Competence-based Adaptation of Learning Environments in 3D Space

A. Nussbaumer[1], C. Gütl[2], D. Albert[1], D. Helic[2]
[1] Department of Psychology, University of Graz, Graz, Austria
[2] Institute Information Systems and Computer Media, Graz University of Technology, Austria

***Abstract*—This paper presents a concept how a learning environment can be established in 3D space and how it can be adapted to the competence state of a learner. In contrast to existing Learning Management Systems learning paths are spatially represented in 3D space. In this approach the learner can immerse into a virtual learning landscape consisting of learning objects and is guided by highlighting a path through the landscape. Path creation is based on skills which are assigned to learning objects and which make up the learner model. Principles of the self-regulated learning approach is realised by visualising the learner model in 3D space and by giving the learner freedom for the own learning process. An implementation of this approach is realised in the Second Life virtual world which is connected with a Web service managing the adaptation strategy.**

***Index Terms*—Adaptation, Competence, E-Learning, Virtual Reality.**

I. INTRODUCTION

Adaptation and personalisation have been active and lively areas of research over the last two decades, which have often been reflected in scientific literature (for example Brusilovsky et al., 1996; Conlan et al., 2002; De Bra et al., 1999). The educational context has been one of the most important application domains. Adaptive systems have been created which are capable of adapting their behaviour and output to the learners' needs and preferences. An important system property is the ability to adapt the sequence of the learning material (also called learning path) to the learner's knowledge level.

Learning Management Systems (LMS) are computer systems which mange the learning process on the level of learning material. Usually they allow for creating and storing learning content and they are responsible for presenting content to the learner in a more or less meaningful and intelligent order. Simple and non-adaptive systems like Moodle (Moodle, 2009) enable the content author and teacher to manually sequencing the learning material. More sophisticated systems like AHA! (AHA!, 2009) have implemented strategies to dynamically sequencing the content and to adapt the sequence to the learner profile.

Traditionally, sequencing has been done by presenting one learning object after another to the learner. From a pedagogical point of view this is often criticised, because the learner has no control and overview on the own learning process. In order to overcome these shortcomings the Open Learner Model approach aims at making visible the model which is used by a system to adapt its behaviour (Bull et al., 2008; Nussbaumer et al., 2008). Additionally the visual models can be interactive to enable learners to control their learning process. This approach should stimulate the learners to reflect on their knowledge and learning process. On a more general level this approach is described in the scientific literature in the context of self-regulated learning, which is described in the next section.

This paper presents an approach how a virtual reality environment can be used to present learning material to the learner. In contrast to other systems the learning path is not a logical sequence of learning objects, but the learning objects are arranged in 3D space in order to represent the path in a spatial way. The adaptation strategy is based on Competence-based Knowledge Space Theory (see next section), which adapts the learning path to the competence state of the learner. A detailed description of this approach is given in Section III. Implementation details and system architecture is described in Section IV.

II. THEORETICAL BACKGROUND

A. *Adaptivity and adaptive systems*

The concept of adaptivity has a long tradition in technology-enhanced learning, for example it has been applied in Intelligent Tutoring Systems (ITS) to some extent, user-model-based Adaptive Systems (AS), and Adaptive Hypermedia Systems (AHS) (Brusilovsky, 2000). Following the discussion in (Brusilovsky, 1996; De Bra et al., 2004), users (learners) differ in terms of (learning) goals, pre-knowledge, individual traits and needs, as well as pedagogical parameters. Based on these characteristics adaptive presentation (adaptation on the content level) and adaptive navigation support (direct guidance, adaptive ordering, hiding, and annotation of links) are the most important features which can be provided by an adaptive system. Domain models and user models are defined in order to specify relationships between users and content, which forms the basis for the adaptation functionality. In educational applications these relationships typically represent the knowledge about learners and content. Furthermore, adaptive systems usually contain adaptation models which determine the adaptation strategy of those systems. In this way an adaptive system can help the learner to navigate through a course by providing user-specific paths.

B. Competence-based Knowledge Space Theory (CbKST)

Knowledge Space Theory (KST) and its competence-based extensions (CbKST) are prominent examples how an adaptation strategy can be grounded on a theoretical framework (Hockemeyer, 2003). KST constitutes a sound psychological mathematical framework for both structuring knowledge domains and for representing the knowledge of learners. Due to (psychological) dependencies between problems prerequisite relations can be established. The knowledge state of a learner is identified with the subset of all problems this learner is capable of solving. By associating assessment problems with learning objects, a structure on learning objects can be established, which constitutes the basis for meaningful learning paths adapted to the learners knowledge state. Competence-based Knowledge Space Theory (CbKST) incorporates psychological assumptions on underlying skills and competencies that are required for solving the problems under consideration. This approach assigns to each problem a collection of skills which are needed to solve this problem and to each learning objects those skills which are taught. Similar to the knowledge state a competence state can be defined which consists of a set of skills which the learner has available. Furthermore, there may also be prerequisite relationships between skills. CbKST provides algorithms for efficient adaptive assessment to determine the learner's current knowledge and competence state, which builds the basis for personalization purposes. Based on this learner information, personalised learning paths can be created.

C. Self-regulated Learning

Self-regulated learning has become increasingly important in educational and psychological research. Compared to adaptive learning systems, the tenor in self-regulated learning is to give the learner greater responsibility and control over all aspects of (technology-enhanced) learning. There are only few attempts trying to build a complete model of self-regulated learning (Puustinen & Pulkkinen, 2001). Most of these models deal with self-regulation as a process that involves goal setting and planning, monitoring and control processes, as well as reflection and evaluation processes. From this it becomes apparent that self-regulation is closely related to meta-cognitive strategies. In (Dabbagh & Kitsantas, 2004) six self-regulatory processes and their significance to Web-based learning tools have been identified. For example (a) goal setting is supported by communication tools, such as e-mail communication with a tutor, (b) the use of task strategies is supported by content delivery tools, such as concept mapping software to organise course content, (c) self-monitoring is supported by use archived discussion forums, (d) self-evaluating is supported by the use of rubrics, evaluation criteria, and peer feedback, (e) time planning and management is supported by communication tools concerning time budgeting, and (f) help seeking is supported by hypermedia tools.

D. Information and knowledge visualisation

The abilities of humans to recognise visual information are highly developed. Patterns, colours, shapes and textures can rapidly and without any difficulty be detected. On the other hand, the perception of text-based content is much more effort than the perception of visual information (Shneiderman, 1996). Information visualisation is the transformation of abstract data and information into a form that can be recognised and understood by humans. In this sense, information visualisation can be seen as an interface to abstract information spaces. So exploring large volumes of data can be done effectively by humans.

Information visualisation techniques are widely used in Web-based social software (e.g. graph visualisation is used to outline online community networks and tag clouds are often used to provide overview on collaboratively tagged Web content) and especially in knowledge management (e.g. visualisation of large knowledge structures for providing overview and interface to it). In contrast to these application areas, information visualisation is barely used in e-learning applications.

An emerging field of applying information and knowledge visualisation is the area of Open Learner Models. These models are visually outlined and presented to the learner in order to provide the possibility of inspection them. Learners might want to know about the basis for the calculations of the system. According to (Bull et al., 2008) opening up these models might increase self-reflection and motivation of the learners.

III. CONCEPTUAL APPROACH

In this section an approach is described on a conceptual level how a virtual learning landscape can be created in 3D space, which is capable of automatically adapting to the learner and of providing guidance and feedback for the learner.

A. The educational perspective

Basically the learning material is created in 3D space by content authors. They create and place learning objects in a 3D landscape, whereby learning objects are more or less interactive 3D models which convey specific knowledge chunks. It is up to the content author how sophisticated the learning objects are designed. They can consist of simple text documents, contain images and diagrams, or they can also contain movies which are played. Furthermore learning objects can also be complex 3D models which represent three-dimensional information, for example molecules. The learner (represented as avatar) can walk through this landscape and make use of the several learning objects. In this way a learning landscape emerges which, however, is still static and has no adaptive features.

Positioning of learning objects in the virtual world is done by the content author manually. This is important because the spatial position may also contain information with respect to content. For example, the solar system domain may be structured in the way that the planets are learning objects and the spatial position represents the real positions of the planets. For this reason positioning must not be done dynamically by the system, though in some cases this would be a possibility to express the learning path. Obviously it is a requirement for the virtual world that a content author can create 3D objects and freely move and position them in 3D space.

Learning paths are an important property of technology-enhanced learning systems. As described in Section 2, Competence-based Knowledge Space Theory (CbKST) provides psychologically sound methods to create meaningful learning paths. In traditional learning systems they are realised as a sequence of learning objects, whereby the learner has a next-option to move forward. In 3D space there is a different situation, since learning objects are permanently positioned and the learner walks through the space. Therefore methods are needed which indicate the path through the learning landscape. For example, objects can be highlighted by pointing a spot light to an object which should be learned next. In computer graphics light sources are well known concepts in 3D worlds which most of them have implemented. Another method would be to highlight a learning object by adding a marker object to each learning objects which can change the colour to indicate that this learning object is the next one.

Realising a learning path is done by successively highlighting one learning object after another one. This mechanism bears also the possibility to highlight more objects at the same time if they are equal (at the same level) regarding the logical sequence. In this way the learner can freely choose between them, which gives more control to the learner. A further level regarding self-regulated learning is obviously given by the fact that the learner can freely move in 3D space and can deal with any learning object in the learning landscape independent of the highlighting state. Obviously this is a very natural way of providing self-regulated learning possibility, since it comes from the general system design and is not an explicitly created system feature.

In order to implement an adaptive strategy and to arrange learning objects accordingly a knowledge representation model is needed. Basically a domain has to be defined which comprises a subject matter at an appropriate size. For example the Pythagorean Theorem can be a manageable domain for pupils. Learning objects are the components which teach the domain and which the learners interact with. Furthermore, skills are defined which describe knowledge of learners on a cognitive level. Skills are related to learning objects in the way that learning objects are teaching specific skills. Furthermore, following Competence-based Knowledge Space Theory, skills are structured through prerequisite relations between skills meaning that certain skills should be learned before other skills.

A further element of the conceptual design is a feedback object which gives information to the learner about the learned skills. As pointed out above, by using a domain model for content structuring, skills are defined including prerequisite relations between them. The skills of a domain and their prerequisite structure are represented in 3D space as a 3D skill structure model. If a learning object has been done than specific skills have been taught by this learning object. These skills can be highlighted in the skills structure model by changing the colour of the respective skills. For example learned skills can be green, the other skills can be grey.

This approach follows the idea of opening up user model to the learner, which is supposed to stimulate self-reflection and motivation (see Section 2). However, instead of presenting this information as list or 2D diagram, in a virtual world the structure can be represented as 3D model. The skill structure model is visible all the time, as long as the learner is not too far away from it. No extra window is needed for the presentation of this information.

The described design also bears possibilities for collaborative learning, if the used virtual world provides communication features between different persons. Instead of a single learner, the walk through the learning landscape can be done by a group of learners. Then they can talk about the subject matter, their difficulties of understanding, and what they actually do understand. Furthermore, a tutor can accompany a learner through the learning landscape for the reason of direct communication and help.

B. *The technical perspective*

Following the design described in (Nussbaumer et al., 2007), the overall design of the system is split into two parts. First the virtual world with a 3D interface contains the learning objects (learning content) and is used by the learners to interact with the system. Second the logical part (CbKST Web Service) stores domain model and user model and is responsible for the adaptation strategy. For the sake of flexibility these two parts are separated into two systems which communicate with each other over the Internet using SOAP protocol (see Figure 1).

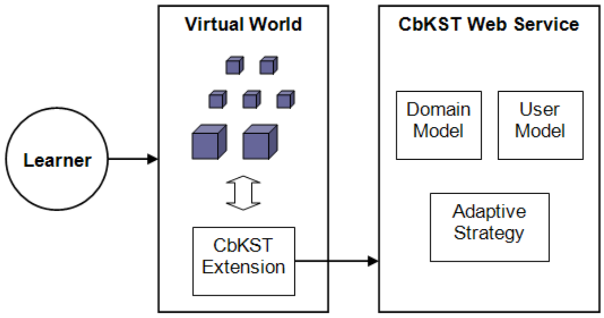

Figure 1. The overall system architecture. The learner interacts with the virtual world and an extension inside the virtual world connects to the CbKST Web Service for providing adaptation, guidance, and feedback.

As outlined in Figure 1, an extension to the virtual world has to be implemented. On the one hand this extension is connected to the learning objects and skill structure object and controls the behaviour and state of them. On the other hand it is also connected to the CbKST Web Service where it gets the information how it should control the learning landscape. Obviously the virtual world must allow integrating program code which can also make connections over the Internet to Web Services.

If a learner has done a learning object, then this has to be indicated by clicking on the marker object attached on the learning object. The marker object changes its state (colour) and sends a message to the CbKST extension that it has been clicked. The CbKST extension passes this information to the CbKST Web Service and retrieves a message which learning objects should be done next. After that the CbKST extension initiates that the respective learning objects (or attached markers) change their state to indicate that the learner should continue with them.

In addition to indicating the next learning objects the skill structure object also has to be updated. After a learning object has been done, the CbKST extension also gets the current skill state and sends this information to the skill structure object which changes its appearance according to the current skill state.

These two communication flows outline that a communication infrastructure is needed. Object in the virtual world have communicate with the control module (CbKST extention) and the control module has to communicate with the CbKST Web Service.

In order to achieve adaptation, guidance, and feedback, an adaptation module is needed, which is designed as Web Service. In this place the algorithms for creating learning paths and for calculating the current skill state are implemented. Furthermore domain and user model are located there which the algorithms need for their calculations. The separation of this component form the virtual world brings independence from the implementation inside the virtual world. In this way more flexibility can be achieved, since the Web Service only has to be developed once, even if different virtual worlds are employed.

IV. IMPLEMENTATION

For the implementation the popular Second Life (Second Life, 2009) has been chosen as virtual world. Compared to other similar virtual worlds, such as (Sun Wonderland, 2009), Second Life has reached a reasonable technical maturity level, since development already has been ongoing for ten years. However, every virtual world can be used for the implementation if the technical requirements are fulfilled.

The logical part which is responsible for managing the knowledge representation model, for managing the user model based on the knowledge representation model and for generating the learning paths on a logical level is implemented as a Web Service in a Tomcat environment. It can be contacted via standard Web Service interface (SOAP, XML-RPC).

The most important requirements for the virtual world are the possibility of authoring the 3D objects and manually positioning them. Furthermore, it must be able to connect to a Web Service via SOAP or XML-RPC in order to get information regarding learning path and user model. Next it must be possible to add small programs (scripts) to 3D objects which can do control tasks, such as starting learning path service or connecting to Web Service. Finally, the virtual world must offer the possibility to dynamically change properties of 3D objects, such as changing the colour of an object.

Second Life offers all these requirements sufficiently. Objects can be easily created without experience in CAD. A simple interface allows the author to create new objects, to change properties, and to add textures. Scripts can be added to each object, which controls their behaviour. A simple scripting language is used which can be learned if an author has little programming skills. This language provides a function to access a Web Service via XML-RPC and gets the result of this call as an event for further processing.

A prototype of the implementation has nearly been finished, which follows the conceptual approach described in Section 3. Figure 2 and Figure 3 present a screenshot of a setting in Second Life with six learning objects and a skill structure object with nine skills. In this example, learning objects are realised as simple text panels, however they also could be more complex models. Each learning object has a marker at the bottom which shows the current state of the learning object. There are three possible states, green means that the learning objects has been done by the learner, orange means that this object should be done next, and grey means that this object should be done later. The skill structure model shows the skills for this example domain in a prerequisite structure. Skill can have two states, green spheres are already acquired skills and grey spheres are not acquired skills. The learner who walks through the learning landscape is represented as the avatar in the front (Figure 2) and in the back (Figure 3).

Figure 2. A learning landscape with six learning objects and a skill structure model at the beginning. The orange marker of one learning object indicates that the learner should do this object first.

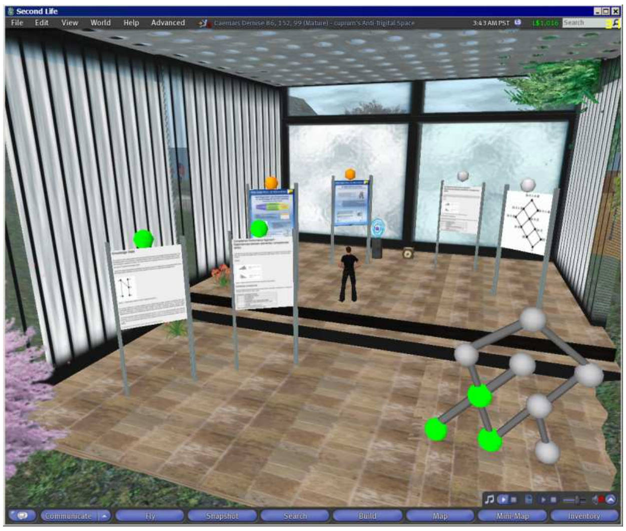

Figure 3. A learning landscape with six learning objects and a skill structure model after the learner has already done two learning objects (green markers). The orange markers indicate that these learning objects are appropriate to do next. The skill structure model shows the acquired skills (green spheres).

Figure 2 shows the situation at the beginning of the learning process. One of the learning objects has an orange marker, which indicates that the learner should do that one. In the skill structure model, no skill is marked as acquired. Figure 3 shows the situation after the learner has done two learning objects (objects with green markers). Two learning objects have orange markers, which indicates that one of them should be done next. The learning is free to choose in this situation. The skill structure model shows that three skills are acquired - obviously these skills are conveyed by the two learning objects with the green markers.

Each of the markers and the skills are attached with scripts. These scripts have the duty to control the appearance of the marker, to handle the interaction with the user, and to do the communication with the main control element. If a learner has finished a learning object, then the respective marker has to be touched. The script of the marker changes the colour from grey to green and reports to the main control element that this learning object has been done. The main control element (the CbKST extension) does the communication with the CbKST Web Service and controls the markers and skill structure model. After a learning object has been done, the main control element contacts the CbKST Web Service and gets information about the updated skill state and which learning object should be done next. Then this control element send messages to the markers of the involved learning objects and a message to skill structure element to update its state.

V. CONCLUSION AND OUTLOOK

In this paper a novel approach has been presented how adaptation of learning paths can be realised in virtual worlds. The adaptation strategy is realised in an external Web Service which controls the adaptation behaviour of a virtual world. Theoretical basis for the adaptation is the psychological sound Competence-based Knowledge Space Theory which already has been applied in traditional learning systems several times. Combining CbKST with virtual reality provides new possibilities regarding self-regulated learning, since the leaner can use the inherent properties of the 3D world by freely moving around in 3D space. A learning path is offered to the learner, the does not restrict the learner to the given sequence.

Future development will concentrate on pre- and post-assessment. If pre-assessment is conducted than the learning path can be adapted to the pre-knowledge of the learner. A post-assessment reveals the actual knowledge of the learner and can indicate which learning objects eventually should be processed again. Both types of assessment deliver sets of skills the learner has available, which can be visualised on the 3D skill structure object in different colours. For example, skills which have been taught, but are not available in the post-assessment be visualised in red colour.

There is a restriction in the usage of a learning landscape. Since highlighting of learning objects markers and skills can only be done once at a time, a learning landscape can be used by only one learner at the same time. A control object is needed where the learner can start the learning path service and stop it. As soon as the learning path service has been activated by a specific learner, than it is blocked for all other learners. However, this does not force other persons to be kept out from the learning landscape unless they do not interact with the system. A possible solution can be to copy the whole learning landscape and put it on a different place.

Privacy is another issue which has to be thought of. The skills structure element shows the current skill state of the learner, however, everybody who is in the vicinity can see this. Not everybody likes to be inspected by others. So some considerations have to been made about that.

REFERENCES

[1] AHA! (2008). http://aha.win.tue.nl/, retrieved on February 15, 2009

[2] Brusilovsky, P. (1996). Adaptive Hypermedia, an Attempt to Analyze and Generalize, In P. Brusilovsky, P. Kommers, & N. Streitz (Eds.), Multimedia, Hypermedia, and Virtual Reality (Lecture Notes in Computer Science, Vol. 1077). Berlin: Springer-Verlag, 288-304.

[3] Brusilovsky, P. (2000). Adaptive hypermedia: From intelligent tutoring systems to Web-based education (Invited talk), In: G. Gauthier, C. Frasson and K. VanLehn (eds.) Intelligent Tutoring Systems. Lecture Notes in Computer Science, Vol. 1839, pp. 1-7, Berlin: Springer Verlag.

[4] Bull, S., & Kay, J. (2008). Metacognition and Open Learner Models. In: The 3rd Workshop on Meta-Cognition and Self-Regulated Learning in Educational Technologies, at ITS2008.

[5] Conlan, O., Hockemeyer, C., Wade, V., & Albert, D. (2002). Metadata driven approaches to facilitate adaptivity in personalized eLearning systems. Journal of Information and Systems in Education, 1, 38-44.

[6] De Bra, P., Brusilovsky, P., and Houben, G. (1999). Adaptive hypermedia: from systems to framework. ACM Computing Surveys, 31.

[7] De Bra, P., Aroyo, L., Cristea, A. (2004). Adaptive Web-based Educational Hypermedia, In: Mark Levene, Alexandra Poulovassilis (Eds.) Web Dynamics, Adaptive to Change in Content, Size, Topology and Use, pp. 387-410, Springer-Verlag.

[8] Dabbagh, N., & Kitsantas, A (2004). Supporting Self-Regulation in Student-Centered Web-Based Learning Environments, In: International Journal on E-Learning, 3(1), (2004), pp. 40-47.

[9] Hockemeyer, C. (2003). Competence Based Adaptive E-Learning in Dynamic Domains, In: F. W. Hesse & Y. Tamura (Eds.), The Joint Workshop of Cognition and Learning through Media-Communication for Advanced E-Learning (JWCL 2003) , pp. 79-82, Berlin.

[10] Moodle (2008). http://moodle.org/, retrieved on February 15, 2009

[11] Nussbaumer, A., Gütl, C., & Albert, D. (2007). Towards a Web Service for Competence-based Learning and Testing. Proceedings of the World Conference on Educational Multimedia, Hypermedia & Telecommunications (ED-MEDIA 2007), 25-29 June 2007, Vancouver, Canada.

[12] Nussbaumer, A., Steiner, C, and Albert, D. (2008). Visualisation Tools for Supporting Self-Regulated Learning through Exploiting Competence Structures. Proceedings of the International Conference on Knowledge Management (IKNOW 2008), 3-5 September 2008, Graz, Austria.

[13] Puustinen, M. & Pulkkinen, L. (2001). Models of self-regulated learning: a review, In: Scandinavian Journal of Educational Research, 45(3), pp. 269-286.

[14] Shneiderman, B. (1996). The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations, In: Proceedings of the IEEE Symposium on Visual Languages (VL 1996), pp. 336–343.

[15] Second Life (2009). http://www.secondlife.com/, retrieved on February 15, 2009

[16] Sun Wonderland (2009). https://lg3d-wonderland.dev.java.net/, retrieved on February 15, 2009

[17] Zimmerman, B.J. (2002). Becoming a Self-regulated Learner: An Overview, In: Theory into Practice, 41(2), Springer-Verlag.

AUTHORS

A. Nussbaumer is member of the Department of Psychology (Cognitive Science Section) at the University of Graz, Austria (email: alexander.nussbaumer@uni-graz.at).

C. Gütl is member of the Institute of Information Systems and Computer Media (IICM) at the Graz University of Technology, Graz, Austria, and member of the School of Information Systems, Curtin Univerity of Technology, Perth, WA (e-mail: cguetl@iicm.tugraz.at).

D. Albert is full professor at the Department of Psychology (Cognitive Science Section) at the University of Graz, Austria (email: dietrich.albert@uni-graz.at).

D. Helic is member of the Institute of Information Systems and Computer Media (IICM) at the Graz University of Technology, Graz, Austria (e-mail: dhelic@iicm.tugraz.at).

The work presented in this paper is partially supported by European Community under the Information Society Technologies (IST) program of the 7th FP for RTD - project GRAPPLE. The authors are solely responsible for the content of this paper. It does not represent the opinion of the European Community, and the European Community is not responsible for any use that might be made of data appearing therein.

Published as submitted by the author(s).