Inside Coherence: Concept for a Non-Linear eLearning Solution

O.Schneider[1,2,3], U.Bleimann[1,2], A.D.Phippen[3] and B.Harriehausen-Mühlbauer[2]

[1]Centre for Advanced Learning, Media and Simulation (igdv),
[2]Institute for Applied Informatics Darmstadt (aiDa),
University of Applied Sciences Darmstadt, Germany
[3]Centre for Information Security and Network Research, University of Plymouth, United Kingdom
e-mail: OliFFM@web.de

Abstract

This paper describes a system that selects and presents non-linear learning content and shows how such a system has to be designed. Hence, this paper describes what functionality the non-linear eLearning system must provide. An Interactive Digital Storytelling based approach has been chosen. It is shown how this concept has been adapted and extended for the purpose of non-linear Extended Blended Learning. Furthermore a general concept for an extremely adaptable engine has been designed, implemented and tested.

Keywords

eLearning, non-linear extended blended learning, interactive digital storytelling

1. Introduction

As the successor of Blended Learning, Extended Blended Learning overcomes the pure eLearning obstacles by integrating eLearning with face-to-face learning and project-based learning (Bleimann and Röll, 2006). The first project realising this approach is Atlantis University (Bleimann, 2004). Within the scope of this project a system has been designed that provides methods for presenting non-linear content.

Such a system should present non-linear content in coherent way. To fit into the Extended Blended Learning environment it also should be user-adaptive and independent of any content (Schneider *et al.* 2006). The system should be able to provide coherence, so that module content is not lined up pointlessly or contradictory. For example a student should not be able to complete an assessment for object-oriented content prior to encountering and learning the corresponding material. In any case, consistency must be guaranteed. The module must be able to lead to a successful end in any situation and with any chosen path, except in the event of failed exams or missed necessary exercises.

The learner's personal learning types (Brückner, 2005) should decisively influence the path through the module's interactive content. Also conceivable are content-sections that question knowledge. All these results should influence the module. The eLearning environment should work completely regardless of any concrete content. This signifies that it is not only independent from the media type, whether the content is a video or a set of slides, but also that the system should be usable for each kind of learning content and learning styles.

2. Related Work

Most of the present learning environments (Baumgartner *et al*. 2002, Niegemann *et al*. 2004) such as Blackboard (Blackboard Inc., 2008) and TopClass (WBT Systems Ltd., 2008) do not support adapting content to the users' actions, behaviours or learning styles (van Rosmalen and Boticario, 2005). Systems like ELAT (Gojny, 2003), Moodle (Gertsch, 2007) and, ILIAS (Ilias, 2008) that are based on newer specifications such as SCORM 2004 (Alexandria ADL Co-Laboratory, 2008), EML (LearningNetworks, 2008) and its successor Learning-Design (Koper and Tattersall, 2005) fulfil the demands of interaction and adaptation, but neither support time management nor provide the means to author more complex content, because these features are based on branching (Schneider *et al.* 2006) in this systems.

A modified storytelling-system fits best to the demands of Extended Blended Learning (Schneider *et al.* 2006); for this research, the *StoryEngine* based on Propp's story model (Grasbon, 2001) is most suitable for Atlantis University (Schneider *et al.* 2007).

3. An Interactive-Storytelling Approach

Propp analysed Russian fairy tales and could verify a subset of a total of 31 action functions in every story (Propp, 1968). These functions are arranged in a generally static order, but may be repeated individually or in groups (Grasbon, 2001 p 63). Several variations (scenes) of a function can exist and all these functions are linked by dependencies (Grasbon, 2001 p 64).

The principle of the summarisation of scenes into functions and relations, to repeat and to apply conditions, can be transferred to the needs of Extended Blended Learning. However, Propp's functions do not fit directly to the Extended Blended Learning scenario.

Because of the mixture between the three pillars, the execution of various Extended Blended Learning modules differ. Thus, one single model with firmly prescribed functions is not adaptable enough to describe those variations. The StoryEngine (Grasbon, 2001) must be advanced in this respect so that it can process different models and that each of these models is able to define its own functions.

The Propp-model is originated in the fairy-tale world and therefore did not consider interaction (Grasbon, 2001 p 70). The functions are not polymorphic; all variations always return the same result. Grasbon has extended the Propp-model by splitting some functions into two part-functions, so that interaction becomes possible. Within the scope of Extended Blended Learning all functions should be polymorphic, so that they can influence the action.

4. Components and Design of a Corresponding System

In spite of the necessary extensions to the model of Grasbon's engine its top level design can be presented. Figure 1 shows the design of this engine.

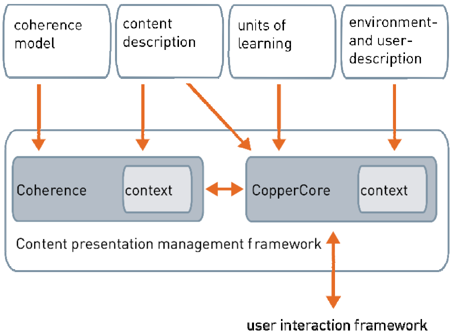

Figure 1: System's overall architecture

The core is the so-called Coherence-Engine that corresponds to Grasbon's StoryEngine. In order to operate, this engine needs the data of the top-level modules. It also may manipulate this data to change, for example, the contexts. It works closely with CopperCore (Martens and Vogten, 2005), which is used as the content-engine and therefore is the interface to the user interaction framework.

4.1. Scenes vs. eLearning Content-Sections

Many characteristic features of the scenes of interactive stories are transferable to this context. In the Atlantis University Project a scene can be compared with a learning content-section or a learning unit. For example, the subjects Addition and Subtraction could correspond to a scene within a mathematics-story. Scenes are small parts of the story, but are coherent and are self-contained, so the scene's unity is guaranteed (Field, 1992). The temporal unity should arise by the organisation of the subjects in scenes, so that a subject can be graduated by a session (e.g., lesson) and is not interrupted by other subjects.

In this respect the local unity is interesting because the concept of Extended Blended Learning intentionally jumps between the learning styles. A content-section should refer only to a subject within a learning style accordingly to the learning content-section's corresponding characteristics. Even if addition is explained within an eLearning-scene followed by a face-to-face lecture also explaining the top addition, they are still different scenes.

While not directly applicable to the concept of the story, all scenes of Atlantis University should be dramatic scenes, and should propel the content by either itself or by user's interaction (Field 1992). For this purpose every scene should be able to find out at least whether it was finished successfully or not.

Because this paper is not about storytelling but learning environments, scenes are called *content-sections*.

4.2. Content-Types

To be able to automatically tie single scenes with each other, they are categorised like Propp's concept of functions. In the scope of Atlantis University the user can define these functions and because the functions typify scenes and design relations between the groups of content-sections, they are called *content-types*. Figure 2 shows some examples of content-types.

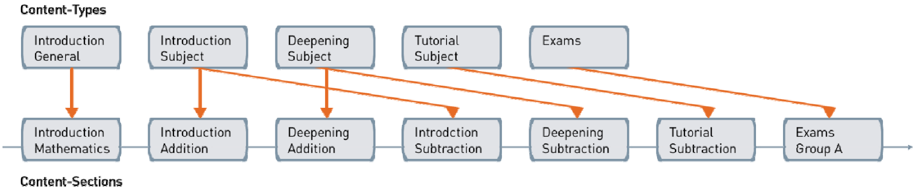

Figure 2: Mathematics module composed of content-types

For example, the content-type *Introduction General* characterises the content-sections that are needed at start of a lecture. The following three content-types refer in special subjects within the lecture. The content-type *Deepening Subject* could characterise content-sections for deepening addition and deepening subtraction. The example lecture repeats the content-types, but not the content-sections. The content-type *Exams* at the end characterises different assessment sheets as content-sections to avoid plagiarism.

The example already points clearly why some content-types have to be repeatable and why relations can exist between them. If content-types are defined for all subjects and only have a linear relationship, all non-linearity would be lost and offer no benefit over the usual eLearning-system approach. It would also be rather pointless to begin with an introduction about addition, then to continue with deepening subtraction and to carry on with a practise for multiplication. The deepening of a subject should always follow an introduction on the same subject.

4.3. Contexts

Contexts provide coherences and dependencies between the content-sections. They serve as variables for the communication between content-sections, module-contents and systems. The engine differs between three different types:

- **Private** contexts are set while presenting content-sections exclusively by the coherence-engine. Having completely presented a content sequence, these kinds of contexts are no longer valid. Their purpose is tying together the single content-sections coherently while presenting them.

- **Public** contexts are likewise set and maintained by the engine. However, this type of contexts can be used by each of the learning-environment's content.

- **External** contexts can be used by the engine as well as by any components of the overall system.

Contexts can be organised in groups that may be further nested in sub-groups. Thus, a context tree can be developed. Several contexts can be checked in an interrelated manner by calling a context-group.

In contrast to usual computer languages every context has two values stretching a value range. For example, the context *acceptedAge* could be set with the minimum value *18* and the maximum value *30*. The content-sections needing these contexts likewise define a value range that is compared to the set range. The comparison delivers a standardised return value between *0* and *1* that is multiplied by a weighting value that is likewise defined in the context query. This value's range is defined between *-1* and *1* and serves to evaluate the result of the corresponding query. The effect reaches fluently from "must not" (*-1*) through "should not" and "should" until "must" (*1*). Hence, the limit values *-1* and *1* signify compelling requirements.

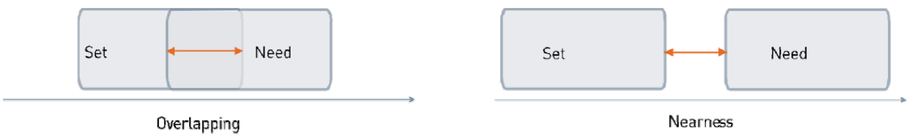

Figure 3: Context comparison modes

The concept provides two different query-modes, illustrated Figure 3. The first mode is overlapping. In this mode the system investigates how much the set context and the needed area overlap. Using overlapping the result is more than *0* as soon as the set context overlaps with the needed one. *1* is returned only if both value ranges are identical. For example, a student could state that he likes to be presented with learning content-sections that are designed for students aged between 20 and 30 years. If this context is defined to 18 – 30, the comparison result is a value near to *1*. However, if the learner-type should be checked, the second query mode is used. In this case the nearness between the set and the requested values is checked. *1* is returned if the values touch. With increasing distance the value becomes smaller and smaller and approaches *0*.

4.4. Coherence-Model

To order the content-types in a sensible sequence and to present coherent stories, the StoryEngine needs a story-model that defines the story-process (Grasbon, 2001 p 62). In the introduced concept this is the coherence-model's task. An XML-dialect is used to put content-types in orders and loops. The model is not related to the content; it only defines the process in relation to the content-types. Arbitrarily many modules can define content-sections for the content-types of a model. For example, a model called *standardLecture-linear* could be deposited in the system, so that every lecturer can provide his own content based on this model.

A model orders weighted content-types coarsely. The weighting defines the necessity of the presentation with a span from *-1* to *1*. Only the limit values force or deny a presentation. If the weight of a context-type is smaller than *1* (for example *0.99)*, paths through a model lacking this content-type would be also possible. However, a rating algorithm would value these paths rather badly, because the necessity of *0.99* is still very high.

Loops iterate their enclosed content-types, until one of the following three exit conditions is fulfilled:

- Within the loop a context has been set that is used as exit condition.

- The content-types within the loop have no more content-sections that can be appended to a sensible path. This can happen either because all available content-sections have been already presented, or because every remaining content-section cannot be presented due to the lack of fulfilled preconditions. If the loop exits because of this condition, it looks for an exit-content-section that must be defined in the loop condition, and presents it.

- Presenting any further content would make the total duration too long, so the coherence-model could not be presented completely. Also in this case an exit content-type must be determined.

To define dependence between two content-types, a *dependencies* block is used in the tail of the coherence-model. Again, the limit values *-1* and *1* are exclusive criteria. A dependency weighted *-1* means that a content-type must not have been presented as a precondition for presenting the dependent type.

4.5. Content-Description

Whereas the coherence-model regulates only the general processing of content-types and can therefore be used for many course-modules, the content-description describes the real content of the respective modules. Therefore, every module needs a content-description that states which content-sections may be presented in a lecture, how important the single content-sections are, which conditions are tied to present a concrete content-section and which conditions are created by it. Also a minimum, a maximum and an average presentation duration is defined. This is used by the coherence-system to determine the total duration and take over the time management.

Although the content-description documents refer to concrete content, for example a small set of slides or a video, they contain only meta-data relevant to the coherence-system. For example, it may only describe that the content-section *DeepeningAddition* takes between 30 and 90 minutes and that before this content-section an *IntroductionAddition* should have been already presented. In this manner the coherence-system remains independently of real content.

This context description references all needed content-types of the model and fills them with content-sections. Each of these content-sections represents a concrete learning content – a video, a set of slide or the similar. If the coherence-model wants to present a content-section of the content-type *Introduction*, it is searched in the content-description.

Each content-section is weighted within a range of *-1* to *1*, so that the coherence-system can recognise the importance of a content-section: The greater the value, the more important is the content-section. The duration a content-section takes is also given minimally, at most and on an average. With the help of these values the coherence-system can carry out the time management

Internal dependencies are needed for a sensible content. It is not a matter only of simply lining up content-sections out of predefined content-types, it must be also possible that dependencies can be defined between the content-sections. Therefore contexts can be set if the content-section has been completed successfully and defined as needed as a prerequisite for presenting a content-section. Indeed, context-conditions are not compelling and not binary. Relations can be also defined by corresponding weightings that could be described best as "would be nice if "or" the closer to the desired value the better.

5. Learning Path Construction

The interaction of the components introduced in the previous section allows the production of paths through the content-space. Originating from the meta-data all possible content-sections for a module are selected and combined to all possible paths corresponding to the coherence-model.

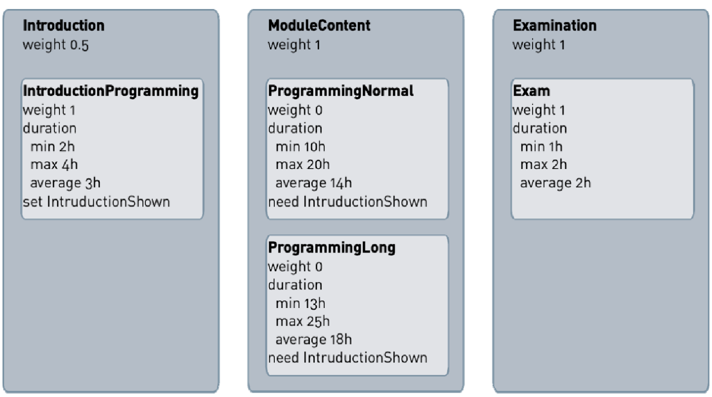

Figure 4: Simple model with content-sections

Figure 4 sketches a simple *Content-Description* with few content-sections and the corresponding *CoherenceModel*.

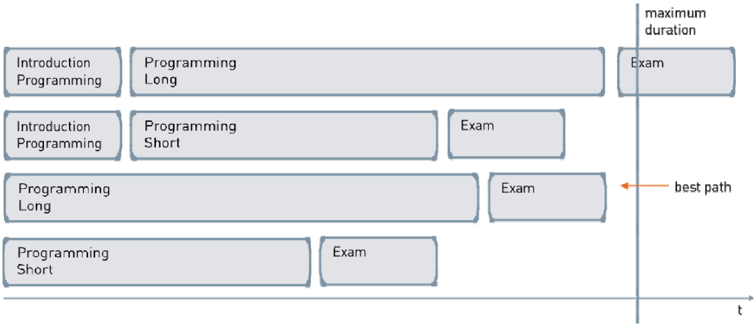

Figure 5: Best fitting path (duration)

Figure 5 illustrates the four possible paths. At first there originate two paths because of the two different content-sections of the type *ModuleContent*. As the model defines a weight of *1* for the content-types *ModuleContent* and *Examination*, both can be considered to be inevitable and therefore have to be combined in all paths and in this order. The *Introduction* is optional because it is weighted *0.5*. Thereby two alternative paths arise beside the first two paths. These additional paths lack the introduction.

The vertical line marks the time limit and shows that the first path retires because it is simply too long. Not all content-sections can be presented within the maximum possible duration. The remaining three paths are within the duration, hence they are theoretically presentable. The best possible one must be selected out of these.

For the example in Figure the defined context *IntroductionShown* is ignored. Because neither model nor content-description define or require contexts, the time is the only evaluation criterion for the choice of the best path in this example. The duration should be used optimally, so the system chooses the path that comes up very near to the at most available duration without being longer. Hence, the third path is selected in the figure.

If contexts are defined in the coherence-model, the content-description or both, time is not the only factor to determine the choice of a path. Instead, content-sections can define which contexts have to be set within the path, so that they can be presented. Therefore, the generator will reject paths that have not set these contexts to the appropriate weighting.

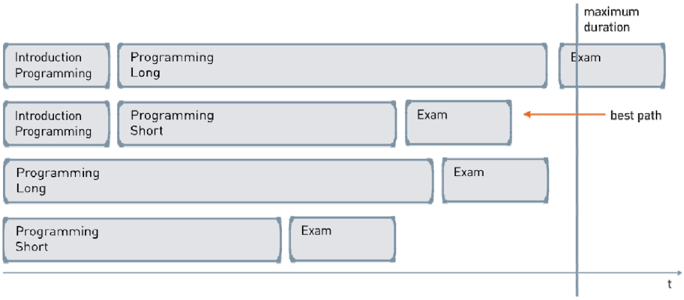

Figure 6: Best fitting path (context)

Figure 6 shows the same coherence-model as the preceding figure. Indeed, this time the defined contexts are taken into account. The *Introduction* sets the context *introductionShown* if it has been presented successfully. Presenting the *moduleContent* content-sections requires that the context *introductionShown* is set, because it is marked as needed.

Although the model permits further paths by its *contentType-w*eights lacking in the content-section *IntroductionProgramming*, the construction algorithm finds out that the paths three and four are not allowed, because the context-conditions are not fulfilled for certain content-sections. Therefore, only path one and two remain as valid paths. Because the first path needs more time than allowed, the second path is selected as the best one: it remains within the time limits and it fulfils all context-conditions.

6. Conclusion and Outlook

This paper describes a multi-level approach to present non-linear content that avoids the obstacles of other approaches such as *branching* and *string of pearls* (Schneider, 2006). The implementation has been tested successfully with several simple test models and different content-descriptions; hence it fulfils the demands of presenting non-linear content from a technical point of view.

However, the tests have also shown that this realisation has serious performance implications. Even simple models result in billions possible paths. For example, if a lecturer provides fifteen different learning content-sections for each learner-type, this results in up to 6^{15} (470.184.984.576) possible paths. Hence, further research is needed to develop a path-algorithm that reduces the number of the paths to be valued drastically. For example, limiting the search deepness severely could mitigate the situation. The algorithm would have to give up putatively bad paths quite early, even if as a consequence good paths may be lost. Another approach would be an algorithm that is based on depth-first search and only saves the best-rated path instead of all possible paths. Similar applications, such as chess computer or navigation systems, could deliver determining tips.

The application is prepared for integration into the Atlantis University Platform by using web-services. But a Portlet has still to be developed as well as the real connections to the other services (that are also under construction at the time of writing).

For creating real content of course an authoring environment is needed, because lecturers will not feel comfortable in writing XML-code. Furthermore an evaluation for lecturers and students with real models and real content is an essential development.

7. References

Alexandria ADL Co-Laboratory (2005). “Sharable Content Object Reference Model SCORM”. http://www.adlnet.org/scorm. (Accessed 28th January 2008).

Baumgartner, P., Häfele, H., and Maier-Häfele, K. (2002), *E-Learing Praxishandbuch – Auswahl von Lernplattformen*, Studienverlag, Innsbruck, Wien, München, Bozen, Germany

Blackboard Inc. (2008). "Blackboard » Educate. Innovate. Everywhere". http://www .blackboard.com. (Accessed 28th January 2008).

Bleimann, U. (2004), "Atlantis University – A New Pedagogical Approach beyond E-Learning", *Campus-Wide Information Systems*, 21(5) pages 191–195.

Bleimann, U. and Röll, F. J. (2006), "Extended Blended Learning – Innovative Learning Scenarios for universities", in Steven Furnell and Paul Dowland (eds.) *INC 2006 Conference Proceedings*, Plymouth 2006, 385–392.

Brückner, S. (2005), "Structure Patterns Depending on Various Learner Types in an e-learning Context", diploma thesis, Fachhochschule Darmstadt, Germany

Field, S. (1992), *Das Handbuch zum Drehbuch: Übungen und Anleitungen zu einem guten Drehbuch*, Zweitausendeins, Frankfurt am Main, Germany

Gertsch, F. (2007), *Das Moodle 1.8 Praxisbuch*, Addison-Wesley Verlag, München, Germany, ISBN 978-3-8273-2514-3

Gojny, U., Vejrazka, C., Leichtweiß, A., Trebing, C., Röder, H., Knorz, G., Bleimann, U., and Li, G. (2003). "ELATnet – E-Learning Entwicklung im Hochschulbereich". In *Querschnitt (Beiträge aus Forschung und Entwicklung der FH Darmstadt)*, number 17, pages 90–104, Darmstadt, Germany.

Grasbon, D. (2001), "Konzeption und prototypische Implementation einer Storyengine: Dynamisch-reaktives System zum Erzählen nichtlinear-interaktiver Geschichten bei größtmöglicher Spannung, gedanklicher Immersion, Identifikation und Motivation des Spielers", diploma thesis, Technische Universität, Darmstadt.

ILIAS open source (2008). "ILIAS Open Source LMS". http://www.ilias.de. (Accessed 28th January 2008).

Koper, R. and Tattersall, C. (2005), editors, *Learning Design*, Springer Verlag. Berlin Heidelberg.

LearningNetworks (2008). "Educational Modelling Language". http://www.learningnetworks .org/?q=EML. (Accessed 28th January 2008).

Martens, H. and Vogten, H. (2005), "A Reference Implementation of a Learning Design Engine", in Rob Koper and Colin Tattersall (Eds.) *Learning Design*, pages 91–108, Springer Verlag, Berlin, Heidelberg, Germany, ISBN 3-540-22814-4

Niegemann, H. M., Hessel, S., Hochscheid-Mauel, D., and Aslanski, K., and Deimann, M., and Kreuzberger, G. (2004), *Kompendium E-Learning*, Springer-Verlag, X.media.press, Berlin Heidelberg New York

Propp, W. J. (1958), "Morphology of the Folktale", in *International Journal of American Linguistics*, volume 24. Indiana University of Research Center in Anthropology, Folklore, and Linguistics, Bloomington, IN

Schneider, O., Bleimann, U., Harriehausen-Mühlbauer, B., and Phippen, A. D. (2006), "Non-Linear Extended Blended Learning", in *Proceedings of the Sixth International Network*

Conference (INC2006), pages 377–384, Plymouth, UK. Network Research Group, University of Plymouth.

Schneider, O., Bleimann, U., Phippen, A. D., and Harriehausen-Mühlbauer, B. (2007), “Learn Your Own Way: Decision Way through Narrative Approaches, Methods and Paradigms for Learning Purposes“, in Bleimann U., Dowland P. S., Furnell S. M. (Eds.) *Proceedings of the Third Collaborative Research Symposium on Security, E-learning, Internet and Networking (SEIN 2007)*, pages 99–114, Plymouth, UK. Network Research Group, University of Plymouth, ISBN 978-1-8410-2173-7

van Rosmalen, P. and Boticario, J. (2005). “Using Learning Design to Support Design and Runtime Adaptation”. In Koper, R. and Tattersall, C., editors, *Learning Design*, pages 291–302, Berlin Heidelberg. Springer Verlag.

WBT Systems Ltd. (2008). “TopClass E-learning Suite”. http://www.wbtsystems.com. (Accessed 28th January 2008).