Article accepted in the **4th International Conference on Internet and Web Applications and Services**. May 24-28, 2009, Venice, Italy.
Draft version: See http://www.e-ucm.es/publications/articles.html for updated citation information

Enhancing IMS LD Units of Learning Comprehension

Iván Martínez-Ortiz, José Luis Sierra, Baltasar Fernández-Manjón
Dpto. Ingeniería del Software e Inteligencia Artificial. Fac. Informática. Universidad Complutense, Madrid, Spain

Abstract

IMS Learning Design (IMS LD) *is a powerful and expressive educational modeling language, which is becoming a* "de facto" *encoding and interchange standard for activity-based courses.* IMS LD *expressivity includes functionalities such as the adaptation of the runtime behavior of the courses according to the performance and the specific needs of the learners. But these formally described courses, called Units of Learning (UoLs) in IMS LD, are difficult to understand and reuse by instructors, hindering IMS LD adoption. In this paper we describe how these shortcomings are resolved in* e-LD, *a flow-oriented authoring tool for* IMS LD. *In* e-LD preexisting *designs encoded in* IMS LD *are automatically processed to produce a more understandable presentation with different views, such as a browse-able hyper-textual view or a graphical representation of the dependencies between UoL conditions and other components of the learning design. These views help to enhance UoLs' comprehension by instructors.*

Keywords: *Educational Modeling Languages, Learning Design, Graphical authoring*

1. Introduction

The design of an activity-based e-learning course must address many different and interwoven aspects. Besides the creation of the course's contents, designers must characterize other elements such as the services and tools that support the learning process, the actors who participate in this process, the roles played by these actors, the activities that they must undertake, etc. All these aspects constitute the teaching and learning *methods* adopted in courses. However, the incorporation of these methods into an e-learning platform is a difficult task which requires the collaboration of two communities with very different backgrounds: *Technicians* and *Instructors*. Technicians master hardware and software technologies, but usually they have limited knowledge about the course's topics or educational and pedagogical elements. On the other hand, instructors are specialists in the course's topics, and they can choose the most suitable teaching and learning methods. However, instructors are not supposed to have any special skills in Computer Science and Programming. *Educational Modeling Languages* (EML) can facilitate this collaboration [1].

EMLs are domain-specific languages oriented to describing learning methods. Thus, in an ideal world, instructors themselves might formalize the different aspects of the learning process in a learning design document, while technicians might provide the hardware and software required to automate these learning designs, making them runnable in standardized Learning Management Systems (LMS). However, this idyllic situation is far from being reached. Real-world EMLs must face a flexibility-usability gap, which seriously hinders the idealized scenario: the more expressive and flexible the EML is, the more difficult its use by instructors without expertise in computer science. A good example is *IMS Learning Design* (IMS LD), a very expressive and the *de facto* standard EML [2]. Expressiveness in IMS LD is essential for making the description of a great amount of pedagogies and learning methods possible. The price to pay for describing highly interactive and adaptable courses, called *Units of Learning* (UoLs) in IMS LD, is the inclusion of advanced features in the IMS LD language, which hampers its use. One of these features is the heavy use of references to promote the reuse of different components (e.g. environments, roles, activities, etc.) in different contexts of the learning design. Another one is Level B's *condition system*. Condition systems are a kind of monolithic rule-based system, such as those used in artificial intelligence, and their use has proven difficult even for experts in computer science [3]. Indeed, IMS LD conditions suffer from many of the production and maintenance problems well known in rule-based systems and rule-based programming [4]. Still, conditions are essential to modeling adaptive learning flows, enabling the automatic adaptation of an LMS's behavior to the individual learner's needs [5]. On the other hand, more user-friendly notations (e.g. the visual ones used by the LAMS authoring tool [6]) can either lack the required flexibility, or they must face difficult importation / exportation issues in order to interoperate with IMS LD (and with IMS LD compliant LMSs).

Our main research interest regarding EMLs is to address the flexibility-usability gap described above. In particular we focus our efforts on IMS LD. For this purpose, we have defined a visual notation which maintains the structural aspects of IMS LD (i.e. level A), but which replaces Level B's flexible but cumbersome condition system with a more usable flow-oriented style of expressing activity sequencing [7]. We have integrated this notation in e-LD [8], an authoring tool for IMS LD *UoLs* that we are developing at the Complutense University of Madrid (Spain). When loading a preexisting UoL through its XML representation, this tool produces a high-level view of the associated learning design, which facilitates the instructors' comprehension of this design. Then, using this UoL *browser*, instructors can undertake additional modifications and redesigns by using the tool's edition capabilities. In this paper we examine these comprehension-oriented functions of e-LD.

The rest of the paper runs as follows: Section 2 introduces the e-LD authoring tool. Section 3 describes e-LD's comprehension-oriented functions. Section 4 describes some related work. Finally, section 5 presents some conclusions and lines of future work.

2. The e-LD authoring tool

e-LD is a research authoring tool for IMS LD UoLs, which supports three main functions:

- *Importation*. This function lets instructors load pre-existing IMS LD UoLs into the tool. The aim of the importation process in e-LD is not to automatically produce a fully-operative representation of the original UoL, but only an initial skeleton along with complementary information useful to fully understand the original learning design. Indeed, while static aspects (IMS Level A) and some dynamic aspects (IMS Level B and C) can be automatically imported into the tool, such is not the case with other advanced (condition-dependent) dynamic aspects. This problem is due to the different paradigms used to describe these aspects: a rule-based one in IMD-LD and a flow-oriented notation in e-LD. Thus, instructors must understand the dynamic aspects in the original IMD-LD UoL, and then reformulate them using the e-LD flow-oriented notation. In order to facilitate this task, e-LD integrates the aforementioned browser, whose comprehension functions are examined with detail in the following sections.
- *Authoring*. This function lets instructors edit the description of a UoL using a visual notation (this notation is detailed in [7]). Among other features, this notation supports a flow-oriented style for sequencing activities, which is based on the Unified Modeling Language's activity diagrams [9], and on flow-oriented notations used for the description of workflows in business process management systems [10].
- *Exportation*. This function enables the automatic generation of IMS LD descriptions from e-LD diagrams. The core of this function is an automatic translation of flow-oriented notations into rule-oriented ones (see [11] for more details on this translation).

Therefore, e-LD can be used as a conventional tool for authoring UoL from the beginning, but also as a tool for reusing and redesigning preexisting learning designs, which are: (i) imported in the tool, (ii) then adapted and completed and (iii) finally exported into improved IMS LD descriptions. This reuse and redesign feature is facilitated by the aforementioned browser. For preliminary works on e-LD and details about the tool see [7] and [8]. Next sections discuss the browser's comprehension functions with more detail.

3. The comprehension view in e-LD

When e-LD loads an IMS LD UoL, it can produce a higher-level *comprehension view* of the associated learning design that is much easier to understand by an instructor, and even developers, than its XML representation (Fig. 1). This view comprises two different parts:

- A hypertextual view of the different *facets* of the learning design (learning objectives, roles, environments, services, properties, method, etc). This view replaces the references between aspects (e.g. activities referred from a method description) with browse-able hyperlinks, and also offers a layered model of reading.
- A graphical representation of the interdependencies between the components of the design which can affect runtime behavior.

Next sections present in detail these comprehension functions.

3.1. The hypertextual view

The hypertextual view includes a tab for each facet of the learning design. Each tab gives access to a suitable representation of the corresponding facet (Fig. 2). These representations contain:

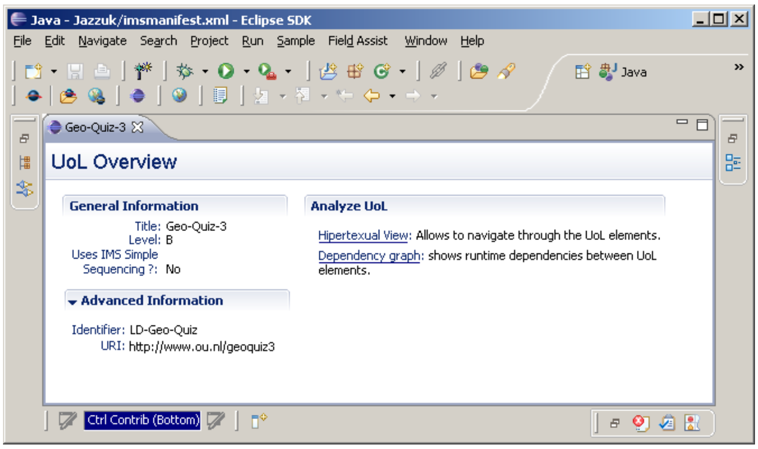

Fig. 1. Entry point to the comprehension view in e-LD

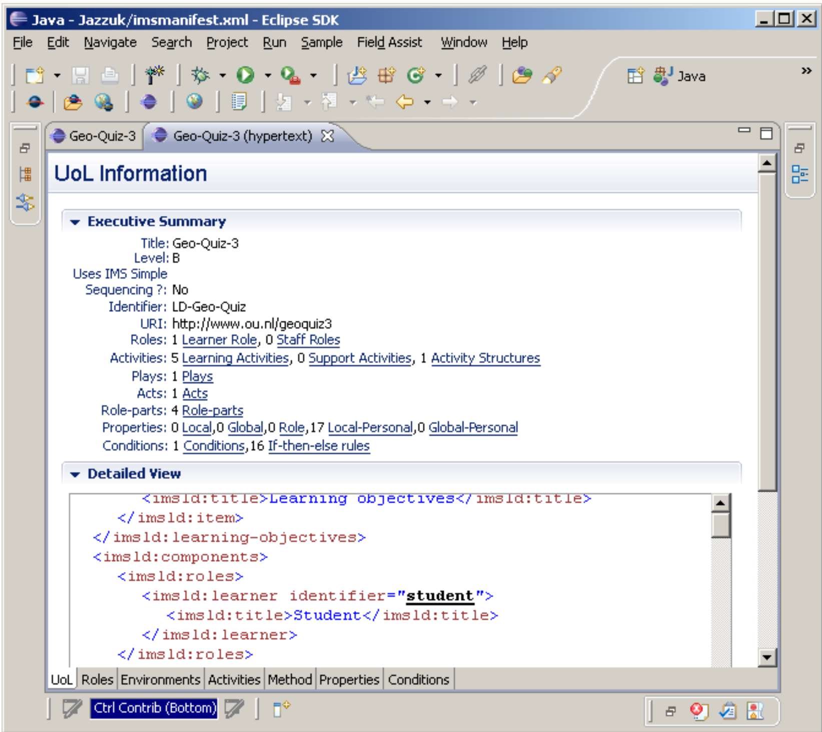

Fig. 2. Hypertextual view

- An index, which is a hyperlinked tree-shaped structure allowing for the exploration of the different facet components. The concrete structure of each index depends on the particular facet. For instance, in the environments facet, this structure is a list of references to the different environments, while in the method facet, it is a tree showing the play – act – role-part structure of this facet. Index entries are interactive, giving access to the associated information element in the design.
- An *inspection frame*, which shows the information associated to the selected node in the index. In turn, this frame has two different tabs: one giving access to an *executive summary* and another one giving access to a *detailed view*. The executive summary filters the relevant information and presents it in a user-friendly form (Fig. 2 shows part of the executive summary corresponding to the *UoL* facet). Likewise, the detailed view directly shows the UoL XML chunk of the selected information element (see Fig. 2).

Finally, the hypertextual view is enriched with hyperlinks, which represent non-hierarchical relationships between the different parts of the design. These hyperlinks are included in both the executive summary and the detailed view (XML), enabling fluent navigation through the different learning design elements and their different *reading levels*. In particular:

- The view enables a top-down, hierarchical reading strategy. The instructors who want to comprehend the design can choose the suitable facet. In each facet, she can select the appropriate information element in the index. The executive summary provides the instructor with the relevant element information while the XML view gives the instructor all the details of the original encoding.
- The view also enables a non-hierarchical reading strategy, driven by the relationships established by cross-referencing in IMS LD. When inspecting an information element that refers to others, the instructor can access those referred elements by clicking the corresponding hyperlinks. Following standard patterns in hypertext and navigation design, the view includes a *go-back* button, and it also enables to follow these links in different tabs/windows.

3.2. The Dependency Graph

The dependency graph is an interactive graphical representation of the dependencies among the design elements related to learning activities sequencing. While the hypertextual view is useful to comprehend the structure of the designs, the dependency graph is useful to understand their behavior. Fig. 3 shows an example of a dependency graph. The user can interact with the graph to select which information is to be displayed and, in particular, to focus on a particular element.

The dependency graph includes the following types of nodes:

- Nodes associated to each IMS Level A element which is relevant from a sequencing perspective: plays, acts, role-parts, activity-structures, learning activities and support activities.
- Nodes associated to *imsldcontents*. In a learning design, content identified as *imsldcontent* can change the values of the design's properties, and therefore influence the runtime behavior of the IMS LD UoL.
- Nodes associated to conditions. Each condition produces three nodes: one representing the condition as a whole, another one associated to the *then* part, and a third one associated to the *else* part.
- Nodes associated to the properties and the property groups of the design.

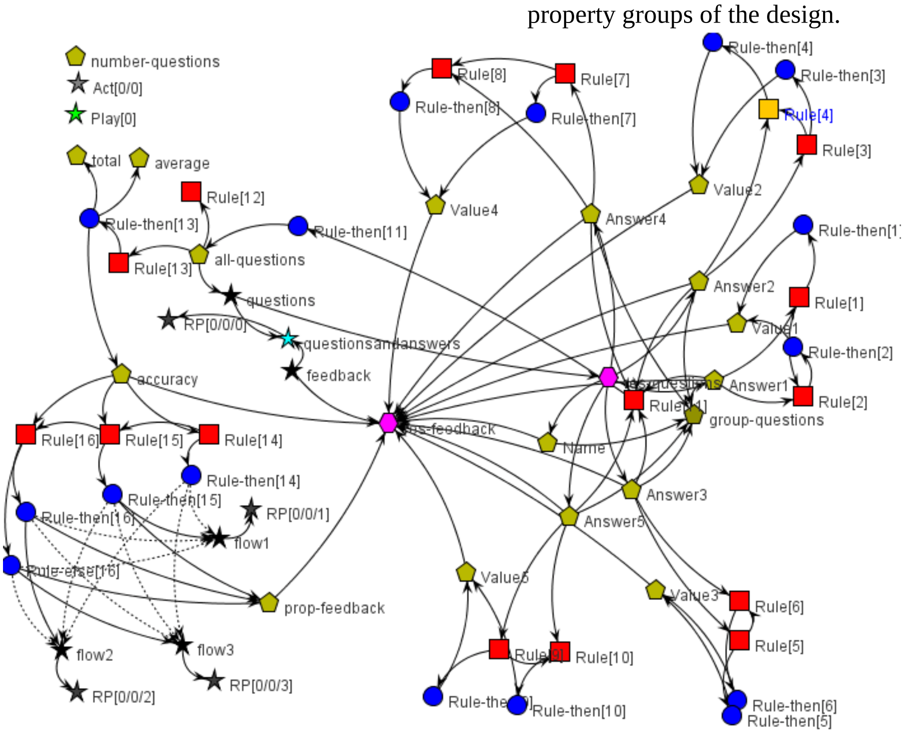

Fig. 3. A Dependency graph in e-LD

The graph’s arcs represent dependencies between these nodes. The types of dependencies considered are:

- *Inclusion dependencies*. These dependencies represent *whole-to-part* relationships. This way, there is an inclusion dependency between each condition and its *then* and *else* parts, and also between each property group and its individual properties.
- *Completion dependencies*. They are dependencies between Level A sequencing elements and properties regarding the completion of the element.
- *Visibility modification dependencies*. A dependency of this type represents the visibility modification of a sequencing element by a condition when such a condition is fired.
- *Property modification dependencies*. This type of dependencies represent the modification of a property done by: 1) a condition's rule, 2) a *imsldcontent* resource, or 3) as the result of a sequencing element completion.
- *Property reading dependencies*. These dependencies represent the acts of reading the values of properties. The reading of a property's value can be done for multiple purposes: to show some evaluation results in a global element, as a completion mechanism for sequencing elements or as part of a condition's rule.

Because the dependency graph includes a lot of dependencies, its graphical representation may become unmanageable in a complex UoL and thus useless for instructors. To avoid this, a set of configurable filters are provided so instructors can control the display of sub-graphs. In addition, each node in the dependency graph is hyperlinked with the hypertextual view. In this way, the hypertextual view and the dependency graph are integrated in a smooth fashion.

4. Related Work

The recovering of higher-level design models from lower-level implementations has been thoroughly addressed in the domain of software development, and, in particular, by the discipline of *program comprehension* [12]. This domain promotes active documentation techniques supporting different levels of reading. In particular, *literate programming*, originally proposed by Donald E. Knuth [13], promotes hypertextual representations of the code, which are interleaved with documentation. The results, called *webs*, are narrations of the programs, in the same way that the programs would be presented in a programming textbook. These documents are marked up for enabling both the assembling of working programs (*tangling*) and the production of documentation printouts (*webbing*). Since IMS LD designs are already represented by (XML-marked) documents, the technique naturally applies to the comprehension of these designs, as promoted by our hypertextual view.

Our dependency graphs are similar in nature to the graphs proposed in [14] to verify the properties of (artificial intelligence) rule-based systems. Since IMS LD conditions are similar to rule-based systems, graphical representations also naturally apply in e-LD. However, notice that dependency graphs in e-LD are mainly oriented to enhancing comprehension and they also involve other different information elements which are relevant to the runtime behavior of IMS LD learning designs.

Comprehensibility of (preexisting) IMS LD learning designs is not an exclusive topic of e-LD, but is also directly or indirectly addressed by the other authoring tools proposed for this language. IMS LD authoring is a hot research topic with a lot of ongoing work currently being done. The work in [15] provide an analysis of the tools needed to work with IMS LD and the difficulties that instructors come across during the authoring of IMS LD learning designs. Some of the available initiatives are: MOT+ [16], Reload LD Editor [17], Prolix GLM [18], ReCourse [19]. Particularly, the last two tools are making a great effort to provide a user friendly graphical notation of learning designs parallel to our work.

5. Conclusions and Future Work

In this paper we have presented an approach to enhance IMS LD comprehension. The approach promotes a hypertextual view to address the barriers to understandability imposed by the heavy use of cross-referencing in IMS LD. It also promotes the use of dependency graphs to improve the comprehension of dynamic runtime aspects of the preexisting learning designs. The approach has been implemented in our e-LD graphical authoring tool for IMS LD.

Currently we have conducted an initial evaluation of the comprehension view included in e-LD, where the approach has been tested in two different scenarios: one involving advanced users (e-learning researchers) and another with PhD students from an e-learning course offered at the Complutense University. As a result of the preliminary experiments we think that the comprehension approach proposed simplifies the process of understanding learning designs during authoring and redesign. Indeed, in these experiments after a short training period, participants with limited background in IMS LD and EML (although with an strong background in computer science) were able to import rather complex IMS LD UoLs in e-LD (in particular, those proposed by the IMS LD's best practices guide [20]), to comprehend the underlying designs, and to rebuild them using the flow-oriented visual notation of e-LD [8].

Next step in the project is to improve the usability of e-LD and test our approach with instructors without a computer science background. As future work we will explore how to use the structures created to enable comprehension in order to improve the automatic importation of UoLs. In particular, we will focus on rules (conditions) clustering and automatic detection of flow-oriented structures. Also we want to explore the applicability of similar approaches with other EMLs (e.g. SCORM Sequencing and Navigation [21]).

6. Acknowledgements

The Spanish Committee of Science and Technology (projects TIN2005-08788-C04-01, Flexo-TSI-020301-2008-19 and TIN2007-68125-C02-01) has partially supported this work, as well as the Complutense University of Madrid (research group 921340, Santander/UCM Project PR34/07 – 15865) and the EU Alfa project CID (II-0511-A).

7. References

[1] Martínez-Ortiz, I., Moreno-Ger, P., Sierra, J.L., Fernández-Manjón, B (2007). Educational Modeling Languages: A Conceptual Introduction and a High-Level Classification, in Fernández-Manjón et al. (eds) Computers and Education: E-learning - from theory to practice. Springer-Verlag New York, Inc.: 27-4.

[2] Koper, R. and C. Tattersall (2005). Learning Design - A Handbook on Modelling and Delivering Networked Education and Training. Heidelberg, Springer Verlag.

[3] Burgos, D., Moreno-Ger, P., Sierra, J.L., Fernández-Manjón, B., Koper, R.(2007). Authoring Game-Based Adaptive Units of Learning with IMS Learning Design and <e-Adventure>. International Journal of Learning Technology, 3(3), 252-268. 2007.

[4] Li, X. (1991). What's So Bad About Rule-Based Programming? IEEE Software 8: 103, 105.

[5] Specht, M.., Burgos, D. (2007). Modeling Adaptive Educational Methods with IMS Learning Design. Journal of Interactive Media in Education 2007/08. Retrieved November 6, 2008 from; jime.open.ac.uk/2007/08.

[6] Dalziel, J. (2006), Lessons from LAMS for IMS Learning Design. Sixth IEEE Int. Conf. on Advanced Learning Technologies (ICALT'06), 2006:. 1101-1102.

[7] Martínez-Ortiz, I.,. Moreno-Ger, P., Sierra, J.L, Fernández-Manjón, B (2008). A Flow-Oriented Visual Language for Learning Designs. 7th International Conference on Web-based Learning (ICWL2008). Jinhua (China), LNCS 5145: 486-496.

[8] Martínez-Ortiz, I., Moreno-Ger, P., Sierra, J.L, Fernández-Manjón, B. (2007). Supporting the Authoring and Operationalization of Educational Modelling Languages. Journal of Universal Computer Science 13(7): 938-947.

[9] Booch, G., Rumbaugh, J., Jacobson, I. The Unified Modeling Language User Guide (2nd Edition). Addison-Wesley. 2005

[10] Aalst, W., Kees, H., Workflow Management: Models, Methods, and Systems, MIT Press. 2004

[11] Martínez-Ortiz,I., Sierra, J.L., Fernández-Manjón, B. Translating e-learning Flow-Oriented Activity Sequencing Descriptions into Rule-based Designs . Submitted to ITNG'09. Draft available at http://www.e-ucm.es/drafts/e-UCM_draft_128.pdf.

[12] von Mayrhauser, A., Vans, A. M (1995). Program understanding: models and experiments, Advances in Computers 40. 1–38.

[13] Knuth, D. E. Literate programming, Computer Journal 27(1) (1984) 97–111.

[14] Ramaswamy, M., Sarkar. S., Chen, Y.S (1997). Using Directed Hypergraphs to Verify Rule-Based Expert Systems. IEEE Transactions on Knowledge and Data Engineering 9(2). 221-237.

[15] Griffiths, D., Blat, J., Garcia, R., Vogten, H., and Kwong, K.L. (2005). Learning Design Tools. In: Koper, R and Tattersall, C, (eds) Learning Design, A Handbook on Modelling and Delivering Networked Education and Training. Springer-Verlag, Berlin Heidelberg, pp 109-135.

[16] Paquette, G., Léonard, M., Lundgren-Cayrol, K., Mihaila, S., Gareau, D. (2006). Learning Design based on Graphical Knowl-edge-Modeling. Educational Technology & Society 9(1), 97-112.

[17] Milligan, C.D. Beauvoir, P., Sharples, P. (2005). The Reload Learning Design Tools. Journal of Interactive Media in Education (Advances in Learning Design. Special Issue, eds. Colin Tattersall, Rob Koper), 2005/07.

[18] Heyer, S., Oberhuemer, P., Zander, S., Prenner, P. (2007). Making Sense of IMS Learning Design Level B: From Specification to Intuitive Modeling Software. In Proceedings of Second European Conference on Technology Enhanced Learning, EC-TEL 2007, LNCS 4753. pp 86-100.

[19] Griffiths, D., Beauvoir, P., Sharples, P. (2008). Advances in Editors for IMS LD in the TENCompetence Project. In Proceedings of Eighth IEEE International Conference on Advanced Learning Technologies, 2008. ICALT '08. pp 1045-1047.

[20] IMS (2003). IMS Learning Design Best Practices and Implementation Guide - Version 1.0 Final Specification. Retrieved November 6, 2008 from: http://www.imsglobal.org/learningdesign/ldv1p0/imsld_infov1p0.html.

[21] ADL, Shareable Content Object Reference Model (SCORM) 2004 3rd Edition Sequencing and Navigation Version 1.0, 2006.