Enforcing reuse and customization in the development of learning objects: a product line approach

A. Ezzat Labib  
ISSI-DSIC  
Universitat Politècnica de València  
Cno. de Vera, s/n. 46022 Valencia. Spain  
alabib@dsic.upv.es

José H. Canós  
ISSI-DSIC  
Universitat Politècnica de València  
Cno. de Vera, s/n. 46022 Valencia. Spain  
jhcanos@dsic.upv.es

M. Carmen Penadés  
ISSI-DSIC  
Universitat Politècnica de València  
Cno. de Vera, s/n. 46022 Valencia. Spain  
mpenades@dsic.upv.es

Abel Gómez  
AtlanMod team  
Mines-Nantes – Inria – Lina  
4 rue Alfred Kastler, 44307 Nantes, France  
abel.gomez-llana@inria.fr

ABSTRACT
The growing use of information technologies in the educational cycles has raised new requirements for the development of Interactive Learning Materials in terms of content reuse, customization, and ease of creation and efficiency of production. In practical terms, the goal is the development of tools for creating reusable, granular, durable, and interoperable learning objects, and to compose such objects into meaningful courseware pieces. Current learning object development tools require special technical skills in the instructors to exploit reuse and customization features, leading sometimes to unsatisfactory user experiences.

In this paper, we explore a new way to reuse and customization following Product Line Engineering principles and tools. We have applied product line-based document engineering tools to create the so-called Learning Object Authoring Tool (LOAT), which supports the development of learning materials following the Cisco’s Reusable Information Object strategy. We describe the principles behind LOT, outline its design, and give clues about how it may be used by instructors to create learning objects in their own disciplines.

Keywords
E-learning; Learning Object; Content Model; Document Product Line; Authoring Tool

1. INTRODUCTION
The efficient development of instructional materials is one of the major challenges of the e-Learning communities. The high diversity of (sometimes overlapping) domains, along with the diversity of learners in terms of languages, skills, and achievements, have led research and development efforts towards the development of course materials, training materials, instructor/learner guides, and assessment units, just to name a few artifacts used in learning processes, from sets of reusable, granular, interoperable, and durable pieces.

This paper is organized as follows. Section 2 introduces a back-ground on the RIO-RLO content model and DPL method. Section 3 describes how DPL is used to create Reusable Learning Objects (RLOs). Section 4 outlines the LOT tool which we are developing to provide a domain oriented layer to the system, and Section 5 presents the conclusions and future work.

2. Background
2.1 The RIO-RLO content model
In 1999, Cisco Systems Inc. developed a Reusable Information Object (RIO) strategy for developing and delivering the learning content in the form of RLO [2]. The Cisco RLO/RIO model is based on the Learning Object Strategy of Merrill [4] and Clark [5]. A RIO is a granular, interoperable, reusable piece of information which contains a metadata level to describe its character-
istics, purpose and relationship with other objects. Each RIO consists of three parts: content items, practice items, and assessment items. All these items are grounded upon a single learning objective. Content items are classified as a definition, example, review, next steps, analogy, topology illustration, block diagrams, etc. Practice items are activities that give the learners the chance to apply their skills and knowledge. Practice must have a direct relationship with the learning objective of the RIO. The last component of RIO is the assessment, used to determine gaps in knowledge and skills before taking the RLO and to ensure that the learner has mastered the objective of the RIO. There are two types of assessment, namely pre-assessment and post-assessment.

RIOs can be combined together to form a larger structure called a RLO. The Cisco RIO-RLO model states that the number of RIOs needed to construct a new RLO are from five to nine; RLOs are completed with an overview and a summary. The structure of RIOs depends on the type of RIO being defined. RLO components are structured items, as summarized in [2]. Figure 1 shows the UML class diagram representation of the Cisco RIO-RLO model.

2.2 Document Product Lines

DPL applies product line engineering principles to the semi-automatic generation of documents in domains with high content variability and reuse. Central to DPL is the notion of a family of documents. By family we mean a set of documents that share some common parts while differ in other parts. Every member of the family is built by assembling a set of content components. The DPL process is composed of two main activities: Domain Engineering and Application Engineering. The Domain Engineering starts with the specification of a family of documents in terms of content features, which represent document fragments that either must or can be included in a specific document. Every content feature must be linked to one or more technology features, which define how a particular content feature is represented. Additionally, every content feature is associated to actual content by linking it to some content component.

Reusable content assets, called InfoElements, are organized and stored in the DPL Repository. Each InfoElement has a specific content plus some descriptive metadata, and can be reused just by attaching it to a particular document feature. There is no pre-scription about the granularity of InfoElements.

Given a document family specification (that is, a document feature model), a specific member of the family is defined by means of a configuration in the Application Engineering stage of the DPL process. There, the user selects which optional features are to be included in the document along with the mandatory ones, which are common to all the members of the family. After the configuration, an automatic process assembles the document taking the InfoElements from the Repository, as described in [6].

We have applied the DPL principles to the development of RLOs. With this aim, we have performed a DPL process from the Domain Analysis to the Application Engineering using DPLFW, an implementation of the DPL method. We have defined the family of RLOs, and we will use it as the starting point for the generation of instructional materials. However, DPLFW is not an end-user tool, that is, its intended users are not instructors but document engineers. To provide both instructors and students with a friendly environment, we are developing LOAT, a Learning Object Authoring Tool that will use DPLFW services to provide variability and reuse management as well as customiza-tion to Learning Object authors.

3. Using DPL to Engineer Reusable Learning Objects

In this section, we detail the principal steps in the creation of the RLO document product line using DPLFW:

Step 1. The RLO Document Feature Model — We show how a family of RLOs is defined in DPL. To do this, we recall the RLO structure depicted in [2], which will guide the definition of the feature model. As described in Figure 2.a, a RLO includes three components: Overview, Summary and a set of RIOs which, in turn, include content, practice and assessment elements, and so on. The only difference to the Cisco model [2] is that, for the sake of flexibility, we have removed the ?and RIOs per RLO constraint. To illustrate the process, we use as example a course entitled "Introduction to Programming using Java". Figure 2.a shows the DPL document feature model of the course. The notation uses exclamation marks to denote mandatory features, and question marks for the optional ones; double-head arrows denote for alternative features. Content features are represented as CDFs whereas the TDFs represent the presenta-tion of the document generated.

At this point, we are just modeling what a RLO will be composed of, without any mention to actual content. To link CDFs with content, we have to search the right content components at the DPL Repository, as we illustrate in the next step.

Step 2. Creating and Storing RIO Components — Building a RLO with DPL depends on the presence of its components in the Repository. As we mentioned earlier, the content components of the Repository are called InfoElements [6] and contain two types of properties: data (i.e. the actual content), and metadata (data used to describe the InfoElement and serve as criteria to retrieve components from the Repository).

Figure 1. UML representation of the Cisco RIO-RLO model

Figure 2. The RLO document feature model (a) and Configuring a specific RLO (b)
4. LOAT: towards a new Learning Object Authoring Tool

All the DPL functionality shown so far is related to the pure document engineering tasks of defining RLOs. This means that there is a need for a domain layer, that is, a wrapper that hides the DPL complexity to the intended users of the RLO authoring tool. Such a layer will be implemented in LOAT, a new tool that enforces reuse and customization to increase the efficacy of Learning Object authoring processes. Three basic principles have been considered in the design and implementation of LOAT: the cognitive level, learning object classification and content-model architecture. The first one derives from the combination of the best practices described in Merrill’s Component Display Theory [4], Bloom’s Taxonomy of Educational Objectives [7], and Clark’s Developing Technical Training [5]. The second one is derived from [5] (concept, fact, procedure, process, or principle). The third one is the content-model architecture for designing a reusable and granular learning object de-fined in RIO-RLO content model.

LOAT follows the classical component content management system strategies. In other words, LOAT is a part of Learning Content Management System that manages content at a granular level (component or asset) rather than at the document level. Each component represents a single topic or asset. Upon this we have a great flexibility in re-using and producing mass customization e-learning materials in different delivery formats.

The LOAT architecture is divided into three layers (Fig. 3). The first one is the Repository Layer, which provides persistence to the artifacts (both RIOs and RLOs) generated. The repository layer uses the DPL Repository for saving the assets and the LOAT Repository for the created RIOs and RLOs. The second layer is the Services Layer, which contains a content authoring tool for the creation of assets and Learning Objects. The Content Model feature in the services layer determines which content model will be applied in the learning object creation process. Collaborative Authoring is an important feature in our tool that allows many partners to share the Learning Object creation process. Finally the Presentation Layer handles how Learning Objects are displayed in different output formats for the purpose of delivery in different mediums.

5. Conclusions and further work

The efficient development of instructional materials is one of the major challenges of the e-Learning communities. With the aim of facilitating the construction and production of instructional materials with higher levels of reuse and flexibility that the one provided by current authoring environments, we have introduced a new approach to RLO development based on product line engineering principles. The Document Product Lines method, and its implementation DPLfw, have been used to define families of RLOs whose components may be selected dynamically from a set of previously developed content pieces that are assembled automatically according to predefined rules.

This is our first effort towards the design and implementation of LOAT, a new authoring tool based on the principles described in this paper. LOAT will act as a wrapper of the DPLfw function-ality so that friendly interfaces will be offered to instructors. LOAT will be integrated in a learning management system, providing interoperability with existing tools in order to reuse existing content in different learning contexts.

6. ACKNOWLEDGMENTS

This project has been funded with support from the European Commission. This publication [communication] reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein. The work of J.H. Canós and M.C. Penadés is partially funded by the Spanish MEC under grant TIPEx (TIN2010-19859-C03-03).

7. REFERENCES