[MBZRSt07] *Marek Meyer, Sonja Bergsträßer, Birgit Zimmermann, Christoph Rensing, Ralf Steinmetz;* **Modeling Modifications of Multimedia Learning Resources Using Ontology-Based Representations;** 13th International Multimedia Modeling Conference (MMM 2007), Singapore, Januar 2007, S. 34-43.

Modeling Modifications of Multimedia Learning Resources Using Ontology-Based Representations

Abstract. Repurposing of multimedia-based Learning Resources is an important issue in E-Learning, as economic success of content production depends on how intensively content is used. Repurposing does not only mean reuse "as is", but also comprises modifications of the contents to suit a different learning or teaching context, as well as reuse of fragments of a large Learning Resource. This paper introduces a method for modeling multimedia content modifications based on an ontology-based content representation. A theoretical background for modeling modifications of multimedia contents independent of the particular format is provided. Also a practical implementation is presented and discussed.

1 Introduction

Reusability is an important concept in the E-Learning community. Reusing existing E-Learning contents saves costs and also enables to benefit from the knowledge of other domain experts. For Web-Based Trainings (WBT) the Shareable Content Object Reference Model (SCORM) is the most common exchange format, which enables reuse of WBTs in different systems [1]. But reuse does not only consist of reuse "as is", but also comprises repurposing. Repurposing means to adapt a Learning Resource to a new learning or teaching context. 15 different relevant adaptations for Learning Resources have been identified in a user survey [2]. These adaptations can be broken down into several different fine-granular modifications.

There are some approaches, such as adaptive Learning Resources, single source publishing or layout templates (e.g. Cascading Style Sheets), that facilitate the adaptation to a few well-known scenarios; most contents though are and probably will be available only in non-adaptive.

However, SCORM is only one format, but there are others, as well. Also, even if SCORM is used, different formats, such as HTML, XML or Flash, may be used for the actual contents. Developing tools for repurposing is therefore a difficult and complex task. Hence, it is not advisable to develop a new tool for each adaptation and each format combination completely from scratch. Instead, frequently used functionality should be moved into a framework, which enables the developer of a repurposing tool to focus on the adaptation itself, instead on file and format handling details. Such a repurposing framework is described in [3]. This framework comprises a content ontology that contains concepts of content elements that are part of Learning Resources [4]. These concepts are independent from the particular format. In addition a Learning Resource Content

Representation (LRCR) is described which is a graph representation of a Learning Resource's content where concepts defined in the ontology are instantiated to describe each node in the graph. But an abstract content representation is not sufficient to support the development of repurposing tools. Modifications have to be modeled also in a format-independent way for completely outsourcing format-specific methods. In this paper, modeling efforts for content modifications are discussed. Modifications of Learning Resource content are considered regarding two aspects: a theoretical approach and an actual implementation.

This paper is structured as follows. A practical use case is presented in section 2 before the Learning Resource Content Representation is explained in section 3. Section 4 focuses on the modeling of content modifications. In section 5, the modification model is illuminated in the context of a practical implementation of an adaptation tool. Related work is discussed in section 6.

2 Use Case

A practical use case is presented here to illustrate the need for modeling content modifications. Consider a user Alice who is responsible for advanced training in her company. She is asked by her management to provide a course on the fundamentals of accounting to some employees. Because her budget is low, she decides not to produce a new course, but to buy an existing one from Bob. Unfortunately, this course does not comply to the corporate design of Alice's company - a fact which is not approved by her management. Furthermore, the terminology of Bob's course is partly unknown to Alice's target group and needs to be replaced.

Alice examines the obtained course and observes that it is a SCORM package that contains about 100 separate HTML documents. Each of these pages has a defined background and text color and Bob's company logo in the upper right corner. Now, Alice has to open each of these 100 HTML documents to change the background color, replace Bob's logo by her company's logo, move the logo to the upper left corner and replace unsuitable terminology by her own terms.

This scenario motivates the use of an adaptation tool, which enables Alice to adapt the layout and terminology of all documents of a Learning Resource at once and more easily.

But two weeks later, Alice notices that some employees have not yet participated in the online training, because they work in the field and do not have access to the Learning Management System most of the time. They would rather prefer learning the contents while traveling by train to their customers. Fortunately, Bob also offers a Microsoft Word document which describes the fundamentals of accounting. Alice buys the document at a special rate - but again layout and terminology do not suit Alice's company. Hence, she opens Microsoft Word for changing layout and terminology manually.

Would not it be great if Alice could use only one adaptation tool for the adaptation of different document types? Changing layout, replacing unsuited terminology, translating a document - from an end user's point of view all these adaptations are always the same, no matter which underlying format is used. Thus, Alice desires one tool for performing adaptations of all her Learning Resources.

And that is what this paper deals with - modifications of multimedia Learning Resources are modeled in a format-abstracted manner for supporting the development of format-independent adaptation tools.

3 Content Representation

To abstract the user from the details of a Learning Resource, a representation of the processed Learning Resource is needed. This representation has to deal with all the elements a Learning Resource may consist of. It needs to be able to deliver the information about the Learning Resource, which is needed to manage the Repurposing process. Hence, a mapping from the Learning Resource into a model which can provide all the required information is needed. Beside these two main requirements there exist several others [4]. The Resource Description Framework (RDF) is used for the content representation of the Learning Resource, because it fulfils all the identified requirements.

As a base for this model a conceptualization of a Learning Resource and the parts it consists of is required. The approach presented here uses an ontology for the conceptualization. In a repurposing process resources in multiple formats are involved, so it must be possible to find an instantiation of a concept defined in the ontology for all of them. A Content Ontology (CO) for describing Learning Resources has been developed. Details about the development of the Content Ontology and the corresponding Learning Resource Content Representation (LRCR) can be found in [4]. The LRCR is based on the representation of the structure of the Learning Resource. A separation of concerns has been realized by distinguishing between structural information and semantic information. The structural concepts can be used to describe the type of elements and also their relations and order. Additional semantic information can be added to the content representation for making the meaning of elements explicit. The meaning of elements can be marked with concepts at different levels of detail. For example a concrete description of an element is an example. A more abstract description of the same element is a SubjectEntity, an Entity which belongs to a certain subject. The level of detail in which a certain element is described depends on the information which is needed about this element and the analysis which is used to identify the element. Attributes and additional information about an element, such as information given in the metadata, are also included in the content representation (see Fig. 1).

By using the concepts defined in the Content Ontology a content representation of a Learning Resource can be build. This representation includes all information which is required in a Repurposing process and enables a view for the user which is uncoupled from all the details of the Learning Resource, e.g. in construction and formats.

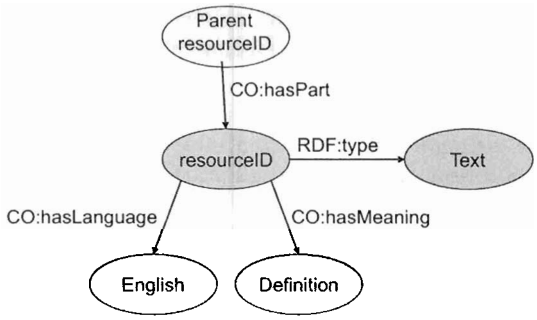

Fig. 1. Example for a LRCR clipping.

4 Modeling Modifications

In the previous section, a format-independent, ontology-supported Learning Resource Content Representation has been introduced. That content representation is static; changes of the contents cannot be specified within the scope of the LRCR. Therefore, another model for content modifications has to be provided. This section deals with format-independent modeling of modifications of multimedia Learning Resources.

4.1 Granularity of Modifications

An important design decision is the granularity of modifications. Is, for example, the replacement of a corporate design a single modification or a combination of several modifications? Zimmermann et al. have identified a structure of Adaptation Processes, which is helpful for the consideration of granularity [5]. On the most general layer, whole adaptation processes are resident, e.g. the adaptation to a different corporate design. An adaptation process divides into several process fragments. Process fragments are composed of adaptation functions, which may either read or modify the contents of a Learning Resource. Which of these granularity levels is best suited for modeling of content modifications?

The goal of modification modeling is to provide an abstraction layer for separating the concerns of repurposing tools and the format-specific content modification methods. Also, reuse of modifications, which are implemented once and reused for several repurposing applications, is a central motivation. In this respect, adaptation processes are too large to be reused easily and often. Process fragments are also application dependent, may rely on information from other process fragments, and sometimes comprise interaction with a user. They are also reused rarely. Adaptation functions, finally, are reusable for multiple process fragments, require no user interaction and need only a manageable amount of parameters to work. Therefore, content modifications are best modeled at the granularity of adaptation functions - restricted to those adaptation functions which cause changes of the content. These modifications are mainly insertion, from the previous example can now be observed in the LRCR space. The function, which modifies $\varphi(r)$ into $\varphi(r')$, is called mod_φ and represents an abstract modification of the Learning Resource content (see Fig. 2).

This algebra helps developing content adaptation tools. Adaptations have no longer to be implemented directly as format-specific methods. Instead, an adaptation tool analyzes $\varphi(r)$ (the LRCR) and specifies adaptations as a concatenation of modifications mod_φ . Each modification mod is transformed by an underlying layer into a format-specific modification mod_F. This transformation from LRCR space into the actual document format space is also called interpretation of an abstract modification. Fig. 3 illustrates these transformations.

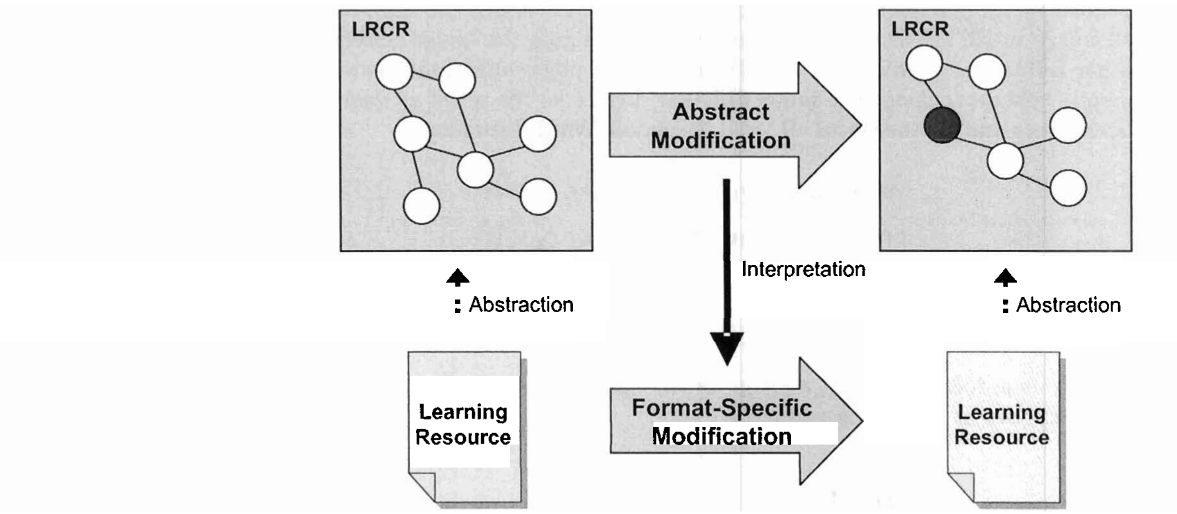

Fig. 3. Interpretation of abstract modifications.

5 Implementation

A repurposing tool for SCORM-based Learning Resources has been developed as part of the Content Sharing project. This repurposing tool is built upon the generic framework for format-independent content modifications, which implements a LRCR and provides an interface for executing abstract modifications as they have been sketched in the previous section. The whole repurposing tool has been implemented in Java (J2SE 1.4.2). The next section describes how abstraction modifications have been realized in practice.

deletion, replacement and rearrangement of elements, as well as changes of attributes and relations.

4.2 Theoretical Approach

The Learning Resource Content Representation is a graph and is considered to be a mapping of the whole contents, containing the information which is relevant for performing adaptations. Modifications at the granularity of adaptation functions produce only delimited local changes of the Learning Resource Content Representation. These changes can be expressed as graph operations. Consider there is a Learning Resource r in which one logo should be replaced by another one. If the Learning Resource consists of HTML documents, a logo is usually embedded by using a reference to the image file, which contains the logo. Replacing an image in HTML documents requires only changing the image reference. For other formats (e.g. Microsoft Word), images are physically embedded in documents; hence a replacement works different. Let H be the set of all valid HTML documents and W the set of all valid Microsoft Word documents.

$$exchangeLogo_H : r_1 \mapsto r_1' \quad |r_1, r_1' \in H$$
$$exchangeLogo_W : r_2 \mapsto r_2' \quad |r_2, r_2' \in W$$

And for the general case:

$$exchangeLogo_F : r \mapsto r' \quad |r, r' \in F, mod \in M$$

where r is a document from a given format space F and mod is a modification out of the set of all modifications M.

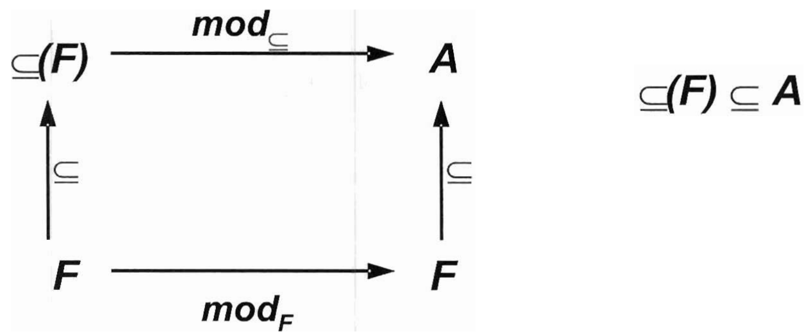

Fig. 2. Mapping into abstract LRCR space.

Consider the projection of Learning Resource r into the Learning Resource Content Representation $r \mapsto \varphi(r)$, where φ is the projection function from the document format space F into the abstract LRCR space A. The modification

5.1 Implementation of Modifications

First of all, it has to be distinguished between modification types, which are classes of modifications (e.g. "deletion of an element") and modification instances, which are actual modification requests at run-time (e.g. "delete element 1234").

Modification types are modeled as Java classes, which are all derived from a common interface called *IModification*. All classes provide a method for retrieving the primary target element of the modification, i.e. the element that is changed. Furthermore, each modification class may define further specific variables and methods, which are regarded as parameters for the particular modification type. A modification instance is an instance of one of the modification classes. There are currently three sub interfaces of *IModification* for structural modifications, layout modifications and content modifications.

At design-time, new modification types can be specified by implementing new Java classes. Notice that modification classes do not provide methods for actually performing a modification. Similarly, a modification instance does not change content by itself, but *represents* what has to be performed.

At run-time, an adaptation application instantiates one of the modification classes to express what needs to be changed. This modification instance is then passed to a framework, which performs the modification.

By now, 17 different modification classes have been implemented (including their format-specific interpretation), and more are yet to come.

5.2 Repurposing Framework

The multimedia content repurposing framework provides content analysis and modification services to repurposing applications. As interfaces to the application it provides access to an abstract content representation of a Learning Resource - the LRCR - and it accepts and executes modification requests. The overall framework is explained in detail in [3].

A repurposing application breaks the intended changes of a document down into a series of modifications. These modifications are instantiated as Java objects and passed to a modification transaction engine (MTE), which is part of the framework. The repurposing framework contains a number of format-specific plug-ins (FP) for the supported document formats. Based on the element of the content representation that is primarily targeted by the modification, the corresponding format-plug-in is identified and invoked. This format-plug-in then interprets the format-independent modification instance in a format-specific way by executing the appropriate Java method. Arguments for a modification method may be derived from variables of the particular modification class.

After all modifications have been performed, the LRCR is updated to represent the new state of the Learning Resource. Fig. 4 illustrates the components of the repurposing framework.

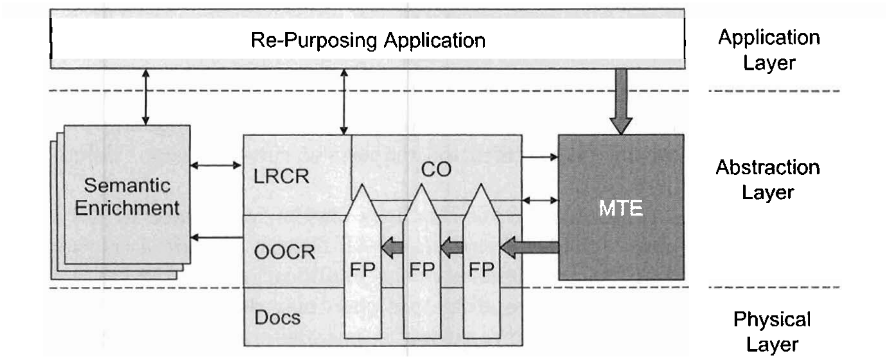

Fig. 4. Repurposing framework.

5.3 An exemplary repurposing application

An adaptation tool has been implemented on top of the repurposing framework. This tool already supports layout adaptations (e.g. replacing logos, background images, background and text colors) and adaptation for better printability (removing fixed widths of page elements).

Adaptations are realized as guided processes. As one part of the layout adaptation process, the tool searches for all background image definitions and background colors in all documents of a Learning Resource. Fig. 5 shows a screenshot of a dialog, where a user may specify new background images and colors for elements, which he has selected in an earlier step. The changes are instantiated by the adaptation tool as a set of modification objects, which are passed to the repurposing framework.

6 Related Work

The model driven architecture (MDA) approach separates application logic from underlying platform technology [6]. Platform independent models document the behavior of an application separated from the technology-specific implementation. Model transformation in the sense of the MDA approach can be seen as an application of a graph transformation [7]. This approach has found a large community and is used in different applications. Beside the different scenario, the idea to abstract from the implementation and specify generic models is also the motivation behind the approach presented here. It abstracts from format-specific resources and format-specific modifications of these resources by generating a resource model and by modeling the modifications which can be performed on the resources.

The ALOCoM framework [8] is an ontology based framework to enable reuse of Learning Objects. The framework is focused on slide presentations. A slide presentation is disaggregated into its components and mapped to a Java Object

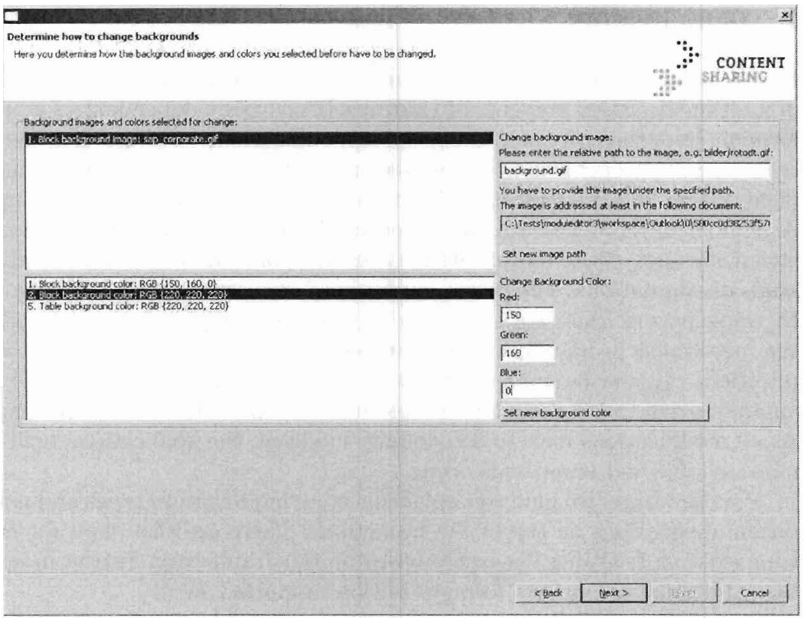

Fig. 5. Application example: dialog for changing layout information

Model. Out of the Java Object Model a RDF representation is generated and stored in a Learning Object Repository. Components are reused by copying existing components into a new slide presentation. This scenario allows the reuse of complete slide presentations, and of parts of these slide presentations, e.g. one image. The generated slide presentation uses default presentation styles. The format can be chosen out of a list of supported formats. The major difference to this approach is that ALOCoM converts Learning Resources into an intermediate format and transforms it back into another format for reuse. This may cause a loss of information. Also, the ALOCoM approach requires modifications of the contents to happen in the source or target format.

Kashyap and Shklar [9] propose an RDF model based approach to adapt content resources for different devices. In there work they use a representation of the features of the different devices and components which represent the content resources. A XML resource can be adapted to the different devices using device-specific style sheets. Depending on the device which is requesting content resources an appropriate style sheet can be generated based on the information in the RDF model. No library or collection is needed, containing specific style sheets for all the possible requirements a device might have. This approach follows the idea to uncouple information from presentation and to adapt certain properties of a content resource; it focuses on web applications.

7 Conclusions

In this paper a theoretical background is presented for modeling modifications of multimedia-based contents of Learning Resources independent of a particular format. Also, the issue of granularity of modifications has been considered. The theoretical approach has been realized in practice by representing content modifications as Java classes at design time and instantiated objects at run time. A number of modifications have been implemented, which support design and layout changes. More modification types are planned, which then enable other kinds of adaptations. The recent experiences are very promising. The concept of the repurposing framework - a relatively complex approach at first sight - and the investment in its development proved first positive results: Adaptation applications can now be implemented with reduced effort [5]. One example for such an adaptation tool has been presented in this paper. And even more important: As all modifications have to be carefully modeled, the applications tend to work more reliable and fewer bugs occur.

For the future, we plan to implement more modification types and new adaptation applications on top of the framework. There are also plans for modularizing existing Learning Resources based on this framework. Furthermore, one or two additional documents formats will be supported soon.

8 Acknowledgments

This work is supported by the German Federal Ministry of Economics and Technology in the context of the project Content Sharing.

References

1. Advanced Distributed Learning: Sharable content object reference model (SCORM) 2004, (http://www.adlnet.org)
2. Zimmermann, B., Bergsträßer, S., Rensing, C., Steinmetz, R.: A requirements analysis of adaptations of re-usable (e-learning) content. (2006) accepted for ED-MEDIA 2006.
3. Meyer, M., Hildebrandt, T., Rensing, C., Steinmetz, R.: Requirements and an architecture for a multimedia content re-purposing framework. In: Proceedings of the First European Conference on Technology Enhanced Learning. (2006)
4. Bergstraesser, S., Faatz, A., Rensing, C., Steinmetz, R.: A semantic content representation supporting re-purposing of learning resources. In: accepted for I-KNOW 2006. (2006)
5. Zimmermann, B., Rensing, C., Steinmetz, R.: Format-bergreifende anpassungen von elektronischen lerninhalten. In: accepted for Deutsche e-Learning Fachtagung Informatik 2006. (2006)
6. Object Management Group: Model driven architecture. (http://www.omg.org/mda/)
7. Rensink, A., Nederpel, R.: Graph transformation semantics for a qvt language. In: Proceedings of the Fifth International Workshop on Graph Transformation and Visual Modeling Techniques. (2006)

8. Verbert, K., Gasevic, D., Jovanovic, J., Duval, E.: Ontology-based learning content repurposing. In: WWW '05: Special interest tracks and posters of the 14th international conference on World Wide Web, New York, NY, USA, ACM Press (2005) 1140–1141
9. Kashyap, V., Shklar, L.: Declarative rdf models for feature-based targeting of content to multiple devices. In: Proceedings of the Tenth International World Wide Web Conference. (2001)