A MODEL-BASED APPROACH FOR CREATING LEARNING OBJECTS: A COLOMBIAN CASE STUDY FROM A PUBLIC UNIVERSITY

Boris Perez[1]*, Eduard Puerto[2] and Fredy Vera[3]

[1]Prof. Ms., Francisco de Paula Santander University, COLOMBIA, borisperezg@ufps.edu.co

[2]Prof. Ms., Francisco de Paula Santander University, COLOMBIA, eduardpuerto@ufps.edu.co

[3]Prof. Ms., Francisco de Paula Santander Univ., COLOMBIA, fredyhumbertovera@ufps.edu.co

*Corresponding author

Abstract

The Systems Engineering (SE) program of Francisco de Paula Santander University has implemented a learning objects repository (LOR). The learning objects (LOs) developed use different basic tools that do not exploit the complete capabilities of the Web, like personalization, improved graphics components and support for multiple platforms, and others. The time and resources required to make improvements to the content to allow us to provide the content to other platforms, such as mobile, or integrate it with different Learning Management Systems (LMS) is too much. Also ensuring adaptability over time is a desirable property of LO. The main objective of this proposal is to apply the concepts of model-driven architecture (MDA), to develop a platform independent model (PIM). The approach will allow us to design the LOs regardless of the technology platform where intended to be deploy it. Thus, the entire contents of LO can be updated by anyone, without the required technological knowledge.

Keywords: learning objects, model-based approach, e-learning, transformation rules.

1 INTRODUCTION

Technology use in teaching processes is becoming ever more popular amongst students and teachers alike, this is why it's important to understand how it can be used in aiding changing teaching methods as well as producing material to aid learning (Souza, Castro and Andrade, 2008).

Something that has come up with the growing use of technology in the classroom are Learning Objects (Souza, Gomes, Barroso and Souza, 2007, pp. 39-52); a Learning Object (hereby referred to as LO) is an entity, be it digital or not, used to aid the learning process (IEEE, 2002). These entities are often accompanied by activities such as exercises, tasks and discussion topics with the means of promoting learning (Martinez-Ortiz, Sierra and Fernandez-Manjon, 2009, pp. 189-202). These objects are used as tools to assist teachers in designing new strategies to facilitate student's learning (Souza, Castro and Andrade, 2010, pp. 701-702).

However, many of these LO's are built as blocks, or "islands", making it difficult for both teachers and authors to update content like examples, exercises and resources amongst other things (Souza, Castro and Andrade, 2010, pp. 701-702). Together with what we've just mentioned there's also the problem of multiple deployment platforms, such as web browsers, mobile devices and Facebook integration. Generally speaking, authors do not have sufficient technical knowledge to handle such issues.

From what we have just mentioned, this work is presented in order to offer autonomy to authors, using a model based approach so that content can be modified when required, without the need for third party assistance. This proposal focuses on allowing material to be updated, taking currentness and exercise difficulty into account.

ISBN: 978-605-64453-2-3

2 PROBLEM

According to (Johnson, Chapaman and Dyer, 2006), there is a huge amount of digital material available online, although there are very few teachers using it in their classes. Such material is often protected and hence cannot be adapted to different classroom situations; even teacher-designed material can become out-dated a few years after creation. To solve this problem, teachers must be able to modify what is needed without the need for help with technical issues.

This author independence with technical issues does not solely apply to LO modification but also to platform deployment. E.g. Teachers may wish to deploy an LO to mobile devices, and later on they may like to integrate it with Facebook or a Learning Management System (LMS). Generally when dealing with mobile devices, the LO design must be adjusted in order to be seen correctly, it is especially important to mention that mobile platforms are important variables when designing LOs, considering students are using them more and more to access digital content.

LOs at UFPS are not currently being designed on any specific platform nor has any particular programming language been determined. Every author develops their LOs following a few guidelines, but overall has total freedom when it comes to design. LO construction is intertwined with author/creator programming language knowledge, e.g. LOs have been developed using PHP, HTML and even Flash, although lesser experienced authors are inclined to use frameworks such as Constructor or eXe Learning.

Thus this way of configuring digital resources holds a few advantages due to the time needed to make modifications and create content for new platforms being considerably reduced, as no third person is required (Johnson, Chapaman and Dyer, 2006).

And so the need has arisen to allow authors with no previous technical knowledge, not of deployment platforms nor for future adjustments in terms of both technology and content to create LOs. Our article focuses on achieving LO construction and maintenance independent of the implementation technology through a model based approach.

In this context we propose the use of a Model Driven Approach (MDA) for LO construction. This approach allows the business logic to be separated from the technical details of a specific technology platform. Using this model we can specify functionality and behaviour of PIM (Platform-independent model) applications that can be described using UML or any other standard. So both business and technical aspects can evolve independently at their own pace. Business logic alone answers to new requirements and technical advancements as and when the market sees fit.

3 BACKGROUND

3.1 Learning Object

According to Wiley (Wiley, 2002) LOs are “any digital resource that can be reused to support learning”, however this definition is quite broad. Polsani’s (Polsani, 2006) definition has a more educational intention: “A media asset or a digital object can become a LO only when it is incorporated into a form and provides a relation to itself as LO in order to facilitate the understanding of that object”. In (Pukkhem and Vatanawood, 2011) Noppamas establishes that los learning objects can be educational components presented in any format. Adamchik (Adamchik and Gunawardena, 2003, pp. 96-99) defines a learning object (LO) as an integrated module containing the core text, code examples, review questions, supplementary material, and programming labs.

Despite these definitions on what an LO is, there’s still no globally accepted definition. So for the purpose of this investigation we decided to use the definition proposed by Ünal (Çakiroğlu, Adnan and Akkan, 2012), who established that “LOs can be united in order to be used in different areas, can be reused, and can be arranged easily”. This definition would only be improved by saying they are also available online. This means students with different learning needs may benefit from LOs, being able to study them at their own pace wherever they may be, autonomously.

An LO looks to be an instructional aid thanks to its flexibility, accessibility, durability, interoperability and reusability (Conceição and Lehman, 2003, pp. 44-49). In an academic environment, LOs offer unique capacities and within the reach of every student so that they can work at their own pace, through personalized tutorials along with feedback. With this, the professor can easily work in heterogeneous classes facing different levels without having the need of slowing the pace of other students (Spektor-Levy and Granot-Gilat, 2012, pp. 83-96).

ISBN: 978-605-64453-2-3

In this day and age, a person's ability to create and consolidate the knowledge required to do so age is more important than ever, rather than just accumulating knowledge. This allows the development of skills such as critical thinking, problem solving, decision-making and technological understanding (Adamchik and Gunawardena, 2003, pp. 96-99).

3.2 Model-Driven Development

Model-Driven Development is a software engineering technique adapting incremental models through transformations governing software's life cycle (Stahl and Völter, 2006, pp. 428). Thanks to such a technique any changes identified are not made to the most recent model but are instead applied to the model holding information relevant to said changes, the change is then spread throughout all other models through semi-automatic transformations (Silva, Barbosa and Maldonado, 2011, pp.F4E-1,F4E-6).

According to (Mellor, Clark and Futagami, 2003, pp. 14-18): "A model is a coherent set of formal elements describing something built for some purpose that is amenable to a particular form of analysis". Model transformations are performed by mapping functions representing knowledge once retained just by the expert (Mellor, Clark and Futagami, 2003, pp. 14-18). Such transformations are automatically or semi-automatically executed (Selic, 2003, pp. 19-25).

What this approach intends to do in the long term is improve flexibility during implementation, integration, maintenance, trial and simulation, such as portability, interoperability and reusability.

4 PROPOSAL

This work proposes a model driven approach for Learning Objects that do not require the author to have any additional technical knowledge of the deployment platform. This way they can self manage and update learning material, whenever they want and without assistance.

The models employed were developed in Eclipse using the EMF (Eclipse Modeling Framework) plugin. This plugin provides two construction methods: The Ecore Model, a text editor that uses navigation trees, and the Ecore Diagram, a graphics editor similar to the tools used for creating UML class diagrams. Both these methods will generate an XMI file (Group, 2007) which is a specification for diagram interchanges.

Our job begins by identifying key LO construction elements, such as introductions, topics, exercises, activities and bibliographies, leaving everything related to content to one side. Viewing and Navigation elements were previously defined and cannot be modified by authors. In building LOs a DSL (Domain Specific Language) was developed, where authors input content from previously identified elements. The first stage requires authors to be accompanied during their first attempts, until they feel comfortable enough to go it alone.

In this model authors input text, video and images as well as some additional style via tagging. However, we must highlight the fact that such visual aspects are not the authors responsibility but instead that of this proposal. The DSL produces a model stored on a server working as LO definition repository. This model then undergoes a transformation to form a second model (Full LO model - FLO), which includes viewing and navigation details. This is also stored on a server.

Though the existence of this FLO model is unknown to authors, it allows concrete changes to be made without affecting transformation rules. Later on, the FLO undergoes a second transformation with the idea producing the LO on the desired deployment platform. This final transformation towards either one or multiple platforms, depending on available rules.

The generated LO is then also stored on a server before being published using an Apache server. The idea is that the LO be created in html and include jQuery to improve usability. If content requires editing, authors have to use DSL to gain access to the LO, and do whatever is needed from there. At this point the system will check for any specific changes made to the FLO in order to take them into account when it comes to regeneration. Finally, authors can accept the changes that have been made and the LO generation process will start over. As a security measure, the server will backup the previous FLO.

Once the LO has been generated it can be accessed immediately.

5 CONCLUSIONS

Looking at preliminary checks made by professors of the systems engineering dept. at UFPS, we can affirm that LO construction using the model driven approach significantly reduces time and effort required for deployment to different platforms. This is important as the idea is for LOs not to simply be technologically

ISBN: 978-605-64453-2-3

individual mechanisms but also and perhaps most importantly, teaching resources to be used for strategic classroom planning, as well as being able to update material as efficiently as possible without assistance, which is a key factor.

At the moment, this work is in the stage of defining two transformations needed for LO production. On top of that ideal deployment platforms that comply with the university's requirements have to be looked at.

Likewise, there will be the constant task of checking new platforms of which to make transformations, keeping LOs up to date with student trends.

6 ACKNOWLEDGMENT

This project has been made possible thanks to the Francisco de Paula Santander University (Cúcuta, Colombia) and the Systems and Informatics Department.

REFERENCE LIST

Adamchik, V., & Gunawardena, A. (2003). A learning objects approach to teaching programming. Information Technology: Coding and Computing [Computers and Communications].

Çakiroğlu, Ü., Adnan, B., & Akkan, Y. (2012). The Effects of Using Learning Objects in Two Different Settings. TOJET 2012, vol.11 (1).

Conceição, S., & Lehman, R. (2003). An evaluation of the use of learning objects as an instructional aid in teaching adults. 2003 Midwest Research to Practice Conference in Adult, Continuing, and Community Education.

Group, O. M. (2007). MOF 2.0/XMI Mapping, Version 2.1.1, Ed: Object Management Group.

IEEE LTSC (2002). IEEE 1484.12.1 - Learning Object Metadata.

Johnson, J., Chapaman, C., & Dyer, J. (2006). Pedagogy and InnLOtion in Education with Digital Technologies. Internacional Conference on Multimedia and ICT in Education.

Martinez-Ortiz, I., Sierra, J. & Fernandez-Manjon, B. (2009). Authoring and Reengineering of IMS Learning Design Units of Learning. IEEE Transactions on Learning Technologies, vol.2.

Mellor, S. J., Clark, A. N., & Futagami, T. (2003). Model-Driven Development. IEEE Software, vol.20.

Polsani, P. (2006). Use and Abuse of Reusable Learning Objects. Journal of Digital Information.

Pukkhem, N. & Vatanawood, W. (2011). Personalised learning object based on multi-agent model and learners' learning styles. Maejo International Journal of Science and Technology.

Selic, B. (2003). The pragmatics of model-driven development. IEEE Software, vol.20.

Silva, M.A.G., Barbosa, E.F., & Maldonado, J.C. (2011). Model-driven development of learning objects. Frontiers in Education Conference (FIE).

Souza, M. C., Castro, J. A., & Andrade, R. M. (2008). ExpertDSL: a profile to support the scoping of a teaching process of developing learning object oriented models. XIX Simpósio Brasileiro de Informática na Educação.

Souza, M. C., Gomes, D. G., Barroso, G. C., & Souza, C. T. (2007). LOCPN: Color Petri Nets in the production of Learning Objects. Revista Brasileira de Informática na Educação, vol.15.

Souza, M .C., Castro, J.A., & Andrade, R.M.C. (2010). Model-Driven Development in the Production of Customizable Learning Objects. Advanced Learning Technologies (ICALT).

Spektor-Levy, O., & Granot-Gilat, Y. (2012). The Impact of Learning with Laptops in 1:1 Classes on the Development of Learning Skills and Information Literacy among Middle School Students". Interdisciplinary Journal of E-Learning and Learning Objects, vol.8.

Stahl, T., & Völter, M. (2006). Model-Driven Software Development. Great Britain: Wiley.

Wiley, D. A. (2002). Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy.

ISBN: 978-605-64453-2-3