Repurposing Learning Object Components

Katrien Verbert[1], Jelena Jovanović[2], Dragan Gašević[3] and Erik Duval[1]

[1]Dept. Computerwetenschappen, Katholieke Universiteit Leuven,
Celestijnenlaan 200A, B-3001 Leuven, Belgium
{Katrien.Verbert, Erik.Duval}@cs.kuleuven.be
[2]FON-School of Business Administration, University of Belgrade,
POB 52, Jove Ilica 154, Belgrade, Serbia and Montenegro
jeljov@gmail.com
[3]School of Interactive Arts and Technology, Simon Fraser University Surrey,
2400 Central City, 10153 King George Hwy., Surrey, BC V3T 2W1, Canada
dgasevic@sfu.ca

Abstract. This paper presents an ontology-based framework for repurposing learning object components. Unlike the usual practice where learning object components are assembled manually, the proposed framework enables on-the-fly access and repurposing of learning object components. The framework supports two processes: the decomposition of learning objects into their components as well as the automatic assembly of these components in real-world applications. For now, the framework supports slide presentations. As an application, we will present in this paper the integration of this functionality in MS PowerPoint.

1 Introduction

Learning objects are often stored in a final presentation form. Such a static representation is neither suitable for flexible content reuse, nor adaptable to the needs of a learner. In many cases, specific parts are assembled manually by copy and paste actions. However, it is possible to reuse learning objects in a much more sophisticated way if their components can be accessed on-the-fly. This requires a more innovative and flexible underlying model for learning object components [2].

In earlier work, we developed an ontology (ALOCoM) for learning objects that is a framework for learning objects and their components [10]. The ontology defines learning object component types, as well as relationships between these components. As such, the ontology enables structuring of composite learning objects and is a solid basis for the proposed dynamic approach. In this paper, we describe a framework that transforms existing learning objects from their tool specific formats (MS Office, OpenOffice.org) into a representation compliant with the ALOCoM ontology. In this transformation process, the framework disaggregates learning objects and provides direct access to their components, enabling their reuse in dynamic compositions of new learning objects.

For now, the framework supports slide presentations as they are one of most common used learning object types [8]. Often a teacher wants to

repurpose a slide or an image, a reference, a definition or just a text fragment of a particular slide. In our approach, we decompose the presentation into these individual components and store them into the learning object repository. As such, these components are accessible on-the-fly. The disaggregation of a slide presentation proceeds in three steps. In the first step, the presentation is parsed and decomposed into clear segments, namely its slides and each slide is further decomposed into its title, paragraphs, lists, list items, images, diagrams and tables. In the second step, text patterns are applied to categorize these segments into more meaningful components like definitions, examples, references, introductions and summaries. Finally, components are described by metadata using the Automatic Metadata Generation framework [1], improving the findability of relevant components.
Having in mind the widespread use of MS PowerPoint [8], we have created a MS PowerPoint add-in that allows searching for components in the learning object repository from within the MS PowerPoint application. When a teacher is creating a new slide presentation, (s)he can search for definitions, slides, examples, references and images (s)he wants to repurpose. Available components are displayed on the Clipboard and can directly be added to the slide presentation.
In the next section, we briefly outline the ontology that formalizes learning object types and their components. In section 3, we present the ALOCoM framework and section 4 illustrates the transformation of MS PowerPoint slide presentations. Section 5 provides more details about the classification of components and section 6 elaborates on annotating components. In section 7, we will demonstrate the integration of the ALOCoM framework into MS PowerPoint. Related work is presented in section 8 and conclusions and remarks on future work conclude this paper.

2 The ALOCoM Ontology

In earlier work, we developed the ALOCoM ontology as a generic Abstract Learning Object Content Model (ALOCoM - see Figure 1) for learning objects and their components [10]. The ontology distinguishes between content fragments (CFs), content objects (COs) and learning objects (LOs). CFs are learning content elements in their most basic form, like text, audio and video. These elements are uncombined with other elements. COs aggregate CFs and add navigation. Navigation elements enable structuring of CFs in a CO. Besides CFs, COs can also include other COs. Finally, LOs aggregate COs around a learning objective.
We defined content types for each of these components. We introduced CF types such as an image, text, and audio and video sequences. For defining CO types, we investigated existing Information Architectures, like the Information Block Architecture developed by Dr. Horn [3] and the IBM Darwin Information Typing Architecture (DITA) [9]. These architectures define information types (e.g. concept, principle, task) and their building blocks (e.g. example, definition, analogy). As a starting point, we defined CO types and their structure using DITA concepts,

since DITA is a recent architecture with rich documentation and online support [9]. Besides CF and CO types, the ontology identifies LO types. For now, only a slide presentation LO type is defined. Finally, the ontology defines the relationships between the LO components. Aggregation and navigation relations are specified. Aggregation relationships between components are represented in the form of a "hasPart" and its inverse "isPartOf" ontology properties. Navigational relationships are specified as a list that defines the order of components in a CO or LO. For more information about the ontology, see [4].

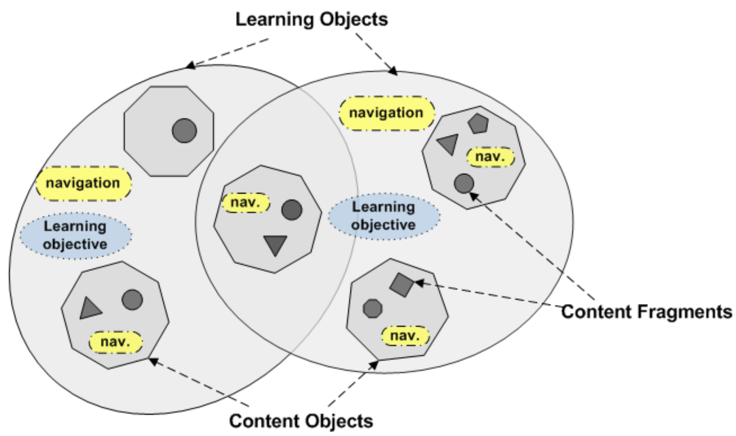

Fig. 1. Abstract Learning Object Content Model

3 An Ontology-based Framework for Component Repurposing

Our main focus is on the development of tools for disaggregating learning objects into their components (i.e. disaggregators) as well as for repurposing learning object components in real-world applications (i.e. aggregators). For now, we developed a framework that provides both functionalities for slide presentations. Since the most popular tools for slide presentation authoring are MS PowerPoint and OpenOffice.org [8], the proposed framework focuses currently on slides presentations authored using these tools. The framework decomposes MS PowerPoint and OpenOffice.org slide presentations and assembles components into new MS PowerPoint and OpenOffice.org slide presentations on-the-fly. The disaggregation and re-aggregation processes are illustrated in Figure 2. In the disaggregation process, a slide presentation is parsed and disaggregated into clear segments (slides, paragraphs, lists, list items, images, diagrams and tables). In the second step, these segments are categorized into more meaningful content objects like definitions, examples, references, summaries, overviews and introductions. We use text patterns to classify these content objects. In the last step, content objects are annotated using the AMG framework [1]. These transformation steps will be further explained in the next three sections.

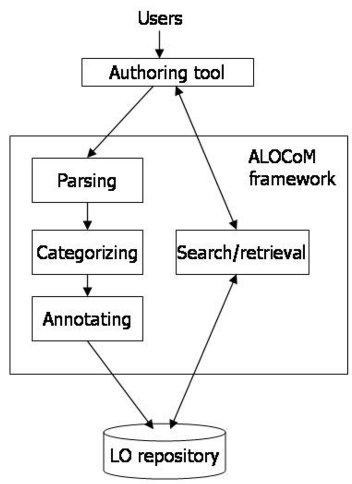

Fig. 2. The ALOCoM framework

The opposite (aggregation) process searches for components in the learning object repository and adds them to a slide presentation. Since authors prefer to use authoring environments they are familiar with, this functionality has to be integrated in present authoring tools. We have currently developed an add-in for MS PowerPoint. The same functionality will be provided for OpenOffice.org. Section 7 illustrates the use of the PowerPoint add-in, while the next one gives implementation details of the procedure for disaggregating slide presentations created in MS PowerPoint.

4 Parsing MS PowerPoint Slide Presentations

In the first transformation step, MS PowerPoint slide presentations are disaggregated into structured components. A slide presentation is decomposed into its slides and each slide into its title, paragraphs, lists, list items, images, diagrams and tables. We use the Microsoft PowerPoint .Net API for this transformation.
Microsoft Office Presentation objects are arranged in a hierarchical order, as shown in Figure 3. The two main classes at the top of the hierarchy are the Application and Presentation classes. An Application object provides a wrapper around the entire application. Each Presentation object

represents a single Presentation document. Each of these objects has many methods and properties that allow manipulating and interacting with it. A Presentation has a Slides property that returns a collection of all the Slide objects in the presentation. A single slide is retrieved by specifying its name, index number or slide ID number. Each slide has a Shapes property that returns a Shapes collection representing all the elements that have been placed or inserted on the specified slide, slide master, or range of slides. This collection can contain drawings, shapes, OLE objects, pictures, text objects, titles, headers, footers, slide numbers, and date and time objects appearing on a slide, or on the slide image on a notes page. Text objects contain presentation related information, enabling us to infer the structure of the text. For instance, if the Bullet property of a paragraph is set to visible, we transform the text fragment into a list. More generally, all content, structure and presentation related information is retrieved and transformed into an explicitly structured format.

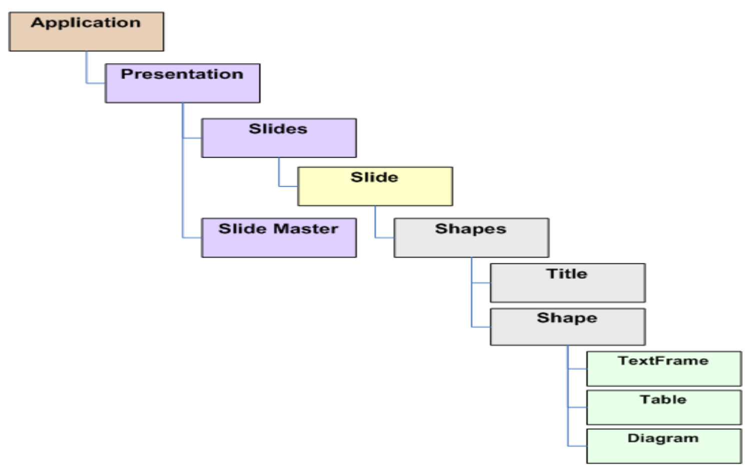

Fig. 3. PowerPoint Object Model

Figure 4 shows an excerpt of a MS PowerPoint slide presentation after being transformed into an ALOCoM compliant format.

```
<SlidePresentation>
<slide>
<title>Overview</title>
<slidebody> <!–...–> </slidebody>
</slide>
</SlidePresentation>
```

Fig. 4. A slide presentation in the ALOCoM XML format

5 Categorization of content

The next step involves the categorization of the segments of the previous step (paragraphs, lists, list items) into more meaningful content object types, like definitions, examples, references, overviews, introductions and summaries. Due to the lack of strict compliant rules for creating content, this is not a trivial task. However, for the categorization of some of these content object types (definitions, concepts) research has been done and solutions exist. For instance, a system called Definder uses rule-based techniques to extract definitions from medical articles [7]. Definder uses cue-phrases (e.g. *is the term for*, *is defined as*, *is called*) and text markers (e.g. − −, ()) in conjunction with a finite state grammar to extract definitions. Based on this system and research of Bing [6], we identified patterns that are suitable for recognizing definitions, examples and references in presentation slides.

5.1 Definition extraction

The following patterns are applied to identify definitions of concepts:

1. {is|are} [*adverb*] {called | known as | defined as} {*concept*}
2. {*concept*} {refer(s) to | satisfy(ies)} ...
3. {*concept*} {is|are} [*determiner*] ...
4. {*concept*} {is|are} [*adverb*] {being used to | used to | referred to | employed to | defined as | formalized as | described as | concerned with | called} ...
5. {What is} [*determiner*] {*concept*}?
6. {*concept*} {- | : } {*definition*}
7. {definition} [of] {*concept*} {- | : } ...

Legend:
{} - compulsory field
[] - optional field
adverb - e.g., usually, normally, generally,...
determiner - e.g., the, one, a, an, ...
definition - definition of a concept

For example, using pattern number five from the presented patterns list, content of a slide with title *"What is an ontology?"* is categorized as a definition of the ontology concept. Similarly, a list item containing the text *"an ontology is a specification of a shared conceptualization"*, will be classified as being a definition according to the third pattern. Although some authors use braces (e.g. () <> []) to wrap definitions, they are not used to detect definitions in our work. Braces are also used to wrap examples, illustrations and descriptions, so they will not help us in distinguishing between these components.

5.2 Example extraction

The following patterns are applied to identify examples of concepts:

1. {example, instance, case, illustration, sample, specimen} [of] {*concept*}
2. {for instance | e.g. | for example | as an example} [,] [*determiner*] {*concept*} ...
3. {*concept*} {illustrates | demonstrates | shows | exemplifies} ...
4. {*concept*} {is|are} [*adverb*] {illustrated by | demonstrated by | shown by} ...
5. {Example} {- | : } {*example*}

Legend (new items):
example representing an example of a concept

5.3 Reference extraction

There are more strict guidelines for references, what makes identifying them easier. For instance, references are often preceded by the sequence "[identifier]", where identifier is a number or character sequence. An other standard uses the sequence "Name (Year)" to start the reference. This results in the following two identification patterns for references:

1. {[}{*identifier*}{]} {*reference*}
2. {*Name*} {(}{*Year*}{)} {*reference*}

Legend:
identifier - number or character sequence, e.g., 1, 2, Nam01 ...
reference - literature reference

5.4 Summary, overview and introduction extraction

Since we are currently working with slide presentations, we can easily classify introductions, summaries and overviews by looking at the title of slides. Slides are classified as summaries if their title is *"conclusion", "summary", "future work"* or a combination of these values. Introduction and overview slides are in most cases entitled respectively *"introduction"* and *"overview"*.

6 Annotating components

The last step of the transformation consists of annotating the learning object components. We are using the Automatic Metadata Generation (AMG) framework to automatically describe each component (http://ariadne.cs.kuleuven.ac.be/amg). The idea behind the framework is to combine metadata, generated by different sources into one metadata instance [1]. The first source is the learning object itself; the second is the context in which the learning object is used. Metadata derived from the object itself is obtained by content analysis, such as keyword extraction and language classification. The contexts typically are learning (content) management systems (like Blackboard) or author institution information. A learning object context provides additional information about the learning object that is used to define the metadata. In our case, we developed an extension of the framework that combines metadata by an inheritance mechanism. The metadata describing a component is also defined by the parents of this component. For instance, each slide inherits the author, language, etc. from the slide presentation it belongs to. Other metadata fields like for instance the title are overwritten. Also the main concept of a slide is in many cases more specific than the main concept of the slide presentation as a whole. Furthermore, dependency relations between learning objects and their components are described as relation metadata within a relation element. Through additional attributes the relation element specification allows distinguishing different relations between parent and child components (”isPartOf”, ”hasPart”) and between components (”ordering”). As such, we defined an extension of the AMG framework that deals with an inheritance mechanism and relationships between components.

7 An application: the ALOCoM framework integrated in MS PowerPoint

We have created an add-in for MS PowerPoint (Figure 5), which allows authors to repurpose components stored in the ALOCoM learning object repository without leaving their authoring environment (in this case MS PowerPoint). The add-in enables authors to search the repository for learning object components they wish to repurpose in the slide presentation they are working on. An author can specify the type of component he/she is interested in (e.g. reference, definition, example, slide, image), as well as keywords or other metadata fields that best describe the component. All components that satisfy the specified search criteria are added to the Clipboard and the author can easily incorporate them into the current slide presentation. Another functionality provided by the developed add-in is related to adding new content to the learning object repository. Each author can add his or her slide presentation to the repository by clicking the ”Save into ALOCoM” button. When this button is clicked, the PowerPoint presentation is disaggregated and stored into the ALOCoM LOR.

A typical use case goes as follows: Suppose an author is creating a slide presentation on ontologies. He/she wants to start with a definition, followed by three examples. The author enters ”ontology” as keyword and selects ”definition” and ”example” as types of components that he/she is interested in. The system then searches the LOR and retrieves all components of the selected types dealing with the selected topic. The components that are found are added to the Clipboard. The author chooses the most relevant components from the set prepared for him/her. Furthermore, the author wants to include a reference to a book (s)he wants to recommend and an image of the book. Again the author searches the LOR and selects the component he/she wants to repurpose from the set of retrieved components. The author enhances the slide presentation with an additional example of an ontology and the presentation is ready for in-class use. Finally, the author clicks the ”Save into ALOCoM” button. The slide presentation is disaggregated and also all (new) components of this presentation are available in the repository.

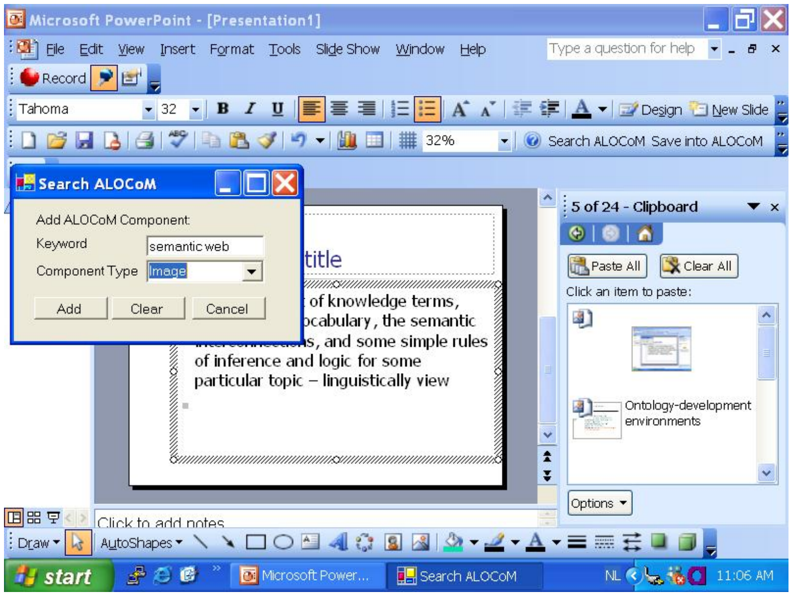

Fig. 5. MS PowerPoint - ALOCoM add-in

8 Related Work

The TRIAL-SOLUTION project is developing tools to create and deliver personalized teaching materials that are composed from a library of existing documents on mathematics at undergraduate level [5]. Analogously to the ALOCoM work, the TRIAL-SOLUTION project defines an ontology for learning objects that includes mathematical categories like definition, theorem, proof, or example. The focus of the project is on document (de-)composition and exchange of learning objects for reuse. The TRIAL-SOLUTION System contains a splitter that decomposes document source files into a hierarchy of slices. For this decomposition, the presentation style of a particular author is taken into account. Also, it takes care of counters and key phrases assigned by the author. As such, the methodology for decomposing learning objects is more accurate but less scalable than the methodology presented in this paper.

9 Conclusions and Future Work

In this paper, we have shown how we can improve present learning object authoring tools (e.g. MS PowerPoint) by integrating functionalities that allow on-the-fly repurposing of learning object components. The developed prototype validates this approach for slide presentations. In the next steps, the framework will be extended to support a broader range of learning objects. Furthermore, the efficiency and effectiveness of this approach for learning object repurposing will be evaluated. This work will then result in a general framework for reusable learning objects, that allows not only automatic repurposing of learning objects, but also their components and that will enable the dynamic generation of learning objects, adapted to the needs of learners.

10 Acknowledgements

We gratefully acknowledge the financial support of the Katholieke Universiteit Leuven research council through the BALO project and of the European Commission through the ProLearn Network of excellence, which has facilitated the collaboration between the Computer Science Department of the K.U. Leuven, the GOOD OLD AI group at the University of Belgrade and Simon Fraser University Surrey (SFU). The research of SFU is also funded by LORNET, a NSERC Research Network.

References

1. K. Cardinaels, M. Meire, and E. Duval. Automating Metadata Generation: the Simple Indexing Interface. In *Proceedings of the fourteenth international conference on World Wide Web*. ACM, 2005.
2. E. Duval and W. Hodgins. A LOM research agenda. In *Proceedings of the twelfth international conference on World Wide Web*, pages 1–9. ACM Press, 2003.
3. R. E. Horn. Structured writing as a paradigm. In *Instructional Development: the State of the Art. Englewood Cliffs, N.J.*, 1998.
4. J. Jovanovic, D. Gasevic, K. Verbert, and E. Duval. Ontology of learning object content structure. In *Proceedings of the 12th International Conference on Artificial Intelligence in Education*, 2005.
5. W. Lenski and E. Wette-Roch. The TRIAL-SOLUTION Approach to Document Re-use Principles and Realization. In *Electronic Media in Mathematics*, 2001.
6. Bing Liu, Chee Wee Chin, and Hwee Tou Ng. Mining Topic-Specific Concepts and Definitions on the Web. In *Proceedings of the twelfth international conference on World Wide Web*. ACM Press, 2003.
7. S. Muresan and J.K. Klavans. A Method for Automatically Building and Evaluating Dictionary Resources. In *Proceedings of the Language Resources and Evaluation Conference*, 2002.
8. J. Najjar, S. Ternier, and E. Duval. The Actual Use of Metadata in ARIADNE: An Empirical Analysis. In *Proceedings of the 3rd Annual ARIADNE Conference*, pages 1–6, 2002.
9. M. Priestley. DITA XML: a reuse by reference architecture for technical documentation. In *Proceedings of the 19th annual international conference on Computer documentation*, pages 152–156. ACM, ACM Press, 2001.
10. K. Verbert and E. Duval. Towards a global architecture for learning objects: a comparative analysis of learning object content models. In *Proceedings of the ED-MEDIA 2004 World Conference on Educational Multimedia, Hypermedia and Telecommunications*, pages 202–209. AACE, AACE, 2004.