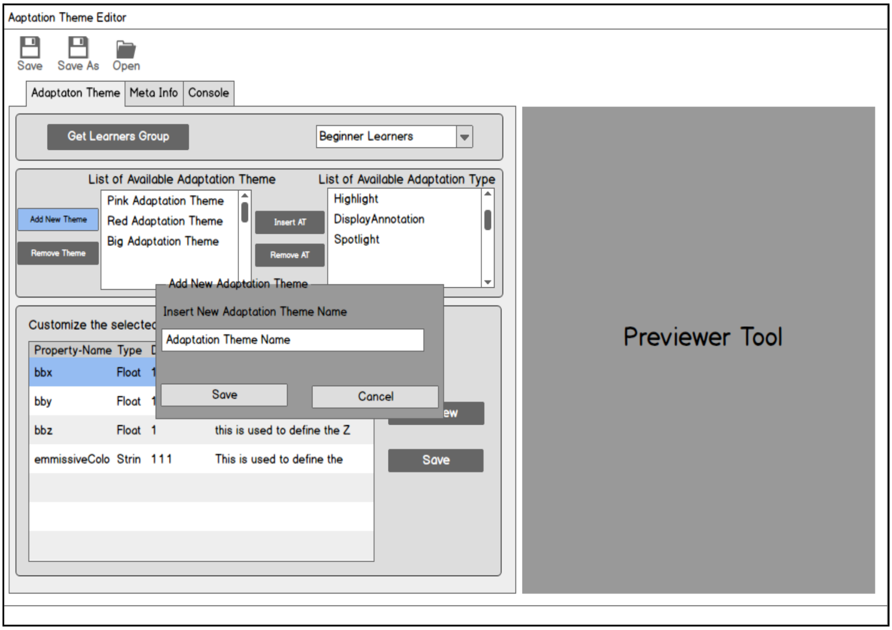


Figure 9.13: Insert new adaptation theme.

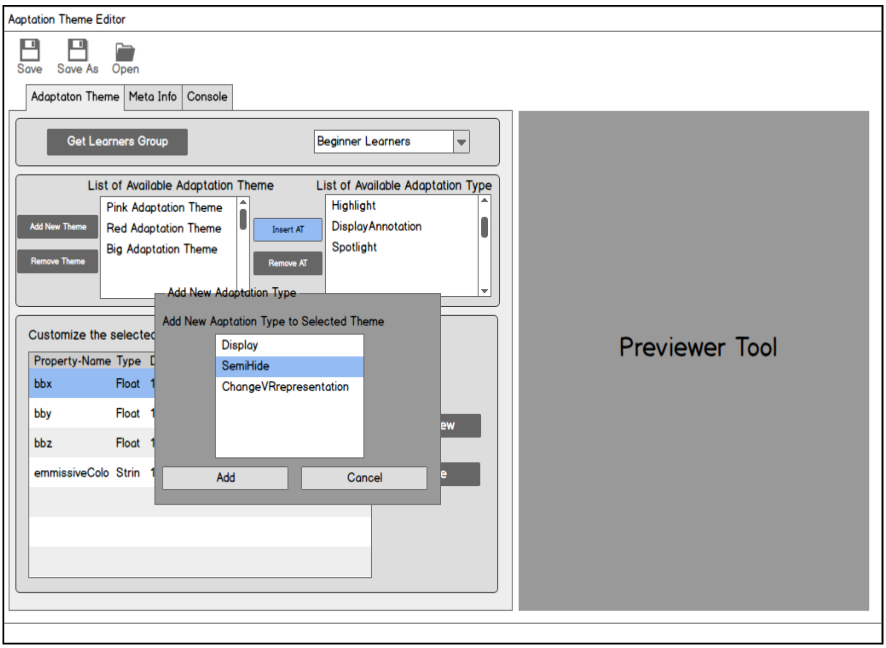


Figure 9.14: Add new adaptation type to defined adaptation theme.

The third panel is used to perform modifications to the selected adaption type. For instance, the author can change the values of the adaptation type's properties such as colour and size. Finally, the author can preview the adaptation type using the previewer tool.

Finally, as shown in Figure 9.15, the Meta-Info tab is used for the adaptation theme's name, the description that explains the goal of the adaptation theme, the AuthorID that identifies the author who created the adaptation theme, the created time and modified time.

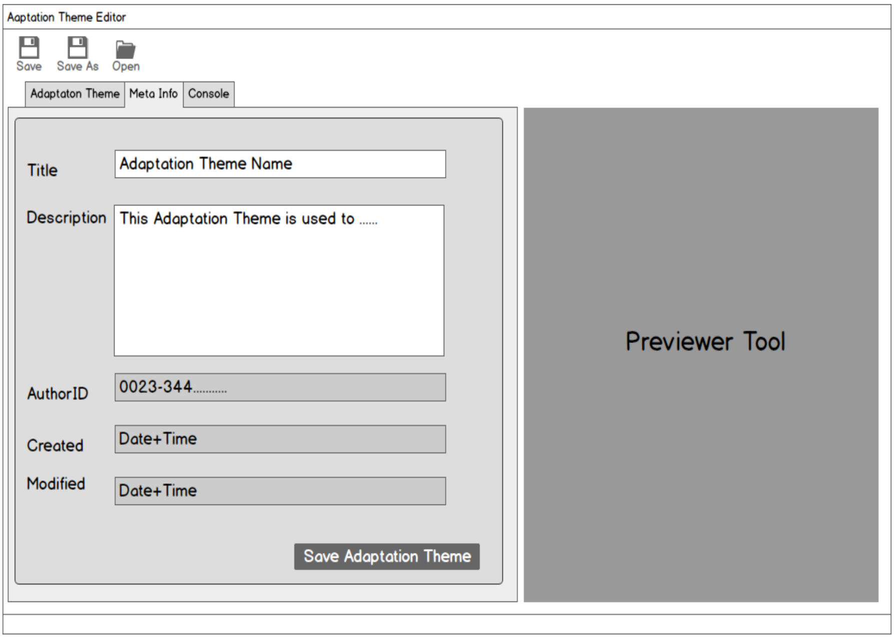


Figure 9.15: Meta-Info tab is used to display adaptation theme meta data.

Once the author is satisfied with the specified adaptation theme, he can store the results by pressing the button *Save Adaptation Theme*. This will invoke a web service to store the file in database with the ".atdm" file extension (see resulted XML schema in http://www.ewais.be/Phd/atdm).

9.5.5 3D Learning Object Editor

The main goal of this editor is to allow authors to identify the required 3D learning objects (3DLO) in a 3D resource by giving them a specific ID.

The authors need this editor because most of the available 3D resources may not always have proper IDs for the different 3D objects. Moreover, when 3D objects have IDs, they may not have meaningful names, at least not from the author's viewpoint. However, these IDs are important for an author, as later on he needs to use them to map these 3D objects to the domain concepts of the course. In addition, identified 3DLO are used in the adaptive delivery environment to identify which 3D objects should be adapted at run time. Therefore, annotating the required 3D objects with meaningful IDs is needed.

The user interface of this editor is organized as two panels that are used to display the X3D structure of a 3D resource as a scene graph based structure (tree structure). Using this structure, the author can select the nodes that he wants to identify as 3DLOs. The first panel displays the original scene graph (see left side of Figure 9.16) and the second one displays the annotated scene graph (with the identified 3D learning objects) (see right side of Figure 9.16).

In the middle part of the screen, located between the two panels (see Figure 9.16), there are two text boxes. The top one displays the node ID selected by the author in the left tree. The bottom text box is an input box and allows the author entering the ID that he wants to assign to the node. In Figure 8.16, the author has selected the node "*EarthCoordinateSystem*" and entered the name "*Earth*" in the lower input box to change it. Each node, for which the ID has been changed, will be given the prefix "*Author_*". So, the node "*EarthCoordinateSystem*" will be identified as "*Author_*Earth" in the rest of the system. This helps the author to see which nodes he has changed.

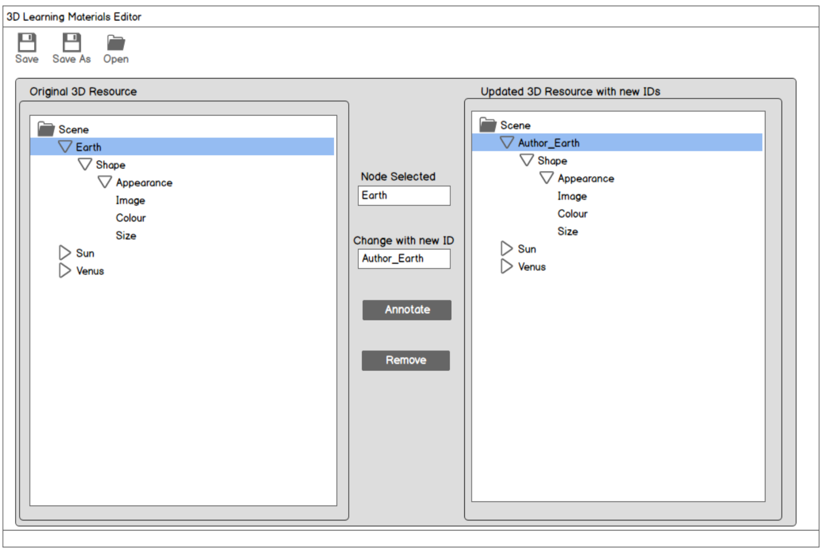


Figure 9.16: Identifying 3D virtual objects.

The *Save the X3D* button is used to store the results. This will invoke the web service to store the file in database as a file with the ".3dlo" file extension (see resulted XML schema in http://www.ewais.be/Phd/3dlo).

9.5.6 3D Adaptation State Editor

This editor is used to define the 3D Adaptation States (3DAS) that are needed in the course. Recall that the term 3D Adaptation State (3DAS) refers to an instantiation of an adaptation type or adaptation strategy. For instance, an author may create a 3DAS called *EnableVRStateMark,* which is a customized instance of the adaptation type *Highlight*. As this 3D adaptation state displays a box around a 3D object, the author can customize it by specifying the size of the box and/or its colour.

Such a 3D adaptation state can be used for different 3D learning material in different course contexts.

The user interface of this editor is organized as three tabs (see Figure 9.17), which are the 3DAS Settings tab, the Meta-Info tab, and the Console tab.

The *3DAS Settings* tab enables authors to customize existing 3D adaptation type or strategy by enabling authors to upload a 3D resource. This is achieved by using the *Upload Resource* button. As a result, the name of the uploaded 3DAS and its different properties are displayed in a panel below. This data can be customized by the author. For instance, the author can change the name or add a new property to the uploaded 3DAS or remove one (using *Add* or *Remove* button). For each property, the author will have to specify the default value as well as the type of property. Note that these properties refer to parameters in the X3D code of the 3DAS. Back to the *EnableVRStateMark* example, the parameters are the size of the box along the X-axis, the Y–axis, and the Z-axis, as well as the colour.

At the same time, the tab enables authors to preview the 3DAS after changing its properties by using the *Preview 3D Adaptation State* button. Finally, the author can save the defined 3DAS by using the *Save 3D Adaptation State* button.

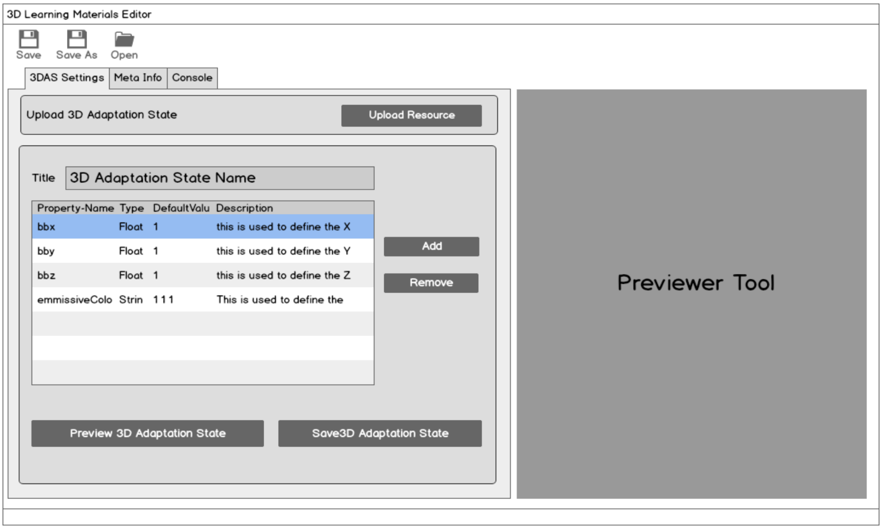


Figure 9.17: 3DAS settings tab is used to customize different properties for adaptation types and strategies.

As shown in Figure 9.18, the *Meta-Info* tab is used to display the meta-data of a 3DAS: its name, description (which explains the goal of the adaptation state), authorID (which is the author who created the 3DAS), creation time, and modification time. The author may save modifications by using the *Save 3D Adaptation State* button. The file is saved as a file with extension ".3dasm". The XML schema for 3DASM is presented in (http://www.ewais.be/Phd/3dasm).

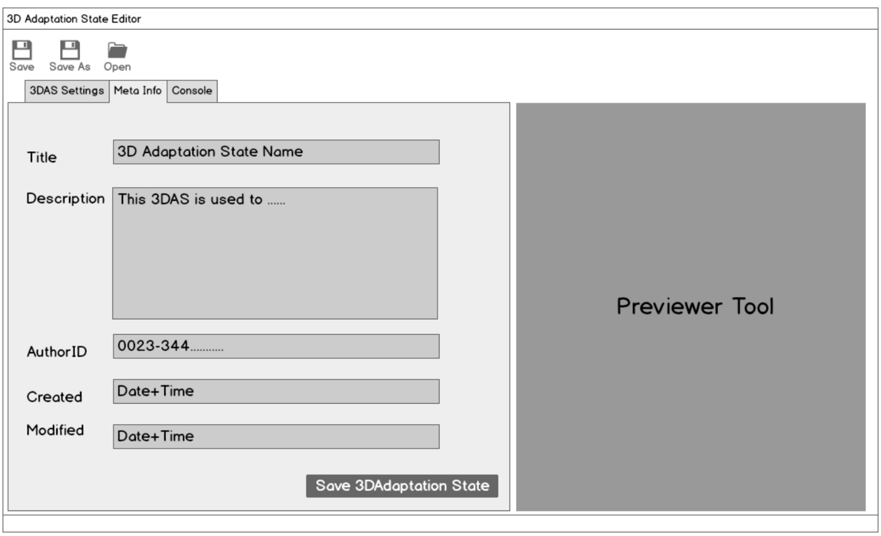


Figure 9.18: Meta-Info tab is used to display 3DAS Meta data.

9.5.7 VR Events Editor

A VR event Editor support authors in defining VR events. As we are dealing with 3D resources based on X3D, VR events correspond to *sensor-nodes*[60] in X3D such as touch-sensor or time-sensor. In principle, this editor is intended for advanced authors who are familiar with 3D technology. However, the tool comes up with predefined VR events based on supported sensors for X3D resources.

A VR event is an important component of the adaptation rules used in the context of the Adaptive Topic Model Editor (see section 9.5.10). A list of possible VR events that can be applied to 3D learning objects is as follows:

- ***Timer*** generates an event after a predefined time. Moreover, there are start and end properties for the timer event. For instance, a course author may like to deactivate 3D object behaviour after a specific time of its activation. To do so, the time sensor node in X3D can be used. For the time sensor XML schema see (http://www.ewais.be/Phd/vre/timer).
- ***Close To*** generates an event when the 'viewer' (user) enters, exits, or moves in a predefined area around a 3D object. Usually, the area is defined by box boundaries. To achieve that, a proximity sensor in X3D can be used. The proximity sensor specification is given in (http://www.ewais.be/Phd/vre/closeto).
- ***Collision*** is triggered when an object collision is detected. Also a proximity sensor can be used for this. The XML schema for proximity sensor is presented in (http://www.ewais.be/Phd/vre/collision).
- ***StringEntered*** generates an event when the user enters a predefined string (sequence of characters) using the keyboard. This can be realized by a String Sensor in X3D. See (http://www.ewais.be/Phd/vre/string).
- ***Touch*** triggers an event when a user clicks on a 3D geometry. This can be realized in X3D by using the so-called Touch sensor (see (http://www.ewais.be/Phd/vre/touch)).
- ***KeyPressed*** generates an event when the user presses a key on the keyboard. The Key Sensor can be used to implement this VR event (see (http://www.ewais.be/Phd/vre/keypressed)).

The graphical User Interface for this editor is almost the same as the one defined for the 3D Adaptation State Editor. The interface has three tabs, which are the *VR Event Settings* tab, the *Meta Info* tab, and the *Console* tab (see Figure 9.19).

[60] http://www.web3d.org/x3d/content/X3dTooltips.html

The first tab enables authors either to upload and customize existing VR events or to define new VR events. Moreover, when uploading and customizing VR events, the tab will display the VR event title and the different properties that are related to the selected VR event. Furthermore, the editor enables the author to change properties values or add new properties. However, the author is also able to define a new VR event by giving it a name and adding its properties.

Every property has a name, type, values, and description, which can be customized directly inside the corresponding panel. Furthermore, new properties can be added or removed using the *Add* or *Remove* buttons.

The same tab enables authors to preview the VR event after modifications by using the *Preview VR event* button. Finally, the author can save the defined VR Event by using *Save VR event* button.

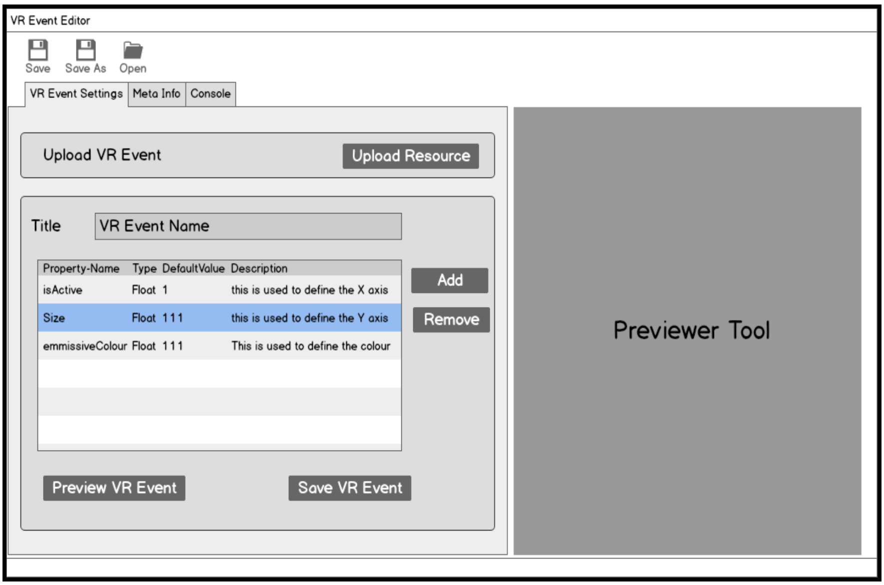


Figure 9.19: VR event settings tab is used to specify VR event properties.

Meta-Info and *Console* tabs have the same interface as in the 3D Adaptation State Editor. The *Meta-Info* tab displays name, authorID, and creation and modification time. The *Console* tab displays the XML specifications for the uploaded VR event.

The XML schema for VR Event Data Model, which is the output of this editor, can be found in (http://www.ewais.be/Phd/vrdm).

9.5.8 3D Learning Materials Editor

The 3D Learning Material (3DLM) Editor allows the author to select 3D Learning Object resources (prepared by the 3D Learning Object Editor), to associate them with possible 3D adaptation states (defined in the 3D Adaptation State Editor), and to apply VR events (defined in the VR Event Editor) to the selected 3D learning object resources.

The main goal of this step is to have a library of 3D resources with required 3D adaptation states and possible VR events. Such library will provide the course authors with reusable resources without the need to create them for every new course from scratch.

The user interface of this editor has three tabs: *Associate 3D Adaptation State*, *Associate VR Event*, and *Console* (see Figure 9.20).

In the *Associate 3D Adaptation State* tab, the author needs to upload the required 3D VLE with its annotated 3D learning objects, i.e. 3D Learning Objects (3DLOs). After that, a list of identified 3DLOs will be displayed in a combo box where the author can select a 3DLO. If the selected 3DLO has associated 3D Adaptation States (3DASs) then the author will be able to view them from the corresponding combo box. Otherwise, the author has the ability to add new 3DASs by adding them from the repository using a third combo box. The author is able to customize the 3DAS by changing its properties. Furthermore, the author is able to preview all applied 3D Adaptation States.

For instance, the author may want to use the 3D Adaptation State *VRStateMark* for the 3D Learning Object Sun, but he would like to change the edges of the box to blue rather than using the default red colour, and change the size of the box as well.

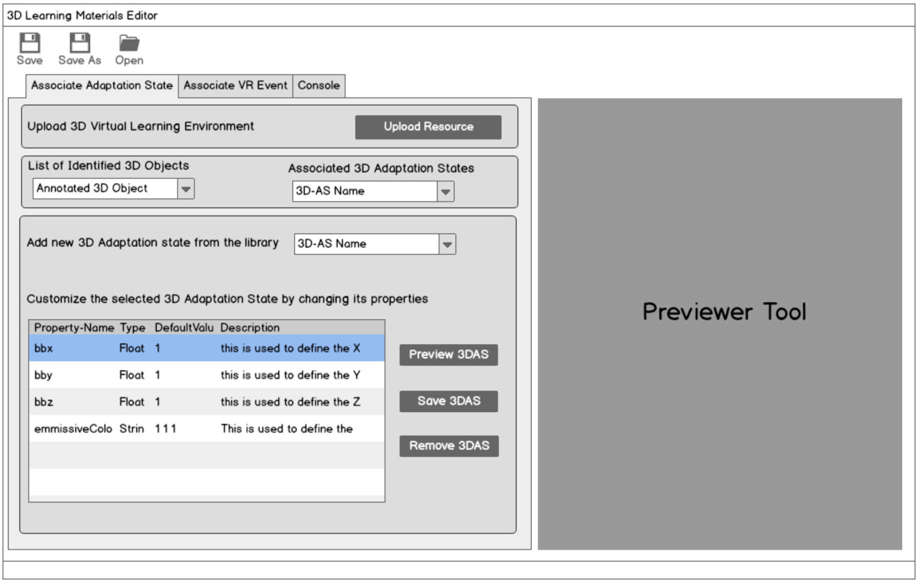


Figure 9.20: Associate adaptation state tab is used to apply customized adaptation types and strategies to identified 3D learning object.

In the *Associate VR Event* tab, the author can also select an annotated 3D Learning Object and assign VR events to it, which will be used in the context of Adaptive Topic Model Editor (see section 9.5.10). Furthermore, the author will be able to preview any modifications applied to the associated VR events (see Figure 9.21).

For instance (see Figure 9.21), the author may need the *timer* VR event for the 3D Learning Object earth and he may want to customize when it should start rotating around sun and when it should end. Also he can define if it is enabled or disabled by default.

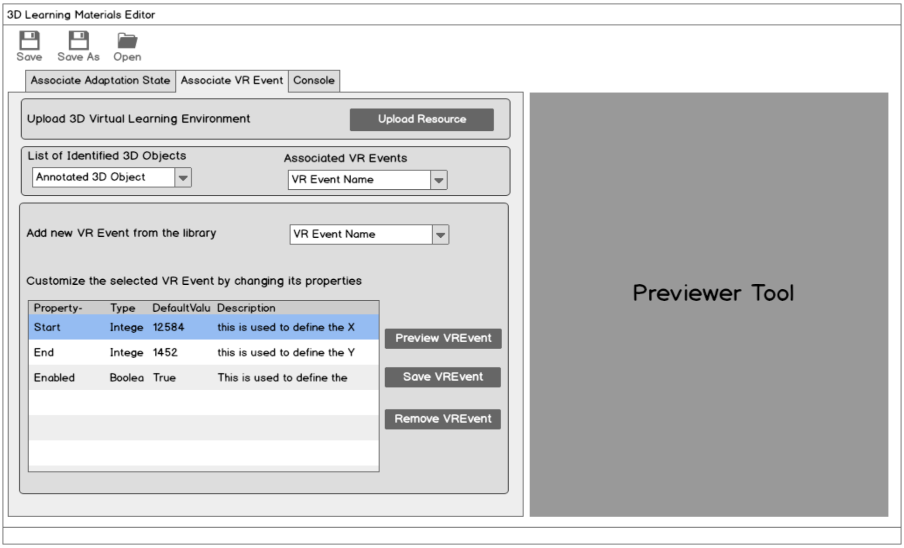


Figure 9.21: Associate VR Event tab is used to assign different VR events to identified 3D learning objects.

The *Console* tab is used to display the XML specifications for the 3D learning materials. The XML schema can be found in (http://www.ewais.be/Phd/3dlm)

9.5.9 Adaptive Storyline Modelling Editor

This tool enables the author to use the Adaptive Storyline Model language to create an adaptive storyline. Furthermore, the tool enables the author to apply an adaptation theme.

The graphical user interface of the tool is divided into two tabs (see Figure 9.22). The first tab enables author to create the adaptive storyline using the proposed modelling language. The second tab contains the console that displays the XML format of the adaptive storyline, which could be used by an advanced author.

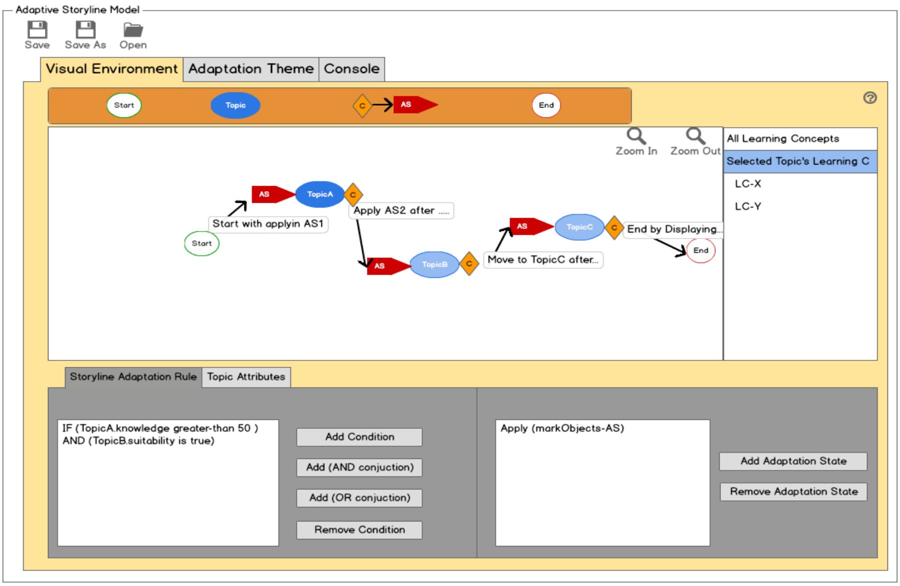


Figure 9.22: Main screen of the adaptive storyline model editor.

To guide the user in creating an adaptive storyline, the first tab is split into two parts (see Figure 9.22). The upper part includes four panels while the bottom part includes two sub-tabs, being the *Storyline Adaptation Rule* sub-tab and the *Topic Attributes* sub-tab. We elaborate on the details of each part.

As mentioned earlier, the first part includes four panels to allow authors to specify the adaptive storyline using the visual language provided. This tab has the following structure.

The upper panel provides the visual notations of the adaptive storyline language symbols (*start, topic, storyline adaptation rule, and end*). The visual notations can be dragged and dropped to the canvas, which is located in the middle panel. The author can create instances of topics and adaptation rules which are composed of condition symbols and adaptation strategies notations. Once the topic symbol is dragged and dropped on the canvas, a topic window will be displayed to enter the name of the topic and to assign the related learning concepts (see Figure 9.23). Furthermore, the middle panel provides a zooming functionality to provide an overview or view details of the adaptive storyline created.

Moreover, the right panel can be used to display all learning concepts, as well as all learning concepts associated with the selected topic using an accordion panel. So, the author will be able to view corresponding learning concepts for each topic during the creation of the adaptive storyline.

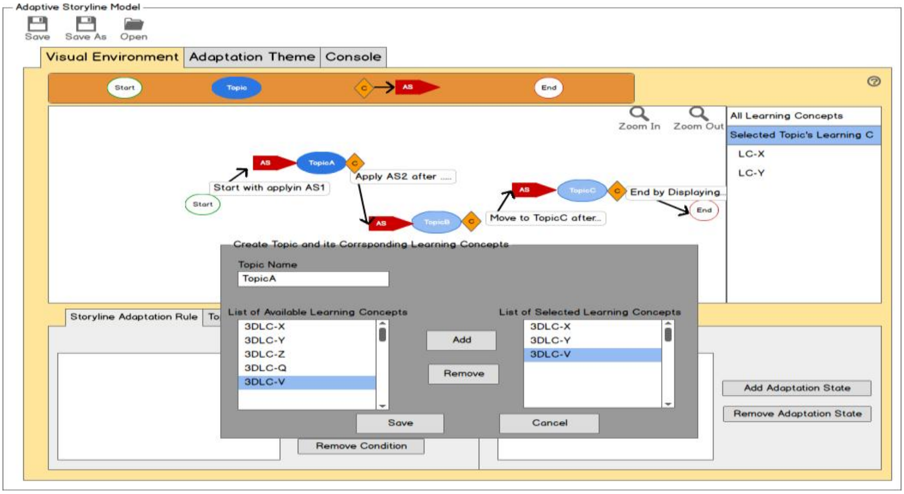


Figure 9.23: Create a topic and insert related learning concepts.

Since it is crucial that authors are not overwhelmed with the technical details of the adaptation rule language, we have opted for allowing them to define the adaption storyline rules by means of a graphical user interface as it is shown in Figure 9.24. As a result, the lower part of the Visual Environment tab is mainly used to define the storyline adaptation rules associated with the topics and to define how topic's attributes should be updated. To achieve that, there are two sub-tabs in this part.

The first one is called the *Storyline Adaptation Rule* sub-tab. This sub-tab is also divided into two parts. The first part (on the left) is related to the condition part of the storyline adaptation rule and is using a popup window, which displays the format of the condition part by means of combo boxes and input field to make it easy for novice authors to define the condition part.

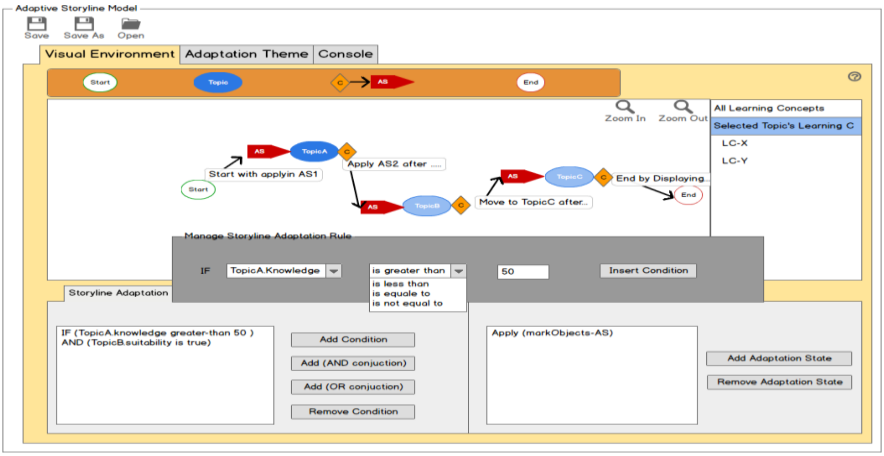


Figure 9.24: Create the condition part of the storyline adaptation rule

The second part (on the right) is related to the action part of the storyline adaptation rule (see Figure 9.25). This part is provided to assign adaptation states to the targeted topic in a storyline adaptation rule. To do so, a popup window is displayed to support the author in selecting adaptation states and entering the text message to be displayed when the adaptation state will be applied to the target topic. A list of adaptation strategies are displayed by means of a combo-box and a text area input field is provided to enter the text message.

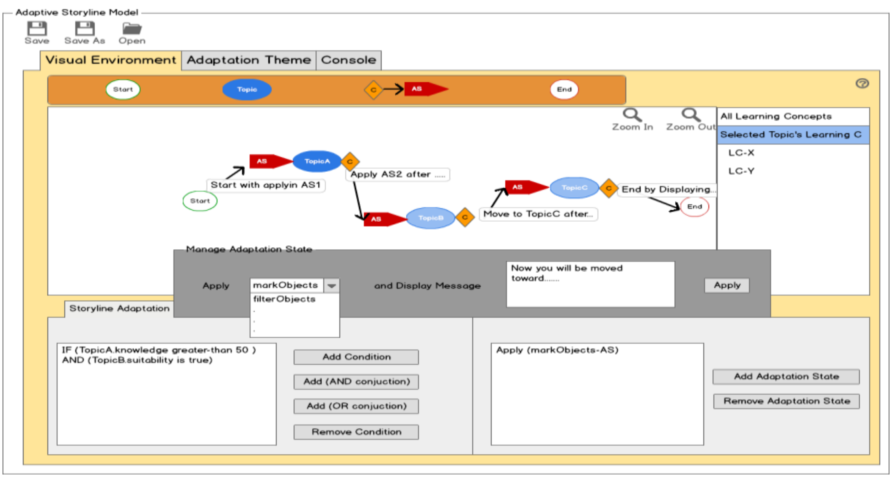


Figure 9.25: Apply an adaptation strategy to selected topic.

Second sub-tab in the lower part is called *Topic Attributes* (see Figure 9.26). This sub-tab enables the authors to define how topic attributes like knowledge and suitability should be updated. This is done by first selecting the corresponding attribute and after that the author can write directly the update mechanism (text input). For instance, the *knowledge* attribute may have the following update mechanism: increasing the topic's knowledge level is based on the value of the related learning concepts' knowledge attribute. Default updating mechanisms are always provided. However, an advanced author is able to change it to fit his situation.

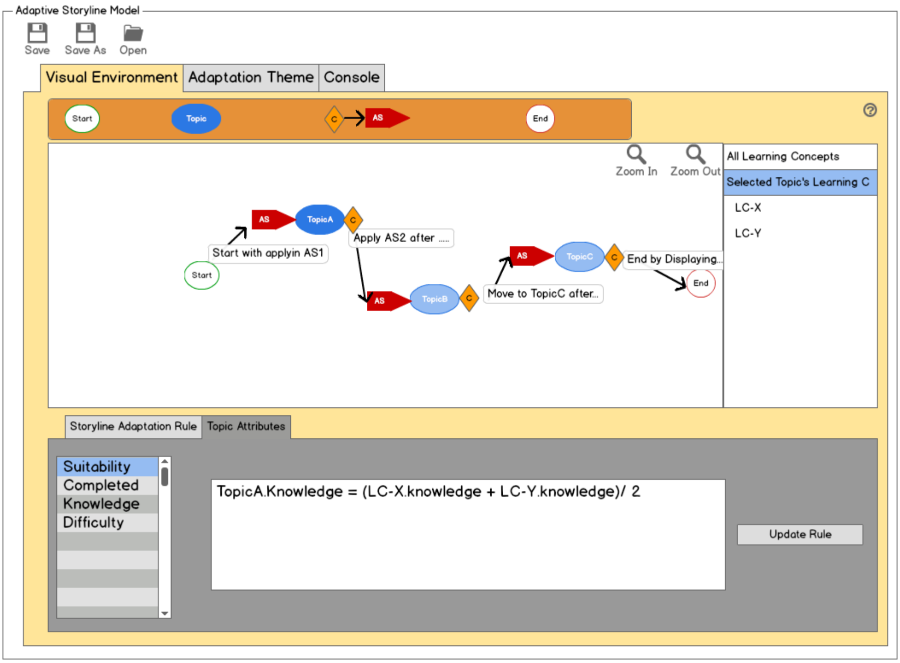


Figure 9.26: Update mechanism to the topic's knowledge attribute.

The author is also able to select an adaptation theme to be used (see Figure 9.27). The author is able to select the group of learners from a list. According to the selected group, the author will be able to view and select one of the adaptation themes that are associated for the targeted group. Once the author selects the adaptation theme, he is able to read the description and the associated adaptation types. The author can also use the adaptation theme to the selected group by using the *use selected AT* button. Remember that the selected adaptation theme will be applied to all defined topics.

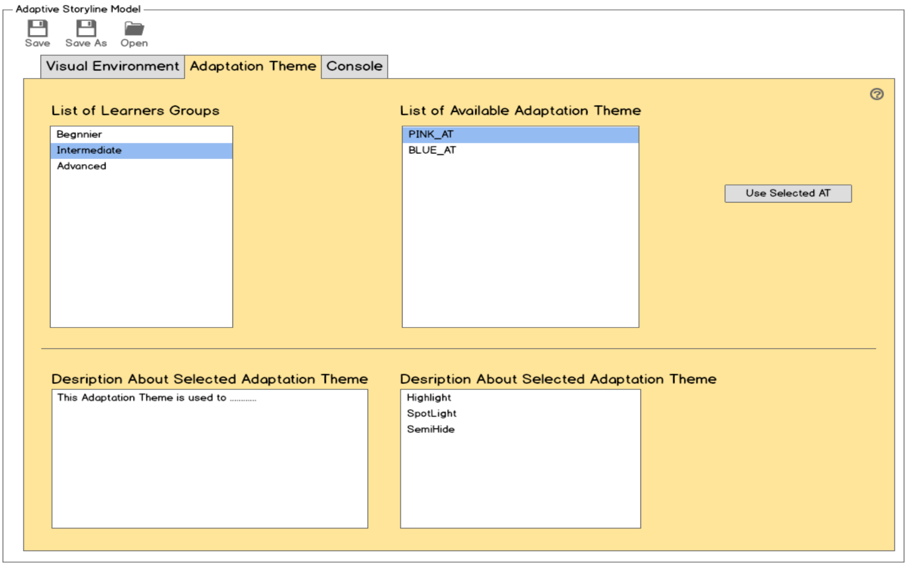


Figure 9.27: Select an adaptation theme to be applied during the learning process.

The author is able to save the Adaptive Storyline Data Model as XML file by using the *save* button in the right top of the screen.

9.5.10 Adaptive Topic Model Editor

The Adaptive Topic Model Editor is used to define when the adaptation types associated with the 3D learning concepts should be applied inside a topic.

Similar as for the previous editor (Adaptive Storyline Model Editor), this tool has two main tabs. The first tab allows the author to specify the required adaptation aspects inside a single topic. The second tab, which is the *Console* tab, is used, as the other *Console* tabs, to display the XML specifications for this model. See Figure 9.28.

To guide the author in creating the adaptation flow inside a topic, the first tab is split into two parts. The upper part is organized in four panels, while the bottom part includes two sub-tabs, being the *Topic Adaptation Rule* sub-tab and the *Console* sub-tab. We elaborate on each part.

The upper part is used to enable authors to create the adaptation inside a single topic using the proposed Adaptive Topic Modelling Language. As mentioned earlier, it is divided into four panels which facilitate authors to do the following:

The author is able to use the visual language using the upper panel, which includes the visual symbols of the modelling language (3D Learning Concept, VR event, Adaptation Type, and Notification Message). He can drag and drop the symbols to the canvas, which is located in the middle panel.

Once the author dragged and dropped a learning concept symbol, a new window will pop up to select a learning concept, which is listed in the panel. Figure 9.29 displays all learning concepts related to TopicA.

Moreover, the author uses the right panel, which includes an accordion component, to display both the associated adaptation types and VR events with the selected learning concepts. For instance, the associated VR events with learning concept so-called 3DLC-X are the *timer* and *touch* VR events.

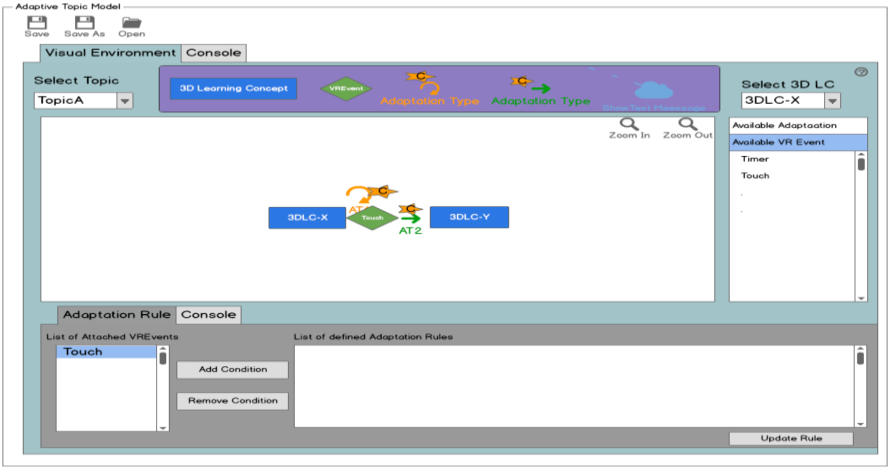


Figure 9.28: Main screen of the adaptive topic model tool.

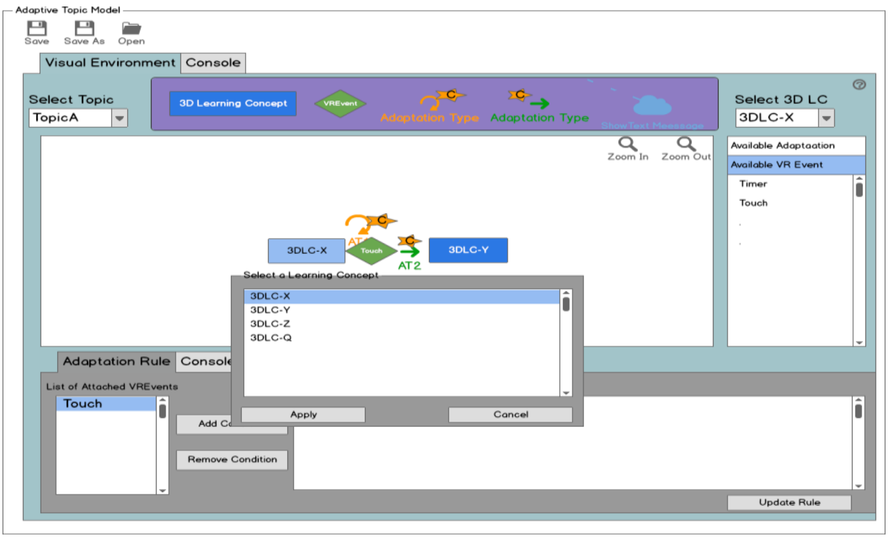


Figure 9.29: Assign a learning concept to the new inserted 3D learning concept symbol.

Similar as for the Adaptive Storyline Model Editor, this editor is also intended to be an easy to use user interface to define the adaptation rules. As a result, the bottom part displays two panels (see Figure 9.30). The left panel displays the list of VR events associated with the learning concept selected in the canvas. The right panel will display corresponding adaptation rules using the selected VR events from the left panel.

To define an adaptation rule, the tool provides the author with a popup window, which includes the components required to formulate the adaptation rules (see Figure 9.30). Remember that adaptation rules in the context of an Adaptive Topic Model are event-condition-action rules. So, the user interface will enable the author to define these three parts explicitly.

The formulation of the event-part is supported in the following way: One combo box displays the list of VR events associated to the selected learning concept. Another combo box is used to allow authors selecting the operator ("is greater than", "is equal to", etc.), and an input field is used to enter the specific value which need to be used for the comparison.

The formulation of the condition-part is also supported by two combo-boxes, which are used to display the list of learning concept attributes and the list of operators respectively. An input field is used to enter the specific value to be evaluated with the selected learning concept attribute.

The formulation of the action-part is supported by two combo-boxes to display the list of target learning concepts with their associated adaptation types, to allow selecting which ones will be applied in the rule as action.

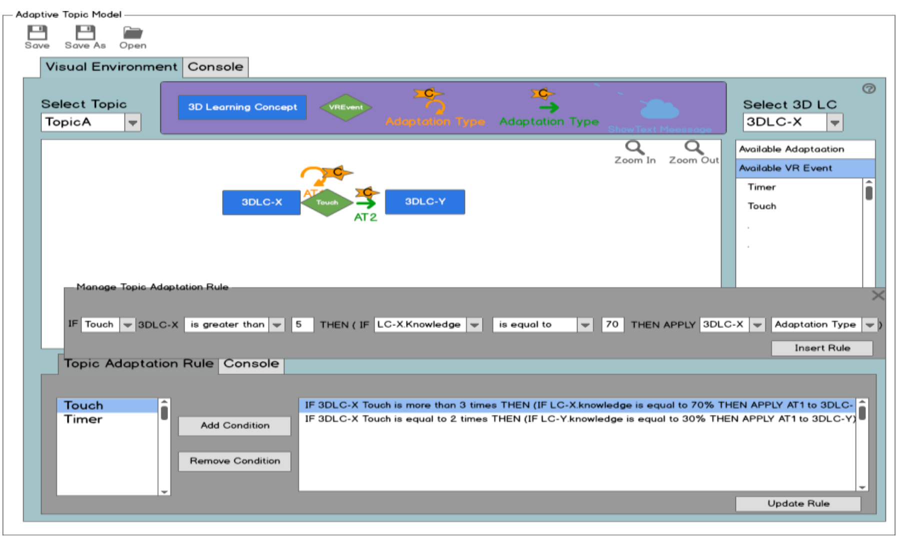


Figure 9.30: Creating an adaptation rule between learning concepts in a topic.

Finally, providing a text message to be displayed, when the adaptation rule is applied, can be specified using the show notification message symbol. A popup window will allow to specify further details about the text message: the name and text to be displayed to learner (see Figure 9.31).

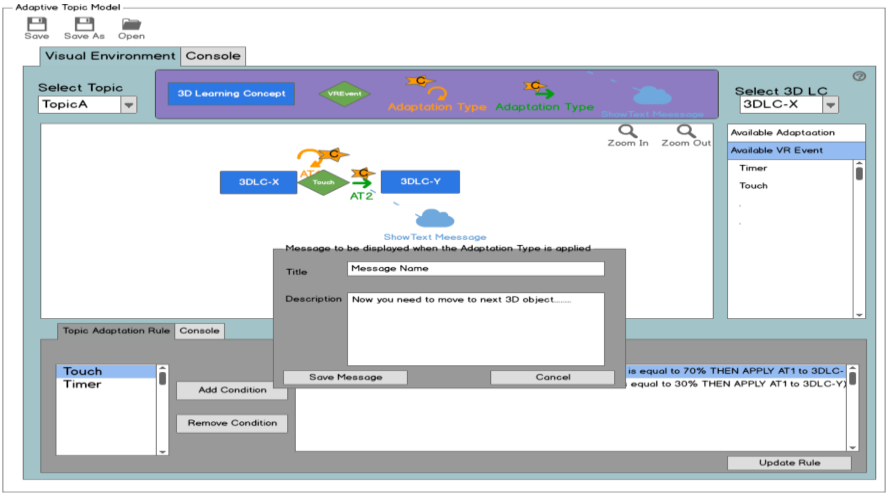


Figure 9.31: Editing text message to be displayed when the defined adaptation rule is triggered.

Every Adaptive Topic Model specification will be stored in a XML format file by using the *save* button in the top right of the screen.

9.6 Summary

This chapter started by describing the functional requirements that need to be satisfied by an authoring tool for our proposed authoring approach. The requirements for this adaptive 3D VLE authoring tool are based on the different scenarios proposed for authoring an adaptive 3D VLE. For example, the author should be able to preview 3D models while he is customizing adaptation types and strategies.

After that, the chapter presents the general architecture of the authoring tool, which is divided to a *Data Model Environment* and a *Pedagogical & Storyline Environment.* Each environment has a number of editors, which are used together to design the adaptive 3D VLE.

Authors are able to create an adaptive 3D VLE by following a number of steps. A number of preparatory steps may be required before being able to specify the 3D VLE using the proposed visual languages. The preparatory steps are related to the preparation of the 3D VLE by identifying 3D objects which will be considered in the authoring process, the customizing of adaptation types and strategies to be applied to identified 3D learning objects, the definition of VR events which are used to formulate adaptation rules, the creation of adaptation theme which are used in the context of the adaptive storyline model, and the definition of pedagogical relationship types which are used to design the Pedagogical Model. This preparation process can be considered as heavy and quite rigid. However, the authoring environment will provide a wide range of repositories related to each step. So the author can share resources, reuse pre-existing ones, or use defaults provided.

Furthermore, the chapter presents how the author defines the Pedagogical Model, Adaptive Storyline Model, and the Adaptive Topic Models using the editors in the *Pedagogical & Storyline Environment*.

Afterwards, prototypes of the user interfaces for the different editors in both environments (the *Data Model Environment* and the *Pedagogical & Storyline Environment*) are discussed. Furthermore, different functionalities to be supported by the proposed authoring tool are discussed too. For instance, the author should be able to drag and drop the graphical notations of the visual modelling languages. The editors are used to hide details irrelevant for the authoring purposes. However, advanced users are still able to use advanced features for authoring special requirements.

The authoring tool is intended to make the authoring process as easy as possible. In particular, the authoring tool tries to minimize the efforts needed and the knowledge required to add adaptation inside a 3D VLE, but the tool cannot completely eliminate such efforts and a minimal knowledge of 3D is required, especially when the author needs to specify and customize adaptation types and strategies.

Chapter 10: Conclusion and Future Research

In the previous chapters we presented the different contributions of this thesis. Firstly, a conceptual framework for adaptive 3D VLEs was described. Secondly, visual languages were devised to support the authoring of adaptive 3D VLEs, and an evaluation of these languages has been performed. Thirdly, a prototype of an authoring tool was presented. In this chapter, we present a summary of the work done and discuss our contributions. Furthermore, we present some new research directions that could be interesting to be investigated and explored.

This chapter is structured as follows. Section 10.1 presents a summary of the work described in this thesis. In section 10.2, a discussion about the main contributions and achievements is given. Section 10.3 presents some limitations of our research and points to future work and research.

10.1 Summary

In the past, most research related to supporting adaptation in e-learning was focused on adaptivity for standard hypertext and hypermedia. 3D VLE is a new emerged trend that can be interesting for adaptive e-learning. In general, a 3D VE includes 3D models, 2D images, text, video, and audio. Furthermore, users are able to interact with the content in different ways, i.e. navigating in the 3D space, interacting with the 3D objects, and performing special tasks through 3D pointing devices, mouse, and keyboard. The research done in this dissertation concerns the use of 3D VEs in the context of e-learning. Moreover, this research work focuses on providing dynamic adaptation for 3D VLEs. It is also important to mention that this research work is related to desktop 3D/VR applications, and more in particular, 3D applications that can be experienced through a web browser.

The goal of this research work was to provide answers for the following main research questions (see section 1.3 for more details):

> **RQ1**: What are possible adaptations that can be applied to 3D VLEs, in order to allow adapting a 3D VLE to the learner's knowledge, skills, and behaviour?

RQ2: How to facilitate the authoring process of adaptive 3D VLE for 3D novice educators?

For providing answers to the previous formulated research questions, we have followed an iterative research design process composed of the six steps as proposed by the Design Science Research Methodology (DSRM) (Peffers et al., 2007). The following summarises how the research aims and objectives have been achieved.

3D VLE components

We started by identifying the different components of a 3D VE (3D virtual environment). This would give us insight into the possible adaptations that can be applied to a 3D VE. Therefore, we divided the 3D VE into the virtual scene, 3D virtual objects, objects behaviours, user interaction, user navigation, communication, and sound. The virtual scene contains all the 3D virtual objects. Furthermore, lights, viewpoints and cameras are considered as components of the virtual scene. Also 2D representations can populate the virtual world. Some 3D objects can perform so-called behaviours like rotation, walking, etc. Users are also able to perform special actions in relation to 3D objects, like picking them up, moving them to different place, and clicking on them. Such actions are called user interactions. On the other hand, users are also able to perform other actions that are not related to 3D objects themselves, like navigating through the virtual environment. To navigate, in general, avatars are used. They can walk, fly, run, etc. Moreover, in some 3D VEs, users can communicate with each other using audio and/or video calls, and/or text messaging. Finally, a 3D VE may also contain sound effects to simulate real world features.

This 3D virtual environment anatomy (described in Chapter 4) made it easy to recognize the different possible adaptation techniques as we will see further.

Adaptation Types and Strategies

We found that there was a need for a clear overview of adaptation mechanisms to be applied to a 3D VLE. Therefore, we performed an in-depth review of different adaptation techniques used in different contexts, like e-learning, e-commerce, and educational games (discussed in Chapter 2). Next, and based on the defined 3D VE anatomy, a number of adaptation techniques are proposed for a 3D VLE, answering research question **RQ1**. We divide them into two categories: Adaptation Types and Adaptation Strategies (explained in section 4.3). Adaptation Types are adaptation techniques that are applied to a single component, like a single 3D object. On the other hand, Adaptation Strategies are used to have adaptation actions for a group of 3D objects. A list of both categories is provided and each adaptation technique is defined by means of a high level specification. An example of an adaptation type that can be applied to single 3D object is *semiDisplay,* which is used to display the 3D virtual object in semi-transparent way. Such an adaptation type can, for instance, be used to inform the user that he is not able to learn about the object yet. The adaptation type *changeSize* can, for instance, be used to draw the user's attention to the 3D object. For the same purpose (drawing user attention) other adaptation types, like *spotlight, semiHide, hide*, *changeMaterialProperties* can be used too. The *displayAnnotation* and *hideAnnotation* adaptation types can be applied to certain 3D objects to associate e.g., a textual explanation to the object and/or identify (name) 3D objects easily. Another subcategory of the adaptation types that can be applied to a single component of the 3D VLE is related to objects' behaviors. *enableBehaviour, disableBehaviour,* and *changeBehaviour* might be used to control the user's behaviour in the 3D VLE. Adaptation types that are related to user interaction, *enableInteraction* and *disableInteraction,* can also be used to control the user's behaviour.

We realized that sometimes we want to have adaptation techniques that have an impact on a group of 3D VLE components. As a result, a number of Adaptation Strategies are proposed. For instance, some defined Adaptation Strategy can be used to allow the user to navigate through the 3D VLE adaptively. For instance, *restrictedNavigation*, *navigationWithRestrictedBehaviour*, *navigationWithRestrictedInteraction,* and *TourGuide* are used to restrict the user's navigation to specific 3D virtual objects. On the other hand, to allow the user to navigate freely inside the 3D virtual world, we propose two adaptation strategies, which are *completelyFree* and *freeWithSuggestions*. Next to adaptation strategies related to navigation, other adaptation strategies that are related to displaying group of 3D virtual objects are defined. For instance, *filterObjects*, *markObjects*, *displayAtMos*t, and *displayAfter* allow adapting groups of 3D virtual objects. Another group of adaptation strategies consists of techniques for adapting 3D objects' behaviours: *behaviourAtMost*, *behaviourAfter,* and *behaviourSpeed*. Finally, the last group of adaptation strategies contains adaptations for limiting interaction according to some conditions. Both *interactionAtMost* and *interactionAfter* are examples of Adaptation Strategies related to user interaction.

To provide consistent adaptation specifications with a 3D VLE, the concept of Adaptation Theme is proposed. An Adaptation Theme defines some default adaptations (explained in section 4.2.3 and section 6.2). For instance, an Adaptation Theme can be used to define that the *boundingBox* adaptation type for beginner users should always be coloured red.

These contributions are summarized as:

Achievement 1: The objective of research question RQ1, related to the possible adaptations that can be applied to 3D VLEs, in order to allow adapting a 3D VLE to the learner's knowledge, skills, and behaviour, was achieved by the definition of the *Adaptation Types* and *Adaptation Strategies*. In addition, the concept of *Adaptation Theme* was introduced to support consistent adaptations.

A Conceptual Modelling Approach

To achieve the aim of integrating adaptive e-learning and 3D virtual environments, a conceptual approach is devised to deal with the different aspects needed for creating adaptive 3D VLEs (discussed in Chapter 5). The proposed approach is considered as model based i.e., each aspect is specified by a separate model. Specifying adaptations inside a 3D VLE in our conceptual modelling approach is broken down into the creation of five models: *Domain Model, Course Model, User Model, Pedagogical Model,* and *Adaptation Model*. In this dissertation, we mainly focused on the Pedagogical Model and the Adaptation Model. The other models are well considered in other research. Each model has its own purpose. For instance, the User Model is mainly used to store/retrieve information about the user, like his preferences, goals, knowledge, interest, background, and skills. The main goal of having a Domain Model is to define the learning concepts covered in the course(s) along with their learning resources. The Course Model is used to define the different topics that are required to be covered during a particular course for the given domain model. More discussion is provided in section 3.2.2.2. In our approach, the User Model is divided to two sub models: the *User Profile* and the *3D Activity Model*. The User Profile is used to store demographic data such as name, age, and gender. The 3D Activity Model is used to maintain information related to user activities inside the 3D VLE. 3D activities, like user interactions with 3D objects, user positions, and behaviours executed should be monitored and interpreted to drive the adaptations.

The *Pedagogical Model* is responsible for defining the pedagogical relationships between the different learning concepts in a course. Such relationships can influence the possible sequences between the learning concepts in a course.

The *Adaptation Model* is used to instruct *what* should be adapted inside the 3D VLE, *when* and *how*. This is done by means of adaptation rules along with adaptation types and strategies. We also proposed adaptation to happen on storyline aspects as well as adaptation to the 3D VLE contents. An adaptive storyline was provided so that different (sequences of) topics can be provided to different learners depending on specific adaptation rules. As a result, an Adaptive Storyline Model and an Adaptive Topic Model are conceived to give the authors (educators) the possibility to specify adaptive at two levels, i.e. at the level of the storyline flow and for each single topic.

This conceptual modelling approach answers the research question **RQ2.1** (What needs to be specified during authoring to allow for an adaptive 3D VLE?) and results in:

<u>Achievement 2:</u> The objective of research question RQ2.1, related to what needs to be specified during authoring to allow for an adaptive 3D VLE, was achieved by the construction of *a conceptual framework* defining the different models needed to specify and drive adaptation in 3D VLEs.

The Visual Modelling Languages

Visual modelling languages are proposed to allow authors to define the *Pedagogical Model*, the *Adaptive Storyline Model* and the *Adaptive Topic Model* (described in Chapter 6). In other words, to support educators, high-level abstractions have been proposed to author adaptive 3D VLEs, i.e. visual modelling languages. The proposed modelling languages are: the *Pedagogical Model Language* (PML), the *Adaptive Storyline Language* (ASLL), and the *Adaptive Topic Language* (ATL). Each visual language uses a number of high level modelling concepts and different visual notations. The visual languages are meant to be simple and intuitive. The visual languages are defined as follow:

- *Pedagogical Model Language* is used to define the pedagogical relationships between learning concepts. It does not only allow defining the pedagogical relationships of the learning concepts, but it also allows defining the way in which the user model should be updated. Visual notations are used to represent learning concepts and pedagogical relationship type.
- *Adaptive Storyline language* is used to define the storyline to be followed by the learners. The Adaptive Storyline language includes two components: topics and storyline adaptation rules. Every topic in the overall storyline includes a number of learning concepts. The flow of the storyline is defined by the educators by means of storyline adaptation rules, which are composed of a condition and an adaptation strategy. Visual notations are used to represent topics and storyline adaptation rules.
- *Adaptive Topic language*, which is used to define the adaptation flow inside a single topic. The visual language is composed of notations for learning concepts (which populate a topic), VR events (which trigger specified adaptations), conditions (which is used to define when the adaptation action should be applied), and adaptation types (which specifies the action part of the composed adaptation rule).

These languages result in:

<u>Achievement 3:</u> The objective of research question RQ2.3, related to how to support 3D novice educators in specifying adaptive 3D VLE, was achieved by the development of 3 visual languages, that together allow specifying adaptive 3D VLEs.

Demonstrating of Authoring Adaptive 3D VLE using the Visual Languages

To demonstrate the capabilities of the visual languages for creating an adaptive 3D VLE an example 3D VLE course (about the solar system) has been elaborated (presented in Chapter 7).

Evaluation of the Visual Languages

To validate the visual modelling languages, two evaluations have been done (presented in Chapter 8). In the first evaluation, we were looking for valuable usability feedback about the proposed modelling languages. Therefore, we asked researchers and PhD students from two universities to be involved in the evaluation process. All participants were from the computer science department. The overall evaluation results showed that the approach is promising and the participants were able to perform the required authoring tasks easily and effectively using the proposed visual modelling languages. In the second evaluation, we were seeking for more insight in problem areas indicated in the first evaluation and for feedback from users who are not computer scientists. The overall evaluation results of the second evaluation were consistent with the previous results. Furthermore, the results showed an appreciation for the authoring approach and the proposed visual languages. The qualitative feedback in the two conducted evaluations provided some important recommendations.

Prototype of Adaptive 3D VLE Authoring Tool

Based on the conceptual model framework, an authoring tool architecture is proposed for allowing authors who do not have a deep background in 3D/VR programming languages, to design a 3D VLE (presented in Chapter 9). For this purpose, we first discussed the requirements for such a tool (section 9.1). For instance, the author should be able to define the pedagogical relationships between the learning concepts of the domain model. Furthermore, the author should be able to define how the 3D resources are adapted inside the 3D VLE. Next, we provided prototypes of the graphical user interface of the authoring tool.

The conceptual framework, and the proposed authoring tool, answer research questions **RQ2.2** , resulting in:

<u>Achievement 4:</u> The objective of research question RQ2.2, related to what is an approach for authoring adaptive 3D VLE usable for 3D novice educators, was achieved by defining a prototype of the authoring tool based on the conceptual framework defined.

Together achievements 2, 3 and 4 answer research question RQ2: How to facilitate the authoring process of adaptive 3D VLE for 3D novice educators?

10.2 Contributions

This dissertation contributes to the adaptive 3D Virtual Learning Environment domain by introducing a novel authoring approach. As mentioned earlier, the domain of adaptive 3D VLE is important since adaptation is considered as one of the key solutions for improving learning outcomes and at the same time overcome usability issues in 3D VLE. Furthermore involving educators in the design of adaptive 3D VLEs is essential for their breakthrough, but defining dynamic adaptations inside 3D VLE by 3D novice educators is still a difficult task. The research work presented in this dissertation tried to provide solutions for this last issue. The following summarize the major contributions of this research work:

- One of the contributions of this research work is *the use of intuitive high-level specifications for describing adaptation in 3D VLE*. Due to the use of intuitive modelling concepts, there is a strong similarity between how one describes adaptation in 3D VLE in our approach and how it would be done using natural language. Therefore, the authors (educators) do not need deep knowledge and background in 3D/Virtual Reality modelling technologies to model an adaptive 3D VLE.

- An important step towards engaging 3D novice educators in the design of adaptation inside the 3D VLE is to support them with *predefined adaptation techniques with a high level of abstraction*. As mentioned in the related work, adaptation is often realised in 3D VLE using dedicated and ad-hoc implementation techniques and are limited to certain aspects of the 3D VLE like adapting 3D object representation or drawing arrows to support learners navigation Our work contributes to the state of the art by providing a list of adaptation types and strategies that can be used by 3D novice educators. The availability of the adaptation types and strategies was highly appreciated in the conducted evaluations. To the same end, a related contribution is the concept of adaptation theme, which reduces the time needed for specifying adaptations and enhances the consistency of the adaptations.

- An interesting finding and outcome during this research work is the need for a learning path that the learners need to follow. Although this aspect is already investigated thoroughly in the domain of serious games, it was not yet explored in the context of adaptive 3D VLE. Therefore, our authoring approach supports *the specification of a storyline.* In addition, it enables an *adaptive storyline for the 3D VLE,* and facilitates the specification by decomposing the adaptation specification into two levels (storyline level and content level). Therefore, adaptation can be realized not only for the 3D components of the 3D VLE but also for the learning path.

- Form the conducted evaluations, we can conclude that the use of intuitive visual languages can *decreased the gap between the educators and the development process of adaptive 3D VLE*, at least for people with some computer background.
- Our last contribution is *a prototype for the GUI of an authoring tool to support our approach*.

10.3 Limitations and Future Work

In this section, we present the limitations of our work. In addition, various interesting directions for future research are discussed.

As most research, this research also has its constraints and limitations. We explain them further.

The proposed visual languages have been applied to a simple example to showcase its advantages and capabilities. Further demonstrations should be done in the context of more complex adaptive 3D VLEs.

It is important to mention that additional evaluations should be performed since the evaluations performed were limited in several respects. For instance, the evaluations were performed without using an authoring tool. An evaluation using a software tool, which implements the proposed functionalities, could reveal other interesting issues. It should also be investigated if the positive evaluation obtained is confirmed when using a tool.

Moreover, only a limited number of participants were involved in the evaluations. Conducting an evaluation with a small number of users (fourteen participants, respectively four in our case) may affect the validity of the result of the evaluation. In addition, all participants in the first evaluation had a computer science background. Additional evaluations should also consider participants with different backgrounds, like experts in VR and non-VR experts, and people with and without modelling experience (as already done in very limited way in the second evaluation). Another important aspect that needs to be considered in future evaluations is the time needed for completing the authoring tasks by participant with different backgrounds. Performing such experiments will validate if the approach is indeed acceptable and fulfils the expectations of a wide range of educators.

Another important limitation is that our approach is focussing on narrative-driven adaptive 3D VLEs. This may exclude other types of adaptive 3D virtual learning environments where a storyline is less appropriate.

The research presented in this thesis can be extended in several directions.

One research direction is related to extending the different adaptation typed and strategies. The proposed adaptation types and strategies were not specified for all 3D VE components; sound and communications are not included. Investigating additional adaptation techniques for audio files will increase the different adaptation possibilities. For example, we could think about delivering sound effects inside the 3D VLE adaptively depending on the speed of the Internet connection. Furthermore, adaptation of audio or video can also happen according to the user model. For instance, beginner users can be delivered audio explanation, as well as video fitting their knowledge level.

Applying adaptation techniques in a collaborative environment is another direction that still needs further investigation. For instance, some adaptation techniques can be used to provide effective audio and text chatting functionalities between students inside a collaborative 3D VLE. However, some other adaptation techniques must be applied with care, as different adaptations for different users may be confusing when users are collaborating on a common task.

Another direction to investigate is to take over some of the work of the authors by providing some kind of intelligence adaptation mechanism. If enough experience is gained about how to effectively adapt a 3D VLE in specific situations, this can be done automatically and there is no need anymore to ask an educator to specify this manually and case by case.

Research can be undertaken regarding supporting learners with 3D VLEs that are adapted to their cultural norms and aspects. Furthermore, a study can be performed to investigate the cultural dimensions that do or do not have an impact on the adoption of an adaptive 3D VLE.

Investigating the usability of delivery environments is also a possible research direction, which is considered as complementary to the authoring. For instance, it would be good to investigate different possibilities to support a learner in the delivery environment. For example, a storyline bar can be displayed to the learner so that he will be able to know in which topic he is, how many topics are completed, and how many topics still need to be learned. Furthermore, it would be interesting to investigate how to achieve an effective and consistent integration between a hypertext part and the 3D part of the delivery environment.

Finally, developing a framework for evaluating the different usability aspects for authoring tools, as well as the delivery environments can be the subject of further research. Such framework can also be used to provide a set of key usability issues and guidelines that should be considered when developing authoring tools and delivery environments in the context of adaptive 3D VLE.

References

3Dxplorer® (2002). 3Dxplorer®. [Online]. 2002. Available from: http://www.3dxplorer.com/. [Accessed: 20 April 2013].

Albert, D., Nussbaumer, A., Steiner, C. M., Hendrix, M. and Cristea, A.I. (2009) Design and development of an authoring tool for pedagogical relationship types between concepts. In: 17th International Conference on Computer in Education (ICCE 2009), Hong Kong, 30, Nov - 4, Dec 2009 pp. 194-196.

Amory, A., Naicker, K., Vincent, J. & Adams, C. (1999). The use of computer games as an educational tool: identification of appropriate game types and game elements. British Journal of Educational Technology. 30 (4). pp. 311–321.

Amory, A. & Seagram, R. (2003). Educational Game Models : Conceptualization and Evaluation Educational Game Models : Conceptualization and Evaluation. Journal of Higher Education. 17 (2). pp. 206 – 217.

Annetta, L.A. (2010). The "I's" have it: A framework for serious educational game design. Review of General Psychology. 14 (2). pp. 105–112.

Aquino, M.S., De Souza, F.F. & Frery, A.C. (2005). VEPersonal: an infrastructure of Virtual Reality components to generate web adaptive environments. In: Proceedings of the 11th Brazilian Symposium on Multimedia and the web. 2005, ACM, pp. 1–8.

Aroyo, L., Denaux, R., Dimitrova, V. & Pye, M. (2006a). Interactive Ontology-Based User Knowledge Acquisition: A Case Study. In: Y. Sure & J. Domingue (eds.). The Semantic Web Research and Applications. Springer Berlin / Heidelberg, pp. 560–574.

Aroyo, L., Dolog, P., Houben, G.-J., Kravcik, M., Naeve, A., Nilsson, M. & Wild, F. (2006b). Interoperability in Personalized Adaptive Learning. Educational Technology & Society. 9 (2). pp. 4–18.

Barbieri, T. & Paolini, P. (2001). Reconstructing Leonardo's ideal city - from handwritten codexes to webtalk-II. In: Proceedings of the conference on Virtual reality archeology and cultural heritage VAST 01. 2001, ACM Press, pp. 61–66.

Bares, W.H. & Lester, J.C. (1999) Intelligent multi-shot visualization interfaces for dynamic 3D worlds. In Proceedings of the 1999 International Conference on Intelligent User Interfaces.

Bazargan, K., Falquet, G. & Métral, C. (2010). Testing the Usability of Information Visualization Techniques in Interactive 3D Virtual Environments: a Toolkit. In: Proceedings of IADIS International Conference Interfaces and Human Computer Interaction. 2010, Freiburg, Germany: IADIS IHCI'2010, pp. 11–18.

Beck, J.E., Woolf, B.P. & Beal, C.R. (2000). ADVISOR : A machine learning architecture for intelligent tutor construction. Proceedings of the National Conference on Artificial Intelligence. pp. 552–557.

Begg, M., Dewhurst, D. & Macleod, H. (2005). Game-Informed Learning : Applying Computer Game Processes to Higher Education. Innovate: Journal of Online Education. 1(6).

Bobkowska , A. (2003). Cognitive Dimensions questionnaire applied to visual languages evaluation -- a case study. In Proceedings of the First Joint Conference of EASE & PPIG (PPIG 15).

Bellotti, F., Berta, R., De Gloria, A. & Primavera, L. (2010). Supporting authors in the development of task-based learning in serious virtual worlds. British Journal of Educational Technology. 41 (1). pp. 86–107.

Bernie Roehl (1995). Some Thoughts on Behavior in VR Systems. 1995.

Bilasco, I.M., Villanova-Oliver, M., Gensel, J. & Martin, H. (2007). Semantic-based rules for 3D scene adaptation. In: International Conference on 3D Web Technology. 2007, ACM, pp. 97–100.

Bille, W., Pellens, B., Kleinermann, F. & De Troyer, O. (2004). Intelligent modelling of virtual worlds using domain ontologies. Proceedings of the IVEVA 2004 Workshop. pp. 272–279.

Blackwell, A. F., Britton, C., Cox, A., Dautenhahn, K., Green, T. R. G., Gurr, C., Jones, S., Kadoda, G., Kutar, M. S., Loomes, M., Nehaniv, C. L., Petre, M., Roast, C., Roe, C., Russ, S., A., W., & Young, R. M., (2001). Cognitive Dimensions of Notations: Design Tools for Cognitive Technology. In M. Benyon & C. L. Nehaniv & K. Dautenhahn (Eds.), Springer-Verlag, pp. 325-341.

Blackwell, A., & Green, T., (2000). A cognitive dimensions Questionnaire Optimized for Users, 12th Workshop of the Psychology of Programming Interest Group, Italy.

Blanco, Á. del, Torrente, J., Moreno-Ger, P. & Fernández-Manjón, B. (2009). Bridging the Gap: Adaptive Games and Student-Centered VLEs. In: Proceedings of the International Conference on Web-based Learning (ICWL 2009). 2009, Aachen, Germany: LNCS 5686,Springer, pp. 130–139.

Bloom, B.S. (1956). Taxonomy of Educational Objectives: The Classification of Educational Goals. B. S. Bloom (ed.). Longman Group.

Bonis, B., Stamos, J., Vosinakis, S., Andreou, I. & Panayiotopoulos, T. (2007). Personalization of Content in Virtual Exhibitions. In: P. B. Bianca Falcidieno, Michela Spagnuolo, Yannis Avrithis, Ioannis Kompatsiaris (ed.). Semantic Multimedia. Springer Berlin Heidelberg, pp. 172–184.

Bopp, M.M. (2008). Storytelling as Motivation In Serious Games. *Part of the Final Consolidated Research Report of the Enhanced Learning Experience and Knowledge Transfer–Project (ELEKTRA, http://www. elektra-project. org/), a project supported by the European Commission (contract number 027986)*.

Boshernitsan, M. & Downes, M. (2004). Visual programming languages: A survey. Computer Science Division, University of California.

Bouras, C., Hornig, G., Triantafillou, V. & Tsiatsos, T. (2001a). Architectures supporting e-learning through collaborative virtual environments: the case of INVITE. Proceedings IEEE International Conference on Advanced Learning Technologies. pp. 13–16.

Bouras, C., Philopoulos, A. & Tsiatsos, T. (2001b). e-Learning through distributed virtual environments. Network and Computer Applications. 24 (3). pp. 175–199.

Bourgonjon, J., Valcke, M., Soetaert, R. & Schellens, T. (2010). Students' perceptions about the use of video games in the classroom. Computers & Education. 54 (4). pp. 1145–1156.

Bowman, D.A., Gabbard, J.L. & Hix, D. (2002). A Survey of Usability Evaluation in Virtual Environments: Classification and Comparison of Methods. Presence Teleoperators Virtual Environments. 11 (4). pp. 404–424.

Bowman, D.A. & Hodges, L.F. (1999). Formalizing the Design, Evaluation, and Application of Interaction Techniques for Immersive Virtual Environments. Visual Languages Computing. 10 (1). pp. 37–53.

Boyle, C. & Encarnacion, A.O. (1994). Metadoc: An adaptive hypertext reading system. User Modeling and UserAdapted Interaction. 4 (1). pp. 1–19.

Bricken, M. & Byrne, C.M. (1992). Summer students in virtual reality: A pilot study on educational applications of virtual reality technology. A. Wexelblat (ed.). Cambridge: Human Interface Technology Laboratory of the Washington Technology Center, University of Washington.

Brodlie, K. (2000). Using web-based computer graphics to teach surgery. Computers & Graphics. 24 (1). pp. 157–161.

Bronack, S., Sanders, RL., Cheney, A., Riedl, R., Tashner, J., & Matzen, N. (2008). Presence Pedagogy: Teaching and Learning in a 3D Virtual Immersive World. International Journal of Teaching and Learning in Higher Education, 20(1), pp. 59-69.

Bruegge, B. & Dutoit, A.H. (2000). Object-oriented software engineering. ACM SIGSOFT Software Engineering Notes. 25 (3), pp. 65.

Brusilovsky, P. (2004). Adaptive Navigation Support: From Adaptive Hypermedia to the Adaptive Web and Beyond. PsychNology Journal. 2 (1). pp. 7–23.

Brusilovsky, P. (2003). Developing adaptive educational hypermedia systems: From design models to authoring tools T. Murray, S. Blessing, & S. Ainsworth (eds.). Information Sciences. pp. 377–409.

Brusilovsky, P. (1996). Methods and techniques of adaptive hypermedia. User Modeling and UserAdapted Interaction. 6 (2-3). pp. 87–129.

Brusilovsky, P., Eklund, J. & Schwarz, E. (1998a). Web-Based Education for All: A Tool for Developing Adaptive Courseware. In: H. Ashman & P. Thistewaite (eds.). Computer Networks and ISDN Systems. 1998, Amsterdam, The Netherlands,: Elsevier Science B. V., pp. 291–300.

Brusilovsky, P., Kobsa, A. & Vassileva, J. (1998b). Adaptive Hypertext and Hypermedia. P. Brusilovsky, A. Kobsa, & J. Vassileva (eds.). Kluwer Academic Publishers.

Brusilovsky, P., Sosnovsky, S. & Yudelson, M. V (2005). Ontology-based Framework for User Model Interoperability in Distributed Learning Environments. In: G. Richards (ed.). World Conference on ELearning, E-Learn 2005. 2005, AACE, pp. 2851–2855.

Brutzman, D. (1998). The virtual reality modeling language and Java. Communications of the ACM. 41 (6). pp. 57–64.

Buche, C. & Querrec, R. (2011). An expert system manipulating knowledge to help human learners into virtual environment. Expert Systems with Applications. 38 (7). pp. 8446–8457.

Bulterman, D.C.A., Rutledge, L., Hardman, L., Jansen, J. & Mullender, K.S. (1999). GRiNS: an authoring environment for web multimedia. In: World Conference on Educational Multimedia, Hypermedia and Educational Telecommunications, ED-MEDIA 99. 1999.

Bunt, A., Carenini, G. & Conati, C. (2007). Adaptive Content Presentation for the Web P. Brusilovsky, A. Kobsa, & W. Nejdl (eds.). The Adaptive Web Methods and Strategies of Web Personalization. 4321. pp. 409 – 432.

Burdea, G. (2006). Virtual Reality Technology - An Introduction. In: IEEE Virtual Reality Conference VR 2006. 2006, Ieee, pp. 307–307.

Burigat, S. & Chittaro, L. (2007). Navigation in 3D virtual environments: Effects of user experience and location-pointing navigation aids. International Journal of Human-Computer Studies. 65 (11). pp. 945–958.

Cai, Y., Miao, C. & Shen, Z. (2009). Building Interactive Teaching and Learning Environment For High Education in 3D Virtual Worlds. Science Education. pp. 2402–2408.

Campbell, C. & Cameron, L. (2009). Using Learning Activity Management Systems (LAMS) with pre-service secondary teachers: An authentic task. In: ASCILITE 2009, 26th Conference (ERA 2010). 2009, Auckland, New Zealand: Auckland University of Technology, and Australasian Society for Computers in Learning in Tertiary Education (ascilite), pp. 96–103.

Carreras, M.A.M. & Skarmeta, A.F.G. (2006). Towards Interoperability in Collaborative Environments. In: International Conference on Collaborative Computing: Networking, Applications and Worksharing. 2006, Atlanta, GA, USA: IEEE, pp. 1–5.

Carro, R.M., Breda, A.M., Castillo, G. & Bajuelos, A.L. (2002). A Methodology for Developing Adaptive Educational-Game Environments. Environment. 2347. pp. 90–99.

Castelli, L., Latinicorazzini, L. & Geminiani, G. (2008). Spatial navigation in large-scale virtual environments: Gender differences in survey tasks. Computers in Human Behavior. 24 (4). pp. 1643–1667.

Celentano, A. & Nodari, M. (2004). Adaptive interaction in Web3D virtual worlds. Proceedings of the ninth international conference on 3D Web technology Web3D 04. 1 (212). pp. 41.

Celentano, A. & Pittarello, F. (2004). Observing and adapting user behavior in navigational 3D interfaces. Proceedings of the working conference on Advanced visual interfaces AVI 04. pp. 275.

Charles, D., McNeill, M., McAlister, M., Black, M., Moore, A., Stringer, K., Kücklich, J. & Kerr, A. (2005). Player-Centred Game Design: Player Modelling and Adaptive Digital Games. In: DiGRA (ed.). Proceedings of the Digital Games Research Conference. 2005 pp. 285–298.

Chen, C.H., Yang, J.C., Shen, S. & Jeng, M.C. (2007). A Desktop Virtual Reality Earth Motion System in Astronomy Education. Educational Technology & Society. 10 (3). pp. 289–304.

Chen, P. (1976). The Entity-Relationship Model - Towards a Unified View of Data. ACM Transactions on Database Systems. 1 (1). pp. 9–36.

Chittaro, L. & Coppola, P. (2000). Animated products as a navigation aid for e-commerce. In: CHI 00 extended abstracts on Human factors in computing systems CHI 00. 2000, ACM Press, p. 107.

Chittaro, L., Ieronutti, L. & Ranon, R. (2004). Navigating 3D Virtual Environments by Following Embodied Agents: a Proposal and its Informal Evaluation on a Virtual Museum Application. PsychNology Journal. 2 (1). pp. 24–42.

Chittaro, L. & Ranon, R. (2007a). Adaptive 3D Web Sites P. Brusilovsky, A. Kobsa, & W. Nejdl (eds.). Environments. pp. 433 – 462.

Chittaro, L. & Ranon, R. (2007b). Adaptive Hypermedia Techniques for 3D Educational Virtual Environments. IEEE Intelligent Systems (SpecialIssue on Intelligent Educational Systems). 22. pp. 31–37.

Chittaro, L. & Ranon, R. (2000). Adding Adaptive Features to Virtual Reality Interfaces for E-Commerce. In: Proceedings of the International Conference on Adaptive Hypermedia and Adaptive Webbased Systems. 2000, Springer, pp. 701–705.

Chittaro, L. & Ranon, R. (2008). An adaptive 3D virtual environment for learning the X3D language. Proceedings of the 13th international conference on Intelligent user interfaces IUI 08. pp. 419.

Chittaro, L. & Ranon, R. (2002a). Dynamic generation of personalized VRML content: a general approach and its application to 3D e-commerce. Information Systems Journal. pp. 145–154.

Chittaro, L. & Ranon, R. (2002b). New directions for the design of virtual reality interfaces to e-commerce sites. Proceedings of the Working Conference on Advanced Visual Interfaces AVI 02. pp. 308.

Chittaro, L. & Ranon, R. (2004). Using the X3D Language for Adaptive Manipulation of 3D Web Content. W. Bra, PaulM.E. and Nejdl (ed.). Springer Berlin Heidelberg.

Chittaro, L. & Ranon, R. (2007c). Web3D technologies in learning, education and training. Computers & Education. 49 (1). pp. 1–2.

Chittaro, L. & Ranon, R. (2007d). Web3D technologies in learning, education and training: Motivations, issues, opportunities. Computers & Education. 49 (1). pp. 3–18.

Ciampi, M., Gallo, L., Coronato, A. & De Pietro, G. (2009). Middleware mechanisms for interaction interoperability in Collaborative Virtual Environments. International Journal of Advanced Media and Communication. 4 (2). pp. 154–166.

Classroom® (2006). Classroom®. [Online]. 2006. Available from: https://cloud.saba.com/classroom/. [Accessed: 20 April 2013].

Conati, C. & Manske, M. (2009). Evaluating Adaptive Feedback in an Educational Computer Game Z. Ruttkay, M. Kipp, A. Nijholt, & H. H. Vilhjálmsson (eds.). Intelligent Virtual Agents. 5773. pp. 146–158.

Conlan, O. (2003). State of the Art: Adaptive Hypermedia. MZones Research Programme State of Art Surveys Release. 2 (May). pp. 47–57.

Conlan, O., Wade, V., Bruen, C. & Gargan, M. (2002). Multi-model, Metadata Driven Approach to Adaptive Hypermedia Services for Personalized eLearning. In: P. M. E. Bra & W. Nejdl (eds.). Adaptive Hypermedia and Adaptive WebBased Systems. 2002, Springer Berlin / Heidelberg, pp. 100–111.

Cristea, A. (2004). Authoring of adaptive and adaptable educational hypermedia: Where are we now and where are we going? Proceedings of the IASTED International Conference on WebBased Education. pp. 360–365.

Cristea, A. I., & De Mooij, A. (2003a). Adaptive course authoring: My Online Teacher. 10th International Conference on Telecommunications 2003 ICT 2003, 2, pp. 1762–1769.

Cristea, A.I., & De Mooij, A. (2003b). LAOS: Layered WWW AHS Authoring Model and its corresponding Algebraic Operators, World Wide Web conference (WWW), Alternate Education track.

Cristea, A.I., & Calvi, L. (2003c) .The three Layers of Adaptation Granularity. UM'03 (International Conference on User Modelling), Pittsburgh, US, Springer, ISBN: 978-3-540-40381-4, pp. 145.

Cristea, A.I., Smits, D., Bevan, J., Hendrix, M. (2009). "LAG 2.0: Refining a reusable Adaptation Language and Improving on its Authoring," in 4th European Conference on Technology Enhanced Learning (ECTEL'09), Nice, France, pp. 7–21.

Cristea, A.I. & Ghali, F. (2011). Towards adaptation in e-learning 2.0. New Review Of Hypermedia And Multimedia. 17 (2). pp. 199–238.

Cubero, J.M., Salgado, L., Bescós, J., Morán, F. & Cabrera, J. (2004). External Interaction Management of VRML Scenes for E-Learning Applications. In: Workshop on Web3D Technologies in Learning, Education and Training-LET-WEB3D 2004. 2004, pp. 47–52.

Dachselt, R., Hinz, M. & Pietschmann, S. (2006). Using the AMACONT architecture for flexible adaptation of 3D web applications. Proceedings of the Eleventh International Conference on 3D Web Technology Web3D 06. 1 (April). pp. 75–84.

Dade-Robertson, M. (2007). Visual Scenario Representation in the Context of a Tool for Interactive Storytelling M. Cavazza & S. Donikian (eds.). Screen. 4871. pp. 3–12.

Dagger, D., Wade, V.P. & Conlan, O. (2004). Developing Adaptive Pedagogy with the Adaptive Course Construction Toolkit (ACCT). Computers in Education.

Dagger, D., Wade, V., & Conlan, O. (2005). Personalisation for All: Making Adaptive Course Composition Easy. Educational Technology & Society, 8 (3), 9-25.

Dalgarno, B. & Lee, M.J.W. (2010). What are the learning affordances of 3-D virtual environments? British Journal of Educational Technology. 41 (1). pp. 10–32.

Dalziel, J. (2003). Implementing Learning Design: The Learning Activity Management System (LAMS). In: G. Crisp, D. Thiele, I. Scholten, S. Barker, & J. Baron (eds.). Learning. 2003, ASCILITE, pp. 7–10.

Davis, F., 1989. Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. MIS Quarterly 13(3), pp. 319-339.

De Bra, P., Aerts, A., Berden, B., De Lange, B., Rousseau, B., Santic, T., Smits, D. & Stash, N. (2003). AHA! The adaptive hypermedia architecture. Proceedings of the fourteenth ACM conference on Hypertext and hypermedia HYPERTEXT 03. 4 (1). pp. 81–84.

De Bra, P., Aerts, A., Rousseau, B., Santic,. (2002). Concept Relationship Types for AHA! 2.0. In M. Driscoll & T. Reeves (Eds.), Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2002. pp. 1386-1389.

De Bra, P., Smits, D., Stash, N., (2006). Creating and Delivering Adaptive Courses with AHA!, Proceedings of the first European Conference on Technology Enhanced Learning, EC-TEL 2006, Springer LNCS 4227, pp. 21-33.

De Bra, P. & Calvi, L. (1998). AHA! a generic Adaptive Hypermedia System. In: P. Brusilovski & P. De Bra (eds.). Proceeding 2nd Workshop on Adaptive Hypertext and Hypermedia. 1998, Pittsburgh, USA, pp. 1–10.

De Bra, P., Smits, D., Van Der Sluijs, K., Cristea, A. & Hendrix, M. (2010). GRAPPLE: Personalization and Adaptation in Learning Management

Systems. In: Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2010. 2010, Association for the Advancement of Computing in Education (AACE), pp. 3029–3038.

De Byl, P. (2009). Designing Games-Based Embedded Authentic Learning Experiences. In: Effective GamesBased Learning Design. IGI Press, pp. 1–24.

De Byl, P. & Taylor, J. (2007). A Web 2.0/Web3D hybrid platform for engaging students in e-learning environments. Turkish Online Journal of Distance Education. 8 (3). pp. 108–127.

De Troyer, O., Kleinermann, F. & Ewais, A. (2010). Enhancing Virtual Reality Learning Environments with Adaptivity : Lessons Learned. In: The 6th international conference on HCI in work and learning, life and leisure: workgroup human-computer interaction and usability engineering. 2010, Klagenfurt, Austria: Springer-Verlag, pp. 244–265.

De Troyer, O., Kleinermann, F., Pellens, B. & Ewais, A. (2009). Supporting virtual reality in an adaptive web-based learning environment. In: Proceedings of the 4th European Conference on Technology Enhanced Learning Learning in the Synergy of Multiple Disciplines. 2009, Nice, France: Springer-Verlag, pp. 627–632.

Dede, C., Ketelhut, D. & Ruess, K. (2003). Designing for Motivation and Usability in a Museum-based Multi-User Virtual Environment. Retrieved November. 1. pp. 2005.

Deol, K.K., Sutcliffe, A. & Maiden, N. (1999). A design advice tool presenting usability guidance for virtual environments. Geobiology. pp. 15–21.

Derryberry, A. (2010). Serious games: online games for learning. pp. 1–15.

Di Blas, N., Hazan, S., Bearman, D. & Trant, J. (2003). The SEE Experience: Edutainment in 3D Virtual Worlds. In: Museums and the Web. 2003, Archives & Museum Informatics, pp. 173–182.

Dickey, M.D. (2006). Game Design Narrative for Learning: Appropriating Adventure Game Design Narrative Devices and Techniques for the Design of Interactive Learning Environments. Educational Technology Research & Development. 54 (3). pp. 245–263.

Dijk, B. Van, Akker, R. Den, Nijholt, A. & Zwiers, J. (2003). Navigation Assistance in Virtual Worlds. Science Journal. 6 (June). pp. 115–125.

Dillon, T. & Tan, P.L. (1993). Object-Oriented Conceptual Modeling. Prentice Hall, PTR Upper Saddle River, NJ, USA

Dolog, P. & Nejdl, W. (2003). Challenges and Benefits of the Semantic Web for User Modelling P. De Bra (ed.). Proceedings of Adaptive Hypermedia. 3. pp. 1–9.

Dolog, P. & Schäfer, M. (2001). Learner Modeling on the Semantic Web. *Proc. of PerSWeb*, *5*.

Dondlinger, M.J. (2007). Educational Video Game Design : A Review of the Literature. Applied Educational Technology. 4 (1). pp. 21–31.

Dong, W., Li, Z., Yan, J., Wu, X., Yang, Y. & Zheng, L. (2008). Adaptive Interaction in a 3D Product Structure Browsing System for Maintenance Training. Hum. Factor. Ergon. Manuf. 18 (1). pp. 14–29.

Dormann, C. & Biddle, R. (2008). Understanding game design for affective learning. Proceedings of the 2008 Conference on Future Play Research Play Share Future Play 08. pp. 41.

Dörr, D.K., Inform, D., Schiefele, J. & Kubbat, W. (2000). Virtual Cockpit Simulation for Pilot Training. RTO MP058. (April). pp. 11–1–11–7.

Dos Santos, C.T. & Osório, F.S. (2004a). AdapTIVE: An Intelligent Virtual Environment and Its Application in E-Commerce. In: Proceedings of 28th Annual International Computer Software and Applications Conference (COMPSAC'04). 2004, pp. 468–473.

Dos Santos, C.T. & Osório, F.S. (2004b). An intelligent and adaptive virtual environment and its application in distance learning. Proceedings of the working conference on Advanced visual interfaces AVI 04. pp. 362.

Drucker, S.M. & Zeltzer, D. (1994). Intelligent Camera Control in a Virtual Environment. In Proceedings of Graphics Interface '94. Banff, Alberta, Canada.

Durkin, K. (2010). Videogames and young people with developmental disorders. Review of General Psychology. 14 (2). pp. 122–140.

Dyson, M. & Campello, S.B. (2003). Evaluating virtual learning environments: what are we measuring? Electronic Journal of e-Learning,. 1 (1). pp. 11–20.

Eck, R. Van (2006). Digital Game-Based Learning : It ' s Not Just the Digital Natives Who Are Restless …. Educause Review. 41 (2). pp. 1–16.

Engine®, T.G. (2001). Torque Game Engine®. [Online]. 2001. Available from: http://www.garagegames.com/. [Accessed: 20 April 2013].

Enochsson, L., Isaksson, B., Tour, R., Kjellin, A., Hedman, L., Wredmark, T. & Tsai-Fellânder, L. (2004). Visuospatial skills and computer game experience influence the performance of virtual endoscopy. Journal of gastrointestinal surgery official journal of the Society for Surgery of the Alimentary Tract. 8 (7). pp. 876–882; discussion 882.

Estalayo, E., Salgado, L., Morán, F., Cabrera, J. & Imágenes, G.D.T. De (2004). Adapting Multimedia Information Association in VRML Scenes for E-Learning Applications. In: the First international Workshop LET-Web3D. 2004, pp. 16–22.

Ewais, A. & De Troyer, O. (2014). Authoring Adaptive 3D Virtual Learning Environments. International Journal of Virtual and Personal Learning Environments (IJVPLE). 5 (1).

Ewais, A. & De Troyer, O. (2013a). Authoring Storyline-based Adaptive 3D Virtual Learning Environments. In: Proceedings of the 5th International Conference on Computer Supported Education (CSEDU 2013). 2013, Aachen, Germany.

Ewais, A. & De Troyer, O. (2013b). Usability Evaluation of an Adaptive 3D Virtual Learning Environment. International Journal of Virtual and Personal Learning Environments (IJVPLE). 4 (1).

ExperienceBuilder (2000). www.experiencebuilder. com .[Accessed: 20 April 2013].

Ferguson, C.J., Cruz, A.M. & Rueda, S.M. (2007). Gender, Video Game Playing Habits and Visual Memory Tasks. Sex Roles. 58 (3-4). pp. 279–286.

Figl, K., Derntl, M., Rodriguez, M.C. & Botturi, L. (2010). Cognitive effectiveness of visual instructional design languages. Journal of Visual Languages Computing. 21 (6). pp. 359–373.

Figl, K., Mendling, J., Strembeck, M. (2009). Towards a Usability Assessment of Process Modeling Languages, In: Proc. of the 8. GI-Workshop EPK: Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten, Berlin, Germany. pp. 118-136.

Foss, J.G.K., Cristea, A.I., Adaptive Hypermedia Content Authoring using MOT3.0, Authoring of Adaptive and Adaptable Hypermedia Workshop (A3H), Workshop at European Conference on Technology-Enhanced Learning(ECTEL), 2009.

Fowler, M. & Scott, K. (2000). UML distilled: a brief guide to the standard object modeling language. Addison-Wesley.

Frasincar, F. & Houben, G.J. (2002). Hypermedia Presentation Adaptation on the Semantic Web. Lecture Notes in Computer Science. pp. 133–142.

Frery, A.C., Kelner, J., Moreira, J. & Teichrieb, V. (2002). User satisfaction through empathy and orientation in three-dimensional worlds. Cyberpsychology and Behavior. 5 (5). pp. 451–459.

Funge, J. (2000). Cognitive modeling for games and animation. Communications of the ACM. 43 (7). pp. 40–48.

Gaddis, T. (1998). Virtual reality in the school. Virtual reality and Education Laboratory. East Carolina University.

Gaffney, C., Dagger, D. & Wade, V. (2010). Authoring and Delivering Personalised Simulations - an Innovative Approach to Adaptive eLearning for Soft Skills. J.UCS. 16 (19). pp. 2780–2800.

GameStudio® (2010). GameStudio®. [Online]. 2010. Available from: http://www.3dgamestudio.com/. [Accessed: 20 April 2013].

Garrett, J.J. (2005). Ajax: A New Approach to Web Applications. 2005. [EB/OL]

Garshol, L.M. (2003). BNF and EBNF: What are they and how do they work? *acedida pela última vez em*, *16*.

Gee, J.P. (2007). *Good Video Games Plus Good Learning* (Vol. 27). Peter Lang.

.Georgouli, K. (2011). Virtual Learning Environments - An Overview. In: 2011 15th Panhellenic Conference on Informatics. 2011, IEEE, pp. 63–67.

Ghali, F., Cristea, A. I., Stewart, C. (2008). My Online Teacher 2.0. 3rd European Conference on Technology Enhanced Learning (EC-TEL 2008), IGACLE workshop, Maastricht, Netherlands.

Göbel, S., Konrad, R., Salvatore, L., Sauer, S. & Osswald, K. (2007). U-CREATE: Authoring Tool for the Creation of Interactive Storytelling Based Edutainment Applications. In: Eva 2007 Florence Electronic Imaging and the Visual Arts Conference Training and Exhibition. 2007, Pitagora Editrice, pp. 53–58.

Göbel, S., Mehm, F., Radke, S. & Steinmetz, R. (2009). 80days: Adaptive digital storytelling for digital educational games. In: Yiwei Cao Anna Hannemann (ed.). Proceedings of the 2nd International Workshop on StoryTelling and Educational Games STEG'09. 2009, CEUR Workshop Proceedings.

Göbel, S., Salvatore, L. & Konrad, R. (2008). StoryTec: A Digital Storytelling Platform for the Authoring and Experiencing of Interactive and Non-Linear Stories. In: U. Spierling & N. Szilas (eds.). International Conference on Automated Solutions for Cross Media Content and MultiChannel Distribution. 2008, IEEE, pp. 103–110.

Google3DWarehouse (2006). Google 3D Warehouse. [Online]. 2006. Available from: http://sketchup.google.com/3dwarehouse/. [Accessed: 20 April 2013].

GoogleSketchUp (2000). Google SketchUp. [Online]. 2000. Available from: http://www.sketchup.com/3Dfor/higher-education. [Accessed: 20 April 2013].

Greeff, M. & Lalioti, V. (2001). Interactive storytelling with virtual identities. In: 7th Eurographics conference on Virtual Environments. 2001, pp. 121–130.

Green, C.S. & Bavelier, D. (2003). Action video game modifies visual selective attention. Nature Publishing Group.

Green, T.R.G. and Petre, M. (1996). Usability analysis of visual programming environments: a 'cognitive dimensions' approach. Journal of Visual Languages and Computing, 7(2), pp. 131-174.

Gutiérrez, M., Vexo, F. & Thalmann, D. (2008). Stepping into Virtual Reality. 1st Ed. Springer-Verlag TELOS.

Gygi, K. (1990). Recognising the symptoms of hypertext…and what to do about it. The Art of Human Computer Interface Design, pp.279-287.

Halpin, T. & Morgan, T. (2008). Information Modeling and Relational Databases. Morgan Kaufmann.

Hart, S.G. & Staveland, L.E. (1988). Development of NASA-TLX (Task Load Index): Results of experimental and theoretical research. In P. A. Hancock & N. Meshkati (Eds.), Human Mental Workload, pp. 139–183.

Hartman, J. & Wernecke, J. (1998). The VRML 2.0 Handbook. Addison-Wesley Publishing.

Hatem Abdul kader (2011). e-Learning Systems in Virtual Environment. The international Arab Journal of Information Technology. 8 (1). pp. 23–29.

Hauger, D. & Kock, M. (2007). State of the art of adaptivity in e-learning platforms. Proceedings of the 15th Workshop on Adaptivity and User Modeling in Interactive Systems. pp. 355.

Hendrix, M., De Bra, P., Pechenizkiy, M., Smits, D. & Cristea, A. (2008). Defining Adaptation in a Generic Multi Layer Model: CAM: The GRAPPLE Conceptual Adaptation Model P. Dillenbourg & M. Specht (eds.). Times of Convergence Technologies Across Learning Contexts. 5192. pp. 132–143.

Hocine, N. & Gouaïch, A. (2011). A Survey of Agent Programming and Adaptive Serious Games. Research report Hal Lirmm. N RR-11013.

Hodgins, W. & Duval, E. (2002). IEEE Standard for Learning Object Metadata. Institute of Electrical and Electronics Engineers New York. 1484.12.1. (Generic) pp. i–32.

Hodhod, R., Kudenko, D. & Cairns, P. (2009). Serious Games to Teach Ethics. In: proceedings of AISB. 2009 pp. 6–9.

Hoffman, H.G., Patterson, D.R., Magula, J., Carrougher, G.J., Zeltzer, K., Dagadakis, S. & Sharar, S.R. (2004). Virtual reality pain control during burn wound debridement in the hydrotank.

Hossain, M.A., Rahman, M.A., Saddik, A. El & Levy, P. (2004). Architecture for 3D navigation and authoring of distributed learning object repositories. In: Second International Conference on Creating Connecting and Collaborating through Computing. 2004.

Hubal, R. (2008). Embodied Tutors for Interaction Skills Simulation Training. International Journal of Virtual Reality. 7 (1). pp. 1–8.

Hughes, S., Brusilovsky, P. & Lewis, M. (2002). Adaptive Navigation Support In 3-D E-Commerce Activities. In: 2nd International Conference on Adaptive Hypermedia and Adaptive Web Based Systems. 2002, Malaga, Spain, pp. pp. 132–139.

Hulpuş, I., Fradinho, M. & Hayes, C. (2010). On-the-Fly Adaptive Planning for Game-Based Learning. In: I. Bichindaritz & S. Montani (eds.). CaseBased Reasoning Research and Development. Springer Berlin / Heidelberg, pp. 375–389.

Hunicke, R. (2005). The case for dynamic difficulty adjustment in games. In: Proceedings of the 2005 ACM SIGCHI International Conference on Advances in computer entertainment technology. 2005, ACM, pp. 429–433.

Iqbal, T., Hammermüller, K. & Tjoa, A.M. (2010). Second life for illiterates: a 3D virtual world platform for adult basic education. In: The 12th International Conference on Information Integration and Web-based Applications & Services. 2010, pp. 373–380.

ISO/IEC-14772 (1996). The Virtual Reality Modeling Language Specification. 1996.

Jara, C., Candelas, F., Torres, F., Dormido, S., Esquembre, F. & Reinoso, O. (2009). Real-time collaboration of virtual laboratories through the Internet. Computers & Education. 52 (1). pp. 126–140.

Jenkins, M., Browne, T. & Walker, R. 2005. VLE surveys: A longitudinal perspective between March 2001, March 2003 and March 2005 for higher education in the United Kingdom. UCISA.

Chin, J., Diehl, V., & Norman, K.(1988). Development of an instrument measuring user satisfaction of the human-computer interface. In CHI '88 Conference Proceedings: Human Factors in Computing Systems, New York: Association for Computing Machinery, pp.213-218.

Jourdan, M., Roisin, C., Tardif, L. & Villard, L. (1999). Authoring SMIL documents by direct manipulations during presentation. World Wide Web. 2 (4). pp. 179–190.

Kalyuga, S. (2011). Cognitive Load Theory : Implications for Affective Computing. Artificial Intelligence. pp. 105–110.

Kang, Y.K.Y. & Tan, A.-H.T.A.-H. (2010). Learning Personal Agents with Adaptive Player Modeling in Virtual Worlds. In: Web Intelligence and Intelligent Agent Technology WIIAT 2010 IEEEWICACM International Conference on. 2010, IEEE, pp. 173–180.

Kareal, F. & Klema, J. (2006). Adaptivity in e-learning A. Méndez-Vilas, A. Solano, J. Mesa, & J. A. Mesa (eds.). Current. 1. pp. 260–264.

Karsai, G., Krahn, H., Pinkernell, C., Rumpe, B., Schindler, M., and Volkel, S. (2009). Design Guidelines for Domain Specific Languages. In 9th Workshop on Domain-Specific Modeling, Orlando, Florida.

Kato, P.M. (2010). Video Games in Health Care: Closing the Gap. Review of General Psychology. 14 (2). pp. 113–21.

Kato, P.M., Cole, S.W., Bradlyn, A.S. & Pollock, B.H. (2008). A video game improves behavioral outcomes in adolescents and young adults with cancer: a randomized trial. Pediatrics. 122 (2). pp. e305–e317.

Katsionis, G. & Virvou, M. (2004). A virtual reality user interface for learning in 3D environments. In: Workshop on Web3D Technologies in Learning, Education and Training-LET-WEB3D 2004. 2004, pp. 58–62.

Kickmeier-Rust, M.D. & Albert, D. (2007). The ELEKTRA Ontology Model: A Learner-Centered Approach to Resource Description. In: ICWL. 2007, Springer, pp. 78–89.

Kickmeier-rust, M.D., Göbel, S. & Albert, D. (2008). 80Days : Melding Adaptive Educational Technology and Adaptive and Interactive Storytelling in Digital Educational Games. In: First International Workshop on Story-Telling and Educational Games (STEG’08) - The power of narration and imagination in technology enhanced learning. 2008, Maastricht, The Netherlands, pp. 18–19.

Kickmeier-Rust, M.D., Peirce, N., Conlan, O., Schwarz, D., Verpoorten, D. & Albert, D. (2007). Immersive Digital Games: The Interfaces for Next-Generation E-Learning. Universal Access in HumanComputer Interaction Applications and Services. LNCS 4556.

Kiili, K. (2005a). Digital game-based learning: Towards an experiential gaming model. The Internet and Higher Education. 8 (1). pp. 13–24.

Kiili, K. (2005b). Educational Game Design: Experiential gaming model revised. Tampere University, Research Report.

Kiili, K. (2007). Foundation for problem-based gaming. British Journal of Educational Technology. 38 (3). pp. 394–404.

Kipper, K. & Palmer, M. (2000). Representation of actions as an interlingua. In: NAACLANLP 2000 Workshop Applied Interlinguas Practical Applications of Interlingual Approaches to NLP Seattle May 2000 pp. 12-17.

Kirschner, P.A. (2002). Cognitive load theory implications of cognitive load theory design-learning. Learning and Instruction. 12 (1). pp. 1–10.

Kleinermann, F. & De Troyer, O. (2009). Design of an authoring platform for adaptive VR learning material, GRAPPLE deliverable.

Klobucar, Senicar & Blazic (2004). Privacy Issues of a Smart Space for Learning. Advanced Learning Technologies 2004 Proceedings IEEE International Conference on. (SN -). pp. 780–782.

Knöpfle, C. & Jung, Y. (2006). The Virtual Human Platform: Simplifying the Use of Virtual Characters. The International Journal of Virtual Reality. 5 (2). pp. 25–30.

Koidl, K., Mehm, F., Hampson, C., Conlan, O., & Göbel, S. (2010). Dynamically adjusting Digital Educational Games towards Learning Objectives. In Proceedings of the *3rd European Conference on Games Based Learnin*g (ECGBL).

Knutov, E., De Bra, P., Smits, D. & Pechenizkiy, M. (2011). Bridging navigation, search and adaptation : adaptive hypermedia models evolution. In: 7th International Conference on Web Information Systems and Technologies. 2011, SciTePress, pp. 314–321.

Kobsa, A. (2001). Generic User Modeling Systems P. Brusilovsky, A. Kobsa, & W. Nejdl (eds.). User Modeling and UserAdapted Interaction. 11 (1). pp. 49–63.

Kobsa, A. (1993). User modeling: Recent work, prospects and hazards M. Schneider-Hufschmidt, T. Kühme, & U. Malinowski (eds.). HUMAN FACTORS IN INFORMATION TECHNOLOGY. 10 (1-2). pp. 111–111.

Kobsa, A. & Fink, J. (2006). An LDAP-based User Modeling Server and its Evaluation. User Modeling and UserAdapted Interaction. 16 (2). pp. 129–169.

Kobsa, A., Nill, A. & Fink, J. (1994). Adaptive hypertext and hypermedia clients of the user modeling system BGP-MS. Intelligent multimedia information retrieval. pp. 339–356.

Kolb, A.Y. & Kolb, D.A. (2005). Learning Styles and Learning Spaces : Enhancing Experiential Learning in Higher Education. Management Learning. 4 (2). pp. 193–212.

Korocsec, D., Holobar, A., Divjak, M. & Zazula, D. (2005). Building interactive virtual environments for simulated training in medicine using VRML and Java/JavaScript. Computer Methods and Programs in Biomedicine. 80 Suppl 1 (Supplement 1). pp. S61–S70.

Kruger, W., Bohn, C.A., Frohlich, B., Schuth, H., Strauss, W. & Wesche, G. (1995). The Responsive Workbench: a virtual work environment. Computer. 28 (7). pp. 42–48.

LaMothe, A. (2003). Tricks of the 3D Game Programming Gurus: Advanced 3D Graphics and Rasterization. Sams Publishing.

Lams (2004). Learning Activity Management System. Proceedings of the 20th Annual Conference of the Australasian Society for Computers in Learning in Tertiary Education ASCILITE. 2005 (April).

Lansdale, R. (2002). VRML2 Is an Excellent Conversion Route from Many CAD Packages. Press release of Okino Computer Graphics.

Larkin, J.H. & Simon, H.A. (1987). Why a Diagram is (Sometimes) Worth Ten Thousand Words. Cognitive Science. 11 (1) pp. 65–99.

Law, E.L.-C. & Rust-Kickmeier, M.D. (2008). 80Days: Immersive Digital Educational Games with Adaptive Storytelling. In *Proceedings of the First International Workshop on Story-Telling and Educational Games (STEG 2008)-The Power of Narration and Imagination in Technology Enhanced Learning.*

.Lazar, J., Feng, J.H. & Hochheiser, H., 2010. Research Methods in Human-Computer Interaction, Wiley Publishing.

Lepouras, G. & Vassilakis, C. (2006). An adaptive virtual reality architecture for shopping malls. Encyclopedia of ECommerce EGovernment and Mobile Commerce.

Lewinski, J. (2000). Developer's Guide to Computer Game Design. Plano, TX, USA: Wordware Publishing Inc.

Lindley, C.A. (2005). Story and narrative structures in computer games B. Brunhild (ed.). Developing Interactive Narrative Content. pp. 1–27.

Linek, S.B., Schwarz, D., Hirschberg, G., Rust, M.K. & Albert, D. (2007). Designing the Non-player Character of an Educational Adventure-game: The Role of Personality, Naturalism, and Color. In: Proceedings of the International Technology Education and Development Conference INTED..,.

Liu, S. & Gun, L. (2006). Construction of advanced virtual scene based on X3D. COMPUTER ENGINEERING AND DESIGN. 2. pp. 303–306.

Livingstone, D. & Kemp, J. (2008). Integrating Web-Based and 3D Learning Environments: Second Life Meets Moodle. UPGRADE The European Journal for the Informatics Professional. IX (3). pp. 8–14.

Lopes, R. & Bidarra, R. (2011a). A semantic generation framework for enabling adaptive game worlds. In: Proceedings of the 8th International Conference on Advances in Computer Entertainment Technology. 2011, Portugal: ACM, p. 6.

Lopes, R. & Bidarra, R. (2011b). Adaptivity Challenges in Games and Simulations: A Survey. IEEE TRANSACTIONS ON COMPUTATIONAL INTELLIGENCE AND AI IN GAMES. 3 (2). pp. 85–99.

Lu, J., Pan, Z., Lin, H., Zhang, M. & Shi, J. (2005). Virtual learning environment for medical education based on VRML and VTK. Computers & Graphics. 29 (2). pp. 283–288.

Mackinlay, J., S. Card, & Roberston, G. (1990) Rapid Controlled Movement through a Virtual 3D Workspace. Computer Graphics, 24(4): pp. 171-166.

Magerko, B. (2008). Adaptation in Digital Games. Computer. 41 (6). pp. 87–89.

Magerko, B. (2005). Story Representation and Interactive Drama R. M. Young & J. E. Laird (eds.). Artificial Intelligence. pp. 85–87.

Magerko, B.S. (2006). Player Modeling in the Interactive Drama Architecture. University of Michigan.

Manslow, J. (2002). Learning and Adaptation. In: S. Rabin (ed.). AI Game Programming Wisdom. Charles River Media, pp. 557–566.

Manslow, J. (2004). Using Reinforcement Learning to solve AI Control Problems. In: S. Rabin (ed.). AI Game Programming Wisdom 2. Charles River Media, pp. 591–603.

Marchiori, E.J., Torrente, J., Del Blanco, Á., Moreno-Ger, P., Sancho, P. & Fernández-Manjón, B. (2012). A narrative metaphor to facilitate educational game authoring. Computers & Education. 58 (1). pp. 590–599.

Marie-louise, L.J., Loria, K., Mostaghel, R. & Saha, P. (2009). E-Learning : Investigating University Student ' s Acceptance of Technology. European Journal of Open, Distance and e-Learning.

Martínez-Ortiz, I., Moreno-Ger, P., Sierra, J.L. & Fernández-Manjón, B. (2006). A General Architecture for the Authoring and the Operationalization of e-Learning Applications with Educational Modelling Languages. In *Proceedings of 8th Simposio Internacional de Informática Educativa, León, Spain.*

Martins, A.C., Faria, L. & Carvalho, C.V. De (2008). User Modeling in Adaptive Hypermedia Educational Systems. Educational Technology & Society. 11 (1). pp. 194–207.

McDerby, M.J., John, N.W., Brunt, J.N.H. & Kacperek, A. (2001). Modelling concepts of proton eye radiotherapy. Physiological Measurement. 22 (3). pp. 611–623.

McKnight, C., Dillon, A., & Richardson, J. (1991). Hypertext in Context: Cambridge University Press.

Medler, B. (2009). Using Recommendation Systems to Adapt Gameplay. International Journal of Gaming and Computer-Mediated Simulations. 1 (3). pp. 68–80.

Mildner, P., Campbell, C., Himmelsbach, M., Malassa, C., Miczka, M., Effelsberg, W. (2012). A serious game for architectural knowledge in the classroom. In Proceedings of the 7th international conference on Edutainment, and Proceedings of the 3rd international conference on E-Learning and Games for Training, Education, Health and Sports (Edutainment'12/GameDays'12), Stefan Göbel, Wolfgang Müller, Bodo Urban, and Josef Wiemeyer (Eds.). Springer-Verlag, Berlin, Heidelberg, pp. 72-77

Miller, G.A. (1956). The Magical Number Seven. [Online]. 1956. The Psychological Review. Available from: http://www.musanim.com/miller1956/.

Mills, C. & Dalgarno, B. (2007). A conceptual model for game based intelligent tutoring systems. *Proceedings of the 2007 Australasian Society for Computers in Learning in Tertiary Education,* pp. 692–702.

Missura, O. & Gaertner, T. (2009). Player Modeling for Intelligent Difficulty Adjustment. Discovery Science. 5808 LNAI. pp. 197–211.

Molka-Danielsen, J. (2011). Exploring the Role of Virtual Worlds in the Evolution of a Co-creation Design Culture. Lecture Notes in Business Information Processing. Springer Berlin Heidelberg. 68. pp. 3-15.

Monahan, T., McArdle, G. & Bertolotto, M. (2008). Virtual reality for collaborative e-learning. Computers & Education. 50 (4). pp. 1339–1353.

Moody, D. (2009). The "Physics" of Notations: Toward a Scientific Basis for Constructing Visual Notations in Software Engineering. IEEE Transactions on Software Engineering. 35 (6). pp. 756–779.

Moreno-Ger, P., Blesius, C., Currier, P., Sierra, J.L. & Fernández-Manjón, B. (2008a). Online learning and clinical procedures: rapid development and effective deployment of game-like interactive simulations Z. Pan, A. D. Cheok, W. Muller, & A. El Rhabili (eds.). Transactions on Edutainment I. LNCS 5080. pp. 288–304.

Moreno-Ger, P., Burgos, D., Martinezortiz, I., Sierra, J. & Fernandezmanjon, B. (2008b). Educational game design for online education. Computers in Human Behavior. 24 (6). pp. 2530–2540.

Moreno-Ger, P., Burgos, D. & Torrente, J. (2009). Digital Games in eLearning Environments: Current Uses and Emerging Trends. Simulation & Gaming. 40 (5). pp. 669–687.

Moreno-Ger, P., Martinez-Ortiz, I. & Fernández-Manjón, B. (2005). The project: Facilitating the Development of Educational Adventure Games. In: Cognition and exploratory learning in the digital age CELDA 2005 Porto Portugal IADIS. 2005, IADIS, pp. 353–358.

Moreno-Ger, P., Martínez-Ortiz, I., Sierra, J.L. & Fernández-Manjón, B. (2008c). A Content-Centric Development Process Model. Computer. 41 (3). pp. 24–30.

Moreno-Ger, P., Sierra, J.L., Martínez-Ortiz, I. & Fernández-Manjón, B. (2007). A documental approach to adventure game development. Science of Computer Programming. 67 (1). pp. 3–31.

Moreno-Ger, P., Sierra-Rodriguez, J.L. & Fernández-Manjón, B. (2008d). Game-Based Learning in e-Learning Environments. Next Generation Technology Enhanced Learning. IX (3). pp. 15.

Moreno-Ger, P., Torrente, J., Bustamante, J., Fernández-Galaz, C., Fernández-Manjón, B. & Comas-Rengifo, M.D. (2010). Application of a low-cost web-based simulation to improve students' practical skills in medical education. International Journal of Medical Informatics. 79 (6). pp. 459–467.

Mulwa, C., Lawless, S., Sharp, M., Arnedillo-Sanchez, I. & Wade, V. (2010). Adaptive Educational Hypermedia Systems in Technology Enhanced Learning: A Literature Review. In: Proceedings of the 2010 ACM Conference on Information Technology Education.

Murdock, K.. (2003). 3Ds max 5 bible. Wiley Publishing.

Mylopoulos, J. (1992). Conceptual modeling and Telos. In: P. Loucopoulos and R. Zicari (ed.). Conceptual modeling, databases, and CASE. Wiley, pp. 49–68.

Nejdl, W., Wolf, B., Qu, C., Decker, S., Sintek, M., Naeve, A., Nilsson, M., Palmér, M. & Risch, T. (2002). EDUTELLA: a P2P networking infrastructure based on RDF. In: Proceedings of the 11th international conference on World Wide Web. 2002, ACM, pp. 604–615.

Nelson, T.H. (1981). From Literary Machines. Proposal for a Universal Electronic. Publishing System and Archive. The New Media Reader CambridgeMassachusettsLondon. pp. 443–461.

NexLearn (2009). NexLearn. [Online]. 2009. Available from: http://www.nexlearn.com/.

Nielsen, J. (1993). Usability Engineering. J. Nielsen (ed.). Morgan Kaufmann.

Nussbaumer, A., Guetl, C., Albert, D. & Helic, D. (2009). Competence-based Adaptation of Learning Environments in 3D Space. In: IMCL2009 4th International Conference on Interactive Mobile and Computer Aided Learning. 2009, Amman-Jordan: International Association of Online Engineering, pp. 103–108.

O'Connor, T.J., Cooper, R.A., Fitzgerald, S.G., Dvorznak, M.J., Boninger, M.L., VanSickle, D.P. & Glass, L. (2000). Evaluation of a manual wheelchair interface to computer games. Neurorehabilitation and Neural Repair. 14 (1). pp. 21–31.

O'Neil, H., Wainess, R. & Baker, E. (2005). Classification of learning outcomes: evidence from the computer games literature. Curriculum Journal. 16 (4). pp. 455–474.

Octavia, J.R., Coninx, K. & Raymaekers, C. (2010). Enhancing User Interaction in Virtual Environments through Adaptive Personalized 3D Interaction Techniques..In *User Modeling, Adaptation, and Personalization*. Springer Berlin Heidelberg. pp. 423-426.

Octavia, J.R., Raymaekers, C. & Coninx, K. (2009). A Conceptual Framework for Adaptation and Personalization in Virtual Environments. In: DEXA Workshops 2009. 2009, pp. 284–288.

OpenQwaq (2007). OpenQwaq. [Online]. 2007. Available from: http://code.google.com/p/openqwaq/. [Accessed: 20 April 2013].

Oppenheim, A.N. (1992). Questionnaire Design, Interviewing and Attitude Measurement. P. P. L. N. Ed (ed.). Continuum International Publishing Group.

Oppermann, R., Rashev, R. & Kinshuk, K. (1997). Adaptability and Adaptivity in Learning Systems A. Behrooz (ed.). Knowledge Transfer. 2. pp. 173–179.

Outline3D (2000). www.outline3d.com. [Accessed: 20 April 2013].

Overmars, M. (2004). Teaching Computer Science through Game Design. Computer. 37 (4). pp. 81–83.

Paige, R.F., Ostroff, J.S., Brooke, P.J. (2000). Principles for modeling language design. In: Information and Software Technology, 42, pp. 665-675.

Palmer, I. (2001). Essential Java 3D: Developing 3D Graphics Applications in Java. Springer.

Paramythis, A. & Loidl-Reisinger, S. (2003). Adaptive Learning Environments and e-Learning Standards. Learning. 2 (1). pp. 181–194.

Peffers, K., Tuunanen, T., Rothenberger, M.A. & Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. Journal of Management Information Systems. 24 (3). pp. 45–77.

Peirce, N., Conlan, O. & Wade, V. (2008). Adaptive Educational Games: Providing Non-invasive Personalised Learning Experiences. 2008 Second IEEE International Conference on Digital Game and Intelligent Toy Enhanced Learning. pp. 28–35.

Pivec, M. & Pivec, P. (2008). Playing to Learn : Guidelines for Designing Educational Games . World Conference on Educational Multimedia Hypermedia and Telecommunications 2008. 2008 (1). pp. 3247–3252.

Prensky, M. (2001). Digital Natives, Digital Immigrants. On the Horizon. 9 (5). pp. 1–6.

Ram, A., Onta'on, S. & Mehta, M. (2007). Artificial intelligence for adaptive computer games. In: Twentieth International FLAIRS Conference on Artificial Intelligence. 2007.

Razzaq, L. & Heffernan, N.T. (2010). Hints : Is It Better to Give or Wait to Be Asked ? In: V. Aleven, J. Kay, & J. Mostow (eds.). Proceedings of the 10th international conference on Intelligent Tutoring Systems. 2010, Berlin, Heidelberg: Springer-Verlag, pp. 349–358.

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B. & Kafai, Y. (2009). Scratch: programming for all. Communications of the ACM. 52 (11). pp. 60–67.

Robertson, J. & Good, J. (2005). Adventure Author: An authoring tool for 3D virtual reality story construction. In: Artificial Intelligence in Education (AIED), Workshop on Narrative Learning Environments. 2005, Amsterndam, Netherlands.

Ross, S., Donnelly, M. & Dobreva, M. (2003). New Technologies for the Cultural and Scientific Heritage Sector. Salzburg Research.

Salt, B., Atkins, C. & Blackall, L. (2008). Engaging with Second Life : Real Education in a Virtual World Literature Review. Education. pp. 1–99.

Sancho, P., Moreno-Ger, P., Fuentes-Fernández, R. & Fernández-Manjón, B. (2009). Adaptive Role Playing Games: An Immersive Approach for Problem Based Learning. Educational Technology & Society. 12 (4). pp. 110–124.

Sastry, L. & Boyd, D.R.S. (1998). Virtual environments for engineering applications. Virtual Reality. 3 (4). pp. 235–244.

Satalich, G. (1995). Navigation and Wayfinding in Virtual Reality: Finding Proper Tools and Cues to Enhance Navigation Awareness. University of Washington.

Sauer, S., Osswald, K., Wielemans, X. & Stifter, M. (2006). U-Create : Creative Authoring Tools for Edutainment Applications. Culture. pp. 163 – 168.

Schaefer, J.J., Vanderbilt, A.A., Cason, C.L., Bauman, E.B., Glavin, R.J., Lee, F.W. & Navedo, D.D. (2011). Literature review: instructional design and pedagogy science in healthcare simulation. Simulation in healthcare journal of the Society for Simulation in Healthcare. 6. pp. S30–S41.

Schlickum, M.K., Hedman, L., Enochsson, L., Kjellin, A. & Felländer-Tsai, L. (2009). Systematic video game training in surgical novices improves performance in virtual reality endoscopic surgical simulators: a prospective randomized study. World Journal of Surgery. 33 (11). pp. 2360–2367.

Schneider, J., Garatly, D., Srinivasan, M., Guy, S.J., Curtis, S., Cutchin, S., Manocha, D., Lin, M.C. & Rockwood, A. (2011). Towards a Digital Makkah–Using Immersive 3D Environments to Train and Prepare Pilgrims. In: Islam Zeitschrift Für Geschichte Und Kultur Des Islamischen Orients. 2011, pp. 1–16.

Schneider, O. (2002). STORYWORLD CREATION: AUTHORING FOR INTERACTIVE STORYTELLING. In *WSCG*, pp. 405-412.

Seaman, B. (1993). Interactive Videodisc Production. New Media Technologies, pp. 99-122.

Siamak Barzegar, M.S. & Keyvanpour, M.R. (2010). Improvement User Interface in Virtual Learning Environment. International Journal of Digital Society (IJDS). 1 (3). pp. 221–229.

Slater, M. (1999). Measuring presence: A response to the Witmer and Singer presence questionnaire. Presence Teleoperators and Virtual Environments. 8 (5). pp. 560–565.

Slaughter, L. A., Harper, B. D. & Norman, K. L. (1994). Assessing the Equivalence of Paper and On-line versions of the QUIS 5.5. Proceedings of the 2nd Annual Mid-Atlantic Human Factors Conference, pp. 87-91, Washington, D.C.

Sleeman, D. & Brown, J.S. (1982). Intelligent Tutoring Systems D. Sleeman & J. S. Brown (eds.). Science. 228 (4698). pp. 456–462.

Smith, G.J., Gero, J.S. & Maher, M. Lou (2004). Towards Designing in Adaptive Virtual Worlds. Computer-Aided Design & Applications; 1 (4). pp. 701.

Social Learning Suite® (2006). Social Learning Suite®. [Online]. 2006. Available from: http://www.mzinga.com/a/pdf/MzingaDS-HRSolutions.pdf.

Spronck, P. (2005). A Model for Reliable Adaptive Game Intelligence. In: IJCAI-05 Workshop on Reasoning, Representation, and Learning in Computer Games. 2005, pp. 95–100.

Spronck, P., Ponsen, M., Sprinkhuizen-Kuyper, I. & Postma, E. (2006). Adaptive game AI with dynamic scripting. Machine Learning. 63 (3). pp. 217–248.

Stach, N., Cristea, A.I., De Bra, P. (2005), Explicit Intelligence in Adaptive Hypermedia: Generic Adaptation Languages for Learning Preferences and Styles. International Workshop on Combining Intelligent and Adaptive Hypermedia Methods/Techniques in Web-Based Education Systems, HT'05, Salzburg, Austria, ACM. pp. 6-9.

Stanney, K. (2003). Usability engineering of virtual environments (VEs): identifying multiple criteria that drive effective VE system design. International Journal of Human-Computer Studies. 58 (4). pp. 447–481.

Thomas, J.M. & Young, M. (2007). Becoming Scientists: Employing Adaptive Interactive Narrative to Guide Discovery Learning. In: AIED-07 Workshop on Narrative Learning Environment. 2007.

Thue, D., Bulitko, V., Spetch, M. & Wasylishen, E. (2007). Interactive Storytelling : A Player Modelling Approach J. Schaeffer & M. Mateas (eds.). Artificial Intelligence. Associatio (July). pp. 43–48.

Togelius, J., De Nardi, R. & Lucas, S.M. (2007). Towards automatic personalised content creation for racing games. 2007 IEEE Symposium on Computational Intelligence and Games. (Cig). pp. 252–259.

Toolkit®, V.V.R. (1998). Vizard Virtual Reality Toolkit®. [Online]. 1998. Available from: http://www.aesthetic.com/home_frame/home_frame.htm.

Torrente, J., Moreno-Ger, P. & Fernández-Manjón, B. (2008). Learning Models for the Integration of Adaptive Educational Games in Virtual Learning Environments Z. Pan, X. Zhang, A. El Rhalibi, W. Woo, & Y. Li (eds.). Learning. 5093. pp. 463–474.

Triantafillou, E., Georgiadou, E. & Economides, A.A. (2006). Adaptive Hypermedia Systems: A review of adaptivity variables. *Proceedings of the Fifth Panhellenic Conference on Information and Communication Technologies in Education, Thessaloniki, Greece*, pp. 75-82.

Truevision3D® (2007). Truevision 3D®. [Online]. 2007. Available from: http://www.truevision3d.com/page-14-create-3d-game-development. [Accessed: 20 April 2012].

Udine3D (2011). Udine3D. [Online]. 2011. Available from: udine3d.uniud.it. [Accessed: 20 April 2013].

Unity® (2012). Unity®. [Online]. 2012. Available from: http://unity3d.com/unity/. [Accessed: 20 April 2013].

Van Deursen, A., Klint, P. & Visser, J. (2000). Domain-specific languages: An annotated bibliography. ACM Sigplan Notices. 35 (6). pp. 26–36.

VastparkCreator (2007). Vastpark Creator®. [Online]. 2007. Available from: http://www.vastpark.com/. [Accessed: 20 April 2013].

Verpoorten, D., Poumay, M. & Leclercq, D. (2007). The eight learning events model: A pedagogic conceptual tool supporting diversification of learning methods. Interactive Learning Environments. 15 (2).pp. 151–160.

Vilozni, D., Barker, M., Jellouschek, H., Heimann, G. & Blau, H. (2001). An interactive computer-animated system (SpiroGame) facilitates spirometry in preschool children. *American journal of respiratory and critical care medicine*, *164*(12), pp. 2200-2205.

Vince, J. (2004). Introduction to Virtual Reality. 1st edition. Springer-Verlag.

Vinson, N.G. (2003). Design guidelines for landmarks to support navigation in virtual environments. In: SIGCHI conference on Human factors in computing systems the CHI is the limit CHI 99. 2003, Pittsburgh, Pennsylvania, United States: ACM Press, pp. 278–285.

Virtual Human Project (2002). Virtual Human Project. [Online]. 2002. Available from: http://www.virtual-human.org.

Virtual Showrooms (2011). Virtual Showrooms. [Online]. 2011. Available from: http://www.virtualshowrooms.co.za.

Virvou, M. & Katsionis, G. (2008). On the usability and likeability of virtual reality games for education: The case of VR-ENGAGE. Computers & Education. 50 (1). pp. 154–178.

Visual3D.net® (2006). Visual3D.net®. [Online]. 2006. Available from: http://www.visual3d.net/.

Warburton, S. (2009). Second Life in higher education: Assessing the potential for and the barriers to deploying virtual worlds in learning and teaching. British Journal of Educational Technology. 40 (3). pp. 414–426.

Westera, W., Nadolski, R.J., Hummel, H.G.K. & Wopereis, I.G.J.H. (2008). Serious games for higher education: a framework for reducing design complexity. Journal of Computer Assisted Learning. 24 (5). pp. 420–432.

Westra, J., Dignum, F. & Dignum, V. (2011). Scalable Adaptive Serious Games using Agent Organizations. Proc of 10th Int Conf on Autonomous Agents and Multia gent Systems AAMAS 2011. pp. 1291–1292.

Westra, J., Van Hasselt, H., Dignum, F. & Dignum, V. (2009). Adaptive Serious Games Using Agent Organizations. F. Dignum, J. Bradshaw, B. Silverman, & W. Doesburg (eds.). Agents for Games and Simulations. 5920. pp. 206–220.

Weusijana, D., Svihla, V., Gawel, D. & Bransford, J. (2007). Learning about Adaptive Expertise in a Multi-User Virtual Environmen. In: Proceedings of the Second Life Education Workshop at SLCC. 2007, San Francisco, pp. 40–45.

Wiecha, J., Heyden, R., Sternthal, E. & Merialdi, M. (2010). Learning in a Virtual World: Experience With Using Second Life for Medical Education G. Eysenbach (ed.). Journal of Medical Internet Research. 12 (1). pp. e1.

Witmer, B.G. & Singer, M.J. (1998). Measuring Presence in Virtual Environments: A Presence Questionnaire. Presence Teleoperators Virtual Environments. 7 (3). pp. 225–240.

Xun, O., Ahmad, K. & Richard, T. (2006). Constructing Virtual Environments for Real Time e-learning. International Journal of Computer Science and Network Security. 6 (5). pp. 1–5.

Yang, J. & Zhang, J. (2010). Towards HTML 5 and interactive 3D graphics. Educational and Information Technology ICEIT 2010 International Conference on. 1. pp. 522–527.

Yannakakis, G.N. & Togelius, J. (2011). Experience-Driven Procedural Content Generation. IEEE Transactions on Affective Computing. 2 (3). pp. 1–16.

Yun, C., Shastri, D., Pavlidis, I. & Deng, Z. (2009). O ' Game , Can You Feel My Frustration?: Improving User ' s Gaming Experience via StressCam. In *Proceedings of the 27th international conference on Human factors in computing systems*, ACM, pp. 2195-2204.

Yusoff, M.F., Zulkifli, A.N. & Mohamed, N.F.F. (2011). Virtual Hajj (V-Hajj) — Adaptation of persuasive design in virtual environment (VE) and multimedia integrated approach learning courseware methodology. In: IEEE Conference on Open Systems. 2011, IEEE, pp. 250–255.

A. Appendix: Authoring Evaluation Scenario: Course on Solar System

Below we present the task the participants of the evaluation of the adaptive 3D VLE authoring tool were required to perform. The evaluator handed over a description of the graphical modelling languages to be used for authoring the Pedagogical Model, the Adaptive Storyline Model, and the Adaptive Topic Model. Following that, the evaluator asked to create a 3D VLE solar system course using the proposed graphical modelling languages. All the material required was provided and someone was available to help the participants wherever needed.

The next subsections provide the description provided to the participants of the evaluation: an introduction describing the evaluation task (section A.1), a description of the modelling languages (section A.2), a description of the specific tasks to be performed (section A.3), and the evaluation questionnaire (section A.4).

A.1 Introduction

You are required to model an adaptive 3D virtual learning environment (i.e. course) on the solar system by using the three graphical modelling languages given. The purpose of this course is to give students a general introduction to the solar system.

The course should show a 3D virtual learning environment with the planets (Earth, Mars, etc.) and the Sun in order to provide the learner/student a notion of distance between the different planets and Sun.

To add pedagogical and educational aspects to the 3D solar system course, pedagogical relationships can be defined. For instance, the prerequisite pedagogical relationship type can be used between different planets and Sun.

A storyline to be followed by learner should be defined too. The 3D solar system course is divided into sub topics like *Learn about Stars, Inner Solar system, Outer Solar system, Moons and Advanced Topics.* Every single topic includes learning concepts represented by different 3D models.

In order to take the knowledge and progress of the learner into consideration, the 3D learning environment can be dynamically adapted. How this adaptation should be done must be specified while modelling the course.

The task to be performed is the modelling of one topic within the course: Inner Solar System. The topic is as follows. In The Inner Solar System, the student should start with learning about the planets that are related to the inner part of the solar system *Mercury, Venus, Mars* and *Earth*. To guide the student what to learn, first all planets should be semi hided[61]. The Sun should be marked by adding a blue square around it[62] and annotate it by its name[63]. After a number of interactions (for instance 3 times) by the learner, Mercury should first be displayed and start to rotate around its axis and on its orbit. Furthermore, you need to specify a text message to be shown to student. This message is needed to explain some additional information about Mercury and its behaviour. Once the user/learner has interacted with Mercury and the learner's knowledge about Mercury is sufficient, Mercury rotation and interaction should be deactivated, and the learner should be taken towards Venus; now Venus should now be marked. A similar learning approach should be used for the other planets of this topic. The course should end by displaying and lighting the last planet in the inner solar system part, planet Earth.

In order to achieve this adaptive 3D virtual learning environment about the solar system, the course author (you) have to use a number of modelling languages. Next section will explain different notations used for the three modelling languages: Pedagogical Model Language, Adaptive Storyline Language and Adaptive Topic Language. Furthermore, some adaptation techniques that can be used inside 3D virtual learning environment are also presented.

A.2 Modelling Languages

A.2.1 Pedagogical Model Language

The Pedagogical Model Language (PML) is used to define the pedagogical structure between the learning concepts. For example, we can specify that concept A is a pre-requisite for concept B. It is important to specify this pedagogical structure, as it will update the information about the learner (stored in the User Model) automatically. The pedagogical relationships are applied to the learning concepts of the Domain Model of the course.

In the Pedagogical Model Language 3D learning concepts are represented as rectangles (see Figure 1-A). These rectangles can be connected through labelled arrows (see Figure 1-B). The arrows represent the pedagogical relationships between the 3D learning concepts. Every pedagogical relationship has a specific colour. For example, the *pre-requisite* is given the green colour.

---

[61] By means of an adaptation strategy
[62] Using the bounding box adaptation type
[63] Using the annotation adaptation type

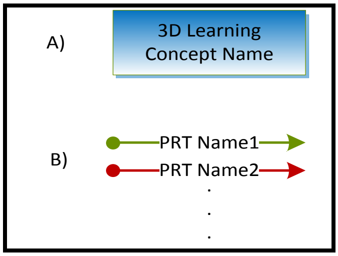


Figure A.2.1 : PML elements; A) learning concept, B) pedagogical relationship (between learning concepts)

A.2.1.1 Pedagogical Relationship Types

A typical Pedagogical Relationship Type is the *prerequisite*. When a learning concept X is a prerequisite for learning concept Y, it means that the learner needs to study concept X before he can start with concept Y. All Pedagogical Relationship Types (PRT) are limited to binary relationships. Furthermore, updating rules are associated with every PRT to define how the user model will be updated, i.e. which attributes of the user model should be updated and how. The list of predefined PRTs and their associated default update rules are given. If there meaning is not intuitive, you can ask for explanation. Note that the attribute "suitability" is used to indicate that the learner (user) is allowed to learn the concept; "knowledge" is used to indicate the level of knowledge of a learner about a concept. UserID is used to uniquely identify the learner.

- X *Prerequisite_for* Y: This is used to express that knowledge about concept X is required to learn concept Y.

  Default associated update rule:

  If (UserID.3DLearningConceptX.knowledge> 'Average_Level')
  Then (UserID.3DLearningConceptY.suitability='TRUE')

- X *Defines* Y: this is used to connect two concepts where one defines the other; X defines Y

  Default associated update rule:

  If (UserID.3DLearningConceptX.knowledge>'Certain_Level')
  Then (UserID.3DLearningConceptY.knowledge ='UpdateVal')

- X *Illustrates Y*: This is used to express that concept X illustrates another learning concept, concept Y.

  Default associated update rule:

  If (UserID.3DLearningConceptX. suitability = 'TRUE')
  Then (UserID.3DLearningConceptY.suitability='TRUE')

- X *Interested_for* Y: This is used to indicate that concept X may be of interest while learning concept Y. For example, if the learner is learning about earth, it might be interesting to also learn about its moon. So both Moon and Earth 3D learning concepts could be connected through the interest-PRT.

Default associated update rule:

If (UserID.3DLearningConceptY. knowledge = 'Certain_Level')
Then (UserID.3DLearningConceptX.interest= 'IncreaseLevel')

- X *updates_knowledge_of* Y: This is used to indicate that the user knowledge of concept Y is also updated upon accessing concept X. For instance, learning about Mars (specific concept) can also update the knowledge about Earth (specific concept).

Default associated update rule:

If (UserID.3DLearningConceptX. suitability = 'TRUE)
Then (UserID.3DLearningConceptY.knowledge = 'MaximumLevel')
Else (UserID.3DLearningConceptY.knowledge = 'Increased_Level')

Figure 2 depicts an example of a Pedagogical Model. An Oral explanation will be provided about it.

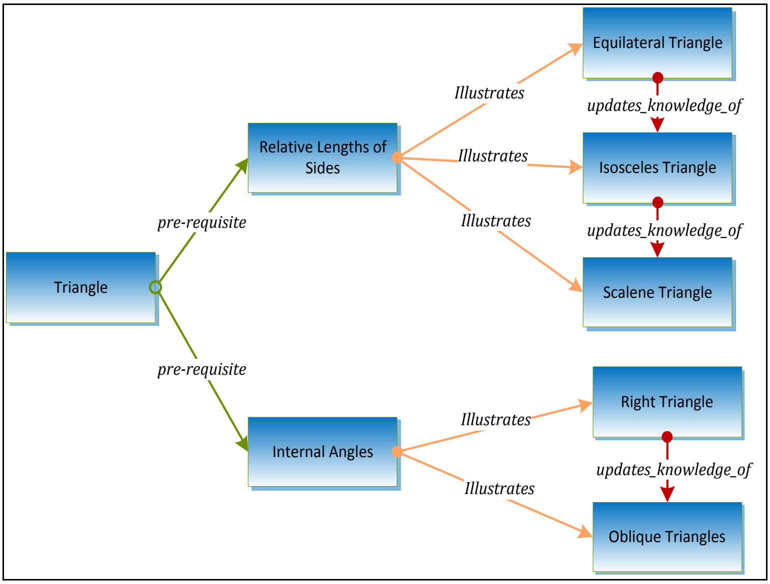


Figure 2: Example about Pedagogical modelling language for Learning Triangular Shapes

A.2.2 Adaptive Storyline Model Language

The storyline is used to express the scenario the learner needs to follow in the course. A storyline can be made up of individual topics, each topic having a specific purpose. A topic can be used to treat a specific learning topic, which can be composed of different learning concepts. The storyline can be delivering in an adaptive way (depending of the user model), which means that the topics will be delivered in an adaptive sequence. To achieve this, *storyline adaptation rules* are used. A topic adaptation rule is associated with a topic, called the source topic, and is used to control the transition to the next topic, being the target topic.

The modelling language to create an adaptive storyline model is as follows. Figure 3-A represents the notation for the start of the storyline, an ellipse named *Start*. Figure 3-B shows the core building block of a storyline, a topic. A topic is represented by a filled ellipse containing the name of the topic. Note that the details of a topic will be given in the Adaptive Topic Model (see section A.2.2.1). Remember that a topic covers all learning concepts that are related to the story (or topic).

To specify adaptivity in the storyline, *storyline adaptation rules* should be used. A topic adaptation rule is a condition-action rule, and is used to specify when (expressed by a condition) the learner should move to the next topic and what kind of adaptations should be applied to the learning concepts in the next topic (the action). In Figure 3-C the notation used for an adaptation rule is shown. The condition (see Figure 3- D) is represented by a hexagon containing the condition (here represented by the letter C) along with an arrow pointing to the action part of the rule. For the action part of a topic adaptation rule, a pentagon shape (see Figure 3-E) is used containing the name of the adaptation strategy that need to be applied to the target topic's 3D learning concepts. For instance, the following is informal syntax for the topic adaptation rule between two topics (TopicX and TopicY). Once the user has the required knowledge about TopicX and the suitability of TopicY is applicable, then he can go to TopicY and the adaptation strategy called TourGuide should be applied on TopicY.

> **If**(UserID.Topic**X**.knowledge='Required_Level' and UserID.Topic**Y**.suitability='TRUE')
> **Then** (TourGuide (TopicY))

The end of the storyline is represented by double ellipse named *End* (see Figure 3-F). It worth to mention that, both start and end symbols are considered as special nodes which includes text messages to be presented to the learner. The text messages should be defined by the author.

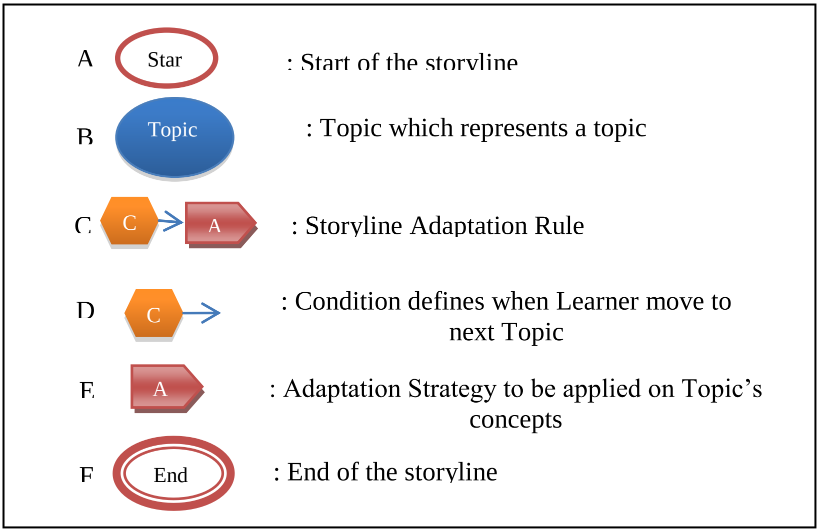


Figure 3: A) start of a storyline; B) Topic in a storyline; C) storyline adaptation rule; D) condition; E) adaptation state; F) end of a storyline.

Figure 4 depicts an example of a Pedagogical Model. An Oral explanation will be provided about it.

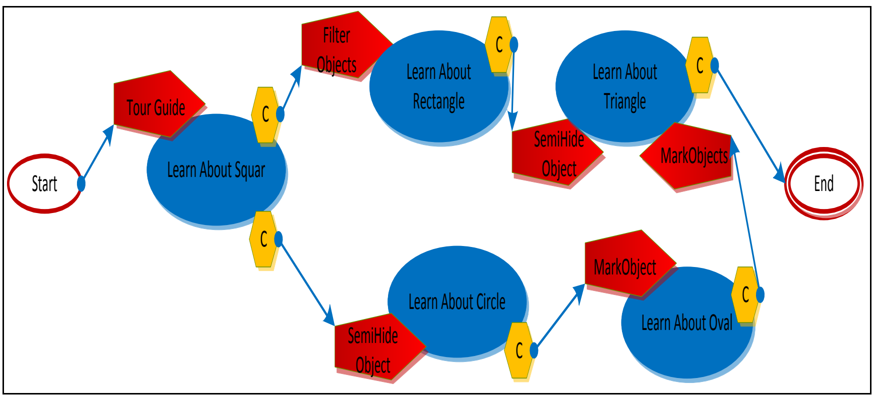


Figure 4: Example about adaptive storyline language for Learning Shapes

A.2.2.1 Adaptation Strategies

Adaptation strategies deal with adaptations that go beyond the adaptation of a single component of the 3D virtual learning environment. If an adaptation strategy is used in the context of an adaptive storyline, it is applied on the components in the topic for which it is specified.

A first group of adaptation strategies available are those that will have an impact on how the learner can navigate through the 3D environment:

- *restrictedNavigation*: this adaptation strategy allows restricting the navigation 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 type of navigation, jumping or waking. By jumping, one goes from one object to the next without viewing the part of the scene between the two objects. In general, this adaptation strategy can be used to force a learner to visit only a pre-defined number of objects and in a given sequence.
- *navigationWithRestrictedBehaviour*: this adaptation strategy allows restricting the possible behaviours of objects while navigating. The restricted behaviour can apply on all objects (in the topic) or on a specified list of objects. This strategy is for instance useful to allow a learner to first explore a (part of) the 3D environment without being annoyed by objects showing all kinds of behaviour; afterwards when he is more familiar with the 3D virtual learning environment, behaviour can be enabled (using another adaption strategy).
- *navigationWithRestrictedInteraction*: this adaptation strategy allows restricting the possible interactions with some objects while navigating. Similar as for navigationWithRestrictedBehaviour, this adaptation strategy can be used to allow a learner to first explore the 3D environment (or a part of it) without being able to fully interact with the objects in the scene; afterwards when he is more familiar with the 3D VLE, interactions can be enabled (using another adaption strategy).
- *TourGuide*: this strategy can be used to provide a tour guide to the learner. A tour guide takes the learner through a tour in a 3D 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 3D VLE.

The following adaptation strategies allow specifying that the learner can navigate freely in the 3D VLE. However, we distinguish two types of free navigation strategies, namely *completelyFree* and *freewithSuggestions*.

- *completelyFree*: this adaptation strategy will allow the learner to navigate freely in the 3D environment. This strategy can for instance be used in exploration-based learning.
- *freeWithSuggestions*: this adaptation strategy will allow the learner to navigate freely in the 3D environement but in addition some objects will be "suggested". 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 are those that allow adapting a group of objects:

- *filterObjects*: this adaptation strategy allows to filter the objects that should be available (visible) in the 3D VLE. The filtering is possible according to pedagogical criteria. This strategy can be used to avoid that the learner get lost in the 3D environment or that he doesn't know on which objects to focus first. It also allows gradually building the 3D environment; the more knowledge the learner obtains the more objects become visible.
- *markObjects*: this adaptation strategy allows to mark (i.e. by highlight or spotlight – see section A.2.3.1) a number of objects in the 3D environment. The marking will be done for all objects satisfying a certain (pedagogical) criteria. 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 3D environment 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*: this adaptation strategy allows to specify when some objects should not be displayed anymore. The condition can be given by means of some pedagogical criteria like the knowledge level that the learner currently has for the object or by setting a limit on the number of times the object should be displayed. This last case can for instance be useful if the purpose is to perform some tests in the 3D environment without having the subject(s) of the study visible.

- *displayAfter*: This adaptation strategy allows to specify the condition(s) that need to be satisfied for objects to be displayed. This condition can be given by means of some pedagogical criteria or by means of a time criterion. This adaptation strategy can for instance be used to keep the 3D environment appealing by dynamically changing some of 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 behaviour of some objects conditionally. As for the previous group of adaptations, the conditions can be some pedagogical criteria or time criteria:

- *behaviourAtMost*: this adaptation strategy allows to specify when (by means of a condition) a behaviour 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 behaviour.
- *behaviourAfter*: this adaptation strategy allows to indicate when (by means of a condition) a certain behaviour should be executed. This adaptation strategy can for instance be used to state that the behaviour for some specific objects should only start when the knowledge level of the learner for these objects is above a certain threshold.
- *behaviourSpeed*: this adaptation strategy allows to specify the speed of a behaviour. This adaptation strategy is in particular useful when the behaviour simulates real world behaviour. Being able to slow down the behaviour will allow the learner to better observe what is happing, especially for behaviours that are happing 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. As for the previous two groups of adaptations, the conditions can be given by means of some pedagogical criteria or by means of a time criterion.

- *interactionAtMost*: this adaptation strategy allows to specify when the learner cannot interact with the objects anymore. Similar as for the behaviourAtMost strategy, this adaptation strategy can for instance be used to avoid that the learner keeps "playing" with objects.

- *interactionAfter*: this adaptation strategy allows to specify when (by means of a condition) the learner can interact with some objects. This adaptation strategy can be used to ensure that the learner has enough knowledge about the objects before he can interact with it.

A.2.3 Adaptive Topic Language

With the Adaptive Topic Language, the author can specify the adaptivity within a single topic. He can control the order in which the students must explore the learning contents (3D learning objects), adapt the content of the 3D virtual learning environment and control what a learner can do in the 3D virtual learning environment based on different conditions (e.g., the number of times that the student interacts with a 3D object). Therefore, the following graphical language that allows the authors to connect 3D learning concepts related to the topic with adaptation types is available. See section A.2.4.1 for list of available adaptation types.

A Topic is composed of 3D learning objects (see Figure 5-A) which are connected through labelled arrows (see Figure 5-B). 3D learning objects are represented as a rectangle including the name of the learning concepts. The arrows represent an adaptation rule and is used to indicate an adaptation on the learning concept to which the arrow points. The arrow name contains the adaptation type to be applied. For instance, *hide* represents the hiding adaptation type used to hide the target object. However, it may be needed to apply different adaptation types to the same 3D learning concept. Therefore, the author will be able to specify additional adaptations by using an arc on the learning concept named with the adaptation type to be applied (see Figure 5-C). Every adaptation type is attached with two other elements, an event and a condition, to completely specify the *event-condition-action* adaptation rule. Together they specify when the adaptation type should be applied. The event part represents a VR event. An event is represented by a diamond (see Figure 5-D). A condition is represented by a hexagon with the letter *C* inside it. Condition can be given a name to make the graphical language more readable and to provide more information about the adaptation. Both elements are attached to the arrow or arc. Finally, Figure 5-E shows a cloud symbol which is used to indicate that a text message should be displayed to the learner when an adaptation type is applied. Such a message could be used for different purposes. For instance, it can be used to inform the student about the next step; explain what is currently happening inside the 3D VLE; guide him to perform a particular action; etc. More details are given in the example further on.

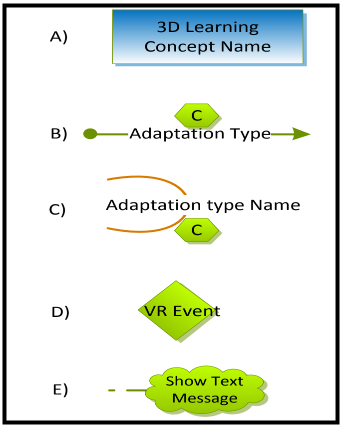


Figure 5: ASL elements: A) 3D learning concept, B) Adaptation type to be applied on a 3D learning concept, C) Adaptation type to be applied to a 3D learning Concept, D) VR event E) Text Message

A VR Event expresses an event that could happen in the 3D learning environment or on the 3D learning object to which it is attached.. The following events can be used in this context:

- ***Timer***: Generates an event after a predefined time. For instance, a course author wants to deactivate, for a while, 3D object behaviour after it has been watched by the student for specific period of time.
- ***Close To***: Generates an event when the user enters, exits and moves within a region of space (defined by a box) around 3D Object.
- ***Collision***: This event is triggered when an object collision-detection happened.
- ***StringEntered***: Generates an event when the user entered a predefined string.
- ***Touch:*** Detects when the user touches (e.g., by clicking with the mouse) a learning object.
- ***KeyPressed***: Generates an event when the user presses a key from the keyboard.

Figure 6 depicts an example of an Adaptive Topic Model. An oral explanation will be provided about it.

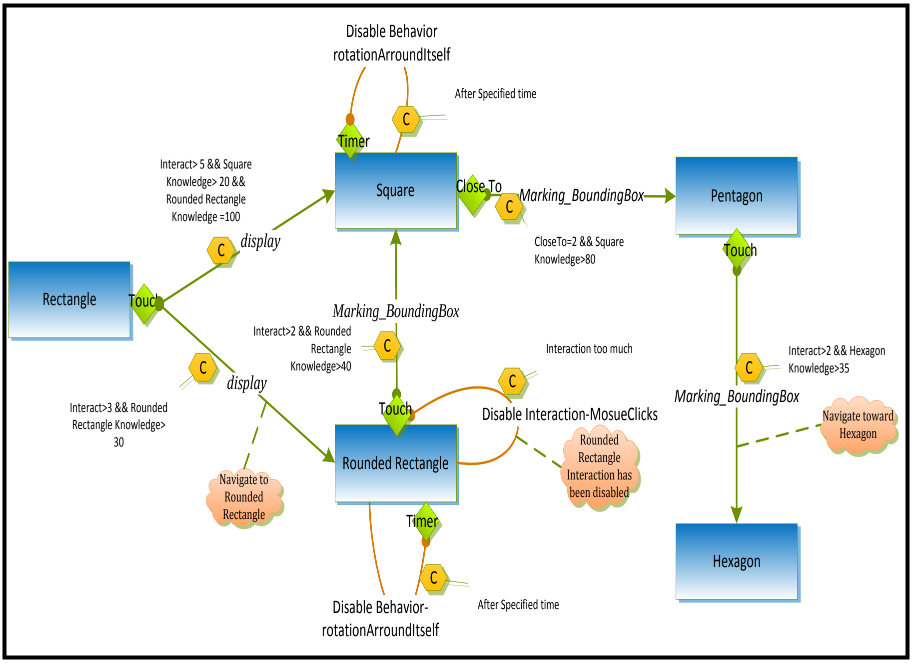


Figure 6: Example about Adaptive Topic Model for Geometric Shapes

A.2.3.1 Adaptation Types

Adaptation types are adaptations that apply on a single component of the 3D virtual learning environment. There are adaptation types for 3D virtual objects, for 3D objects behaviour, for user interaction, and for navigation.

Adaptation Types for Objects

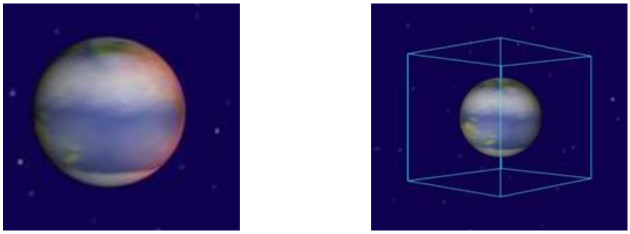
The adaptation types for objects are concerned with the adaptation of the visualization of an object, i.e. how to display it and how to hide it, and with marking them, i.e. using a spotlight or put a box around the object or associate it with an annotation.

- *semiDisplay*: this adaptation type is used to display the object in a semi-manner, by having a semi-transparent bounding box around it. See figure 7(a) for an example: the Sun has been semi-hided.
- *changeSize*: this adaptation type is used to change the visual appearance of an object by changing its size.
- *changeMaterialProperties*: this adaptation type is used to change the material properties (colour or texture) of an object. Figure 7 (b) shows an illustration of changing the texture of the sun.

- *changeVRRepresentation:* this adaptation type is used to change the visual representation of an object completely; its current visualization will be replaced by a different representation.
- *semiHide*: this adaptation type allows the visual appearance of objects to be semi-hidden. This is similar as the semiDisplay but the purpose is slightly different, i.e. to hide the object in a semi-manner.
- *hide*: this adaptation type allows hiding an object visually.
- *display*: this adaptation type allows displaying an object that has been hidden before.
- *spotlight:* this adaptation type allows to mark an object by putting a spotlight on the object; in this way the object becomes more visible and can be used to draw the attention of the learner to this object. Figure 8(a) shows an illustration of the planet earth being highlighted with a red spotlight.
- *highlight*: this adaptation type allows to mark an object by drawing a box around the object where only the edges of the box are displayed. Figure 8(b) shows the planet earth being highlighted.
- *displayAnnotation*: this adaptation type allows displaying an annotation (e.g., text message) with an object.
- *hideAnnotation*: this adaptation type allows hiding an annotation associated with an object.

(a) semiDisplay (b) changeMaterialProperties

Figure 7: Different visual representation for Sun



(a) spotlight (b) highlight

Figure 8: Earth with spotlight and highlighting box techniques

Adaptation Types for Behaviours

Behaviours are used to make the 3D VLE dynamic, i.e. to create environments where objects are active by performing some behaviour. However, to guide the learning process it may be useful to disable and enable behaviours when appropriate, e.g., a solar system where all planets are rotating at the same time may be very confusing for a beginner. It may also be useful to adapt the parameters of a behaviour, for instance to show the behaviour of the sun with a different value for its temperature. Possible adaptation types for behaviours are:

- *enableBehaviour*: this adaptation type allows enabling a behaviour associated with an object.
- *disableBehaviour*: this adaptation type allows disabling a behaviour associated with an object.
- *changeBehaviour*: this adaptation type allows changing a behaviour by modifying the values of its parameters.

Adaptation Types for User Interactions

In a 3D VLE, there are different ways to interact with an object, e.g., by clicking on an object, by touching an object, by passing closed by an object. Furthermore, interaction can trigger the start (or end) of behaviour. For example, by interacting with an object the end-user can move it or start behaviour. In the context of a 3D VLE, it may also 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 the possible types of interaction provided for an object. Possible interaction types considered for objects are: touching, clicking, passing by, and manipulation:

- *enableInteraction*: this adaptation type allows enabling an interaction type (given as parameter) for an object.
- *disableInteraction*: this adaptation type allows disabling an interaction type (given as parameter) for an object.

A.3 Tasks to be performed

A.3.1 Create Pedagogical Model

You are required to define the Pedagogical Model of the solar system course using the notations given in section A.2. Furthermore, the list of pedagogical relationship types is provided in section A.2.1.1. If you like to customize the default update rule associated with a pedagogical relationship type, then use the table given under the canvas.

Pedagogical Model Canvas

Updating Rules for used PRTs

PRT Name	Defined Updating Rule Format

A.3.2 Create the Adaptive Storyline

The storyline to be followed by learner should be defined too. The 3D solar system course should be divided to sub topics: *Learn about Stars, Inner Solar system, Outer Solar system, Moons,* and *Advanced Topics.* Below are the requirements for the storyline to be defined:

Step 1: At the start of the storyline, the learner should be giving a guided tour to the 3D learning objects related to the topic called "Learn about Stars".

Step 2: After crossing a defined threshold (current topic knowledge is more than specific value and the suitability of "inner solar system" topic is true), the learner should be able to see 3D learning objects related to the Inner Solar system by using *markingObjects* adaptation strategy. On the other hand, if the other defined threshold (current topic knowledge is more than specific value and the suitability of "outer solar system" topic is true) is crossed then the learner could be able to view filtered objects by using *filterObjects* adaptation strategy related to outer Solar System.

Step 3: In case the learner was guided to Inner Solar System topic, then a guided tour will be provided through the different 3D moons in the Moons topic. This will happen once the condition part of the defined adaptation rule in Moons topic is activated, then a filtered 3D objects will be displayed in the outer solar system.

Step 4: After a defined threshold (current topic knowledge is more than specific value and the suitability of "Advanced Topics" topic is true), is crossed, the 3D objects related to the Advanced Topics topic should be marked.

Step 5: As the Advanced Topics topic is the last one, a text message should be displayed to the learner when he finishes the course.

The storyline should be modelled using the notations given in section A.2.2 in the space provide below. The topic adaptions rules used should be given in the table below the canvas by means of their name; source and target topic should be indicated. The learning concepts contained in each topic should be given in the second table provided.

Adaptive Storyline Model Canvas

Source Topic	Storyline Adaptation Rules	Targeted Topic

Topic Name	Related 3D Learning Concepts

A.3.3 Create Adaptive Topic Model

Below are the requirements for the Inner Solar System Topic:

Step 1*:* At the start of the topic, the 3D object Sun should be marked with a blue edge cube and annotated by the label "This is Sun".

Step 2: After the third interaction of the learner with Sun and having the certain required knowledge about mercury, the 3D object Mercury should be displayed and start to rotate around its axis.

Step 3: Again the second interaction of the learner with Mercury, Mercury should stop moving around its orbit.

Step 4:

- When the learner interacts once more with Mercury, the learner should not be able to interact with it anymore.
- A text message should be displayed to inform the learner that he is not able to interact anymore with Mercury.

Step 5: The rest of planets will follow the same scenario of adaptation.

The topic should be modelled using the notations given in section A.2.3 in the space provide below.

Adaptive Topic Model Canvas

A.4 Evaluation Questionnaire

You have been asked to participate in an Adaptive 3D Virtual Learning Environment authoring evaluation. This study aims at evaluating the modelling languages that are used to create an adaptive 3D Virtual Learning Environment. We would like to get your feedback on the usability of the languages and possible benefits or drawbacks. The outcomes of the evaluation will give evidence of the quality of the provided modelling languages and will be used to derive ideas on how they can be further improved.

In general, there are no right or wrong answers. We want to know your opinion and viewpoints. Note that in no way this is an evaluation of you personally, only the languages and their notations will be evaluated. If you have any questions about the study, please feel free to ask.

I understand this information and agree to participate.

There are 55 questions in this survey

Start Online Survey

A. Demographic Questionnaire

Thank you for participating in this study! First we would like to collect some background data that are relevant to our research work. Please answer the questions below. Your data and the information collected in this evaluation will be treated anonymously.

1. Person code............
2. Gender:
   - Male
   - Female
3. Profession
   Please write your answer here:
4. Age
   Please write your answer here:
5. How often do you use computers?
   - Never
   - Less frequently than once a week
   - Several times a week
   - Daily
6. How often do you use 3D Video Games, Virtual Reality or 3D Environments?
   - Never
   - Less frequently than once a week
   - Several times a week
   - Daily

7. How much experience do you have in authoring (i.e. creating) courses with a learning/authoring environment, e.g AHA?

- Haven't so far
- For one to several months
- For about 1 year
- For years

8. Have you ever authored (i.e. created) a Virtual Reality application or course?

- Yes
- No

9. Do you have good knowledge of graphical modelling notations (e.g., ER, UML, ORM, ...)?

- Yes
- No

10. How much time in minutes did you spend approximately on creating the adaptive 3D Virtual Learning Environment course using the three modelling languages?

Please write your answer here:

B. Questionnaire on adaptive 3D VLE Authoring (A3DVLE)

The following questions address specifically the authoring process and your perceptions and opinion.

Please answer the following questions by indicating how much you agree for each statement.

1-Very Low 5=Very High

1. The proposed modelling languages (Pedagogical Model Language, Adaptive Storyline Language and Adaptive Topic language) allow creating the specified scenarios.

   1 2 3 4 5

2. The proposed modelling languages (Pedagogical Model Language, Adaptive Storyline Language and Adaptive Topic language) are easy to use.

   1 2 3 4 5

3. The graphical notations (symbols) used were clear

   1 2 3 4 5

4. Defining the Pedagogical Model was very easy.

   1 2 3 4 5

5. Using the Adaptive Storyline Language to specify the main flow of the course was very easy.

   1 2 3 4 5

6. Using the Adaptive Topic Language to specify the flow of a topic was very easy.

   1 2 3 4 5

7. Using the Adaptive Topic Language to specify the adaptations required for the 3D learning objects in a topic was very easy.

   1 2 3 4 5

8. The language/terminology used in explanation is understandable.

   1 2 3 4 5

9. The language/terminology and symbols used in the graphical notation are difficult to learn.

   1 2 3 4 5

10. The goal of the Pedagogical Model is clear.

   1 2 3 4 5

11. Using the Pedagogical Model Language to specify the pedagogical relationships between learning concepts is difficult to learn.

   1 2 3 4 5

12. The defined Pedagogical Relationship Types are difficult to understand.

   1 2 3 4 5

13. Using the Adaptive Storyline Language to specify the adaptive storyline is difficult to learn.

   1 2 3 4 5

14. 3D Adaptation Strategies are difficult to understand.

   1 2 3 4 5

15. Being able to add a 3D Adaptation Strategy to a topic is very useful.

   1 2 3 4 5

16. Using the Adaptive Topic Language to specify the adaptation of the 3D learning objects during a topic is difficult to learn.

   1 2 3 4 5

17. The defined 3D Adaptation Types are difficult to understand.

1 2 3 4 5

18. Being able to add 3D Adaptation Types to 3D learning concepts in a topic is very useful.

   1 2 3 4 5

C. ISONORM 9241/110-S Evaluation Questionnaire (ISONORM)

In the following, you are kindly requested to evaluate the 3D VLE authoring. Please read the following carefully:

The goal of this evaluation questionnaire is to detect weaknesses in the used modelling languages and to obtain concrete suggestions for improvement.

In order to achieve this, your judgement as author who uses the three modelling languages is of paramount importance. Your evaluation of the proposed modelling languages in question should be based on your personal experience. It is important to remember that this is not an evaluation of you personally. Rather, we are interested in your personal opinion of the adopted modelling languages.

1-Very Low   5=Very High

1. The used modelling languages inappropriately meet the demands of authoring an adaptive 3D VLE enabled course.

   1 2 3 4 5

2. The proposed modelling languages complicate the authoring task due to a complex process.

   1 2 3 4 5

3. The proposed modelling languages are not designed according to a consistent principle.

   1 2 3 4 5

4. The proposed modelling languages require a lot of time to learn.

   1 2 3 4 5

5. The Pedagogical Modelling language requires the memorization of too many issues.

   1 2 3 4 5

6. The Adaptive Storyline Modelling language requires memorization of too many issues.

   1 2 3 4 5

7. The Adaptive Topic Modelling language requires memorization of too many issues.

   1 2 3 4 5

8. The modelling languages force the user to follow an unnecessarily rigid sequence of steps.

1 2 3 4 5

D. Subjective Impression Questionnaire (SIQ)

The following sentences describe thoughts and feelings you may have regarding the use of the 3D VLE Authoring languages. For each of the following statements please indicate how much you can agree on the given scale.

1-Very Low 5=Very High

1. Using the modelling languages is a good idea.
   1 2 3 4 5

2. Using the modelling languages is unpleasant.
   1 2 3 4 5

3. Using the Pedagogical Modelling language is easy for me.
   1 2 3 4 5

4. Using the Adaptive Storyline Modelling language is easy for me.
   1 2 3 4 5
5. Using the Adaptive Topic Modelling language is easy for me.
   1 2 3 4 5
6. It could easy for me to become skilful at using the modelling Languages.
   1 2 3 4 5

7. I find that the modelling languages are useful in authoring adaptive 3D courses.
   1 2 3 4 5

8. Using the modelling languages make it easy to author adaptive 3D courses.
   1 2 3 4 5

9. I find the modelling languages easy to use.
   1 2 3 4 5

10. I find the modelling languages easy to understand.
    1 2 3 4 5

11. Remembering the notations and their use for the modelling languages is easy
    1 2 3 4 5

E. Qualitative feedback:

1. What did you like best about the authoring approach in general and its languages?
   Please write your answer here:

2. What did you like least about the authoring approach in general and its languages?
   Please write your answer here:

3. Was the Pedagogical Model language expressive enough to specify the pedagogical aspects for the adaptive 3D courses? (Please describe why Yes it was or No it was not)
   Please write your answer here:

4. Was the Adaptive Storyline Model language expressive enough to specify the overall storyline of the adaptive 3D course? (Please describe why Yes it was or No it was not)
   Please write your answer here:

5. Was the Adaptive Topic Model language expressive enough to specify the details of each topic? (Please describe why Yes it was or No it was not)
   Please write your answer here:

6. What should be improved and how?
   Please write your answer here:

F. Workload Perception Questionnaire (WPQ)

The purpose of this short questionnaire is to measure the perceived workload while learning and working with the 3D VLE authoring languages (subsequently referred to as 'task'). Please answer the questions below by rating each item based on your subjective impression.

1-Very Low 5=Very High

1. Mental Demand: How mentally demanding was the task?
   1 2 3 4 5

2. Physical Demand: How physically demanding was the task?
   1 2 3 4 5

3. Performance: How successful where you in accomplishing what you were asked to do?
   1 2 3 4 5

4. Effort: How hard did you have to work to accomplish your level of performance?

    1 2 3 4 5

5. Frustration: How insecure, discouraged, irritated, stressed and annoyed were you when doing the task?

    1 2 3 4 5

Thank you for your participation in the 3D VLE Authoring Evaluation.

B. Appendix: Interview Evaluation Questions

B.1 Pedagogical Model Language

The following is the list of questions which are related to pedagogical model language:

1. Can you explain the goal of the Pedagogical Model?
2. Do you understand the concept of Pedagogical Relationship Type? Can you explain this concept in your own words?
3. Can you recall some Pedagogical Relationship Types?
4. Do you understand the predefined Pedagogical Relationship Types? Which one do you understand and which one are more difficult to understand?
5. Given a Pedagogical Model using the PML, is it easy to understand what it expresses, i.e. can you "read" the diagram? Are there parts that you really don't know what they mean?
6. Is the visual language useful or helpful in expressing the pedagogical relationships between the learning concepts? Are there some parts that you put in just because there were in the example? If so, which ones?
7. Which parts seem to be a particularly strange way of doing or describing something?
8. Are the visual notations of the Pedagogical Model Language easy to remember?
9. What are positive/negative aspects of PML? How could it be improved?
10. Is there anything else you would like to discuss about PML?

B.2 Adaptive Storyline Language

The following is a list of questions which are related to the adaptive storyline language:

1. Can you explain the goal of the Adaptive Storyline Model?
2. Do you understand the meaning of the adaptation strategies?
3. Can you recall some adaptation strategies?

4. What do you think about the need to add learning concepts to a topic using the Adaptive Storyline Language?
5. Do you understand how the mechanism of the storyline adaptation rules is working?
6. What do you think about the possibilities of realising adaptations for a storyline by applying adaptation strategies? Would you do it differently?
7. Given an Adaptive Storyline Model using the ASLL, is it easy to understand what it expresses, i.e. can you "read" the diagram? Are there parts that you really don't know what they mean?
8. Which parts seem to be a particularly strange way of doing or describing something?
9. Are the visual notations of the Adaptive Storyline Language easy to remember?
10. Do you find the visual language easy to use? Why (not)? Are there some parts that you put in just because there were in the example? If so, which ones?
11. Is the ASLL small enough, i.e. are there any language features that do not contribute to the goal of the language?
12. What are positive/negative aspects of the Adaptive Storyline Language? How could it be improved?
13. Is there anything else you would like to discuss about ASLL?

B.3 Adaptive Topic Language

The following is the list of questions which are related to the adaptive topic language:

1. Can you explain the goal of the Adaptive Topic Model?
2. Do you understand the meaning of the adaptation types?
3. Do you understand how the mechanism of the topic adaptation rules is working?
4. What do you think about the possibilities of realising adaptations for a topic by using adaptation types and topic adaptation rules? Would you do it differently?
5. Given an Adaptive Topic Model using the Adaptive Topic Language, is it easy to understand what it expresses, i.e. can you "read" the diagram? Are there parts that you really don't know what they mean?

6. Which parts seem to be a particularly strange way of doing or describing something?
7. Are the visual notations of the Adaptive Storyline Language easy to remember?
8. Do you find the visual language easy to use? Why (not)? Are there some parts that you put in just because there were in the example? If so, which ones?
9. Is the Adaptive Topic Language small enough, i.e. are there any language features that do not contribute to the purpose of the language?
10. What are positive/negative aspects of the Adaptive Topic Language? How could it be improved?
11. Is there anything else you would like to discuss about Adaptive Topic Language?

B.4 General

The following is list of questions which are related to the three visual languages in general:

1. Is the similarity of concepts clear from their notation?
    a. Are there visual notations that mean similar things but were different? Please give examples.
    b. Are there places where the notation used was similar with elsewhere but indicates different things? Please give examples.
2. Is it possible to design a large course using the three visual languages? Please explain why yes or no?
3. Do you find that the visual languages are useful in authoring adaptive 3D VLE? Why (not)?
4. Do you think using the visual languages makes it easy or on the contrary difficult to create an adaptive 3D VLE? Why?
5. What kind of things requires the most mental effort?
6. Do some things seem especially complex or difficult to work out in your head (e.g. when combining several things)? What are they?
7. Do you think that the availability of an authoring tool would have influences your answers and opinions?