# A Meta-Modeling Approach for Extending the Instructional Design Semantics of Learning Management Systems

Esteban Loiseau, Pierre Laforcade and Sébastien Iksal
*Laboratoire d'Informatique de l'Université du Maine, Avenue Olivier Messiaen, Le Mans, France*

Keywords: Meta-Modeling, Domain Specific Language, Composition.

Abstract: Nowadays Learning Management Systems (LMS) are not restricted to distant learning. Nevertheless, the pedagogical expressiveness of courses designed by teachers is strongly dependent on their knowledge and level of expertise on the LMS they use. The GraphiT project aims to help teachers design pedagogically sound and technically executable learning designs. To this end, we propose to support teachers by providing them with an LMS-specific Visual Instructional Design Language, according to a Domain Specific Modeling approach and tooling. This paper focuses on the abstract syntax of such language. We propose a specific LMS-centered approach for raising the pedagogical expressiveness of their implicit learning design semantics. We discussed how the LMS low-level parameterisations could be abstracted in order to build higher-level building blocks. Based on the Moodle LMS, we present and verify our meta-modeling approach by formalising the abstract syntax of a Moodle-dedicated instructional design language.

## 1 INTRODUCTION

Nowadays, Learning Management Systems (LMSs) are widely spread in academic institutions and are not restricted to online and distant courses. Indeed, they can also be useful during, or in completion of, face-to-face learning sessions (Garrisson and Kanuka, 2004). Nevertheless, the results of a study we conducted with 214 teachers, put forward their heavy form-oriented human-interfaces and tools/content-oriented instructional design lead to reduce their uses. In order to set up complex learning activities, teachers must develop high-level skills for managing and sequencing the LMS's available features and tools. Such skills can be acquired through specific teacher education programs that generally focus on the technical aspects of the platform and not the way they can be used to support pedagogical practices. Because of the multiplicity of educational theories (Ormrod, 2011) and approaches, as well as the lack of tools and processes dedicated to existing LMSs, teachers develop *ad hoc* and individual learning design techniques.

In such contexts, it seems relevant to help teachers understand the instructional design possibilities offered by the LMS at their disposal. This should encourage individual and collective understanding about the pedagogical uses of the targeted LMS. the GraphiT project we present (funded by the French Research Agency) is based on an LMS-centered designing approach. Its main objective is to investigate several Model Driven Engineering (MDE) and Domain Specific Modeling (DSM) techniques to help specify LMS-centered graphical instructional design languages and develop dedicated editors. This paper focuses on the main challenge: raising the pedagogical expressiveness of the LMS learning design semantics by using meta-modeling techniques. Indeed, our past research led us to identify and formalize, according to a specific process, the LMS instructional design semantics as a dedicated metamodel. However, this metamodel needed to be extended, in order to provide the semantics of future learning scenarios. This article presents our current results related to identifying and formalizing the pedagogical semantics for this metamodel extension. Because it is widely spread, we have chosen the Moodle LMS to verify, as a first validation step, the feasibility of our proposal.

We discuss, in Section 2, the current approaches for instructional design and operationalization on LMSs. In comparison with them, we then detail the original position of the GraphiT project regarding MDE and DSM. We also detail the teachers' design


Loiseau E., Laforcade P. and Iksal S..
A Meta-Modeling Approach for Extending the Instructional Design Semantics of Learning Management Systems.
DOI: 10.5220/0005002800720080
In *Proceedings of the 9th International Conference on Software Paradigm Trends* (ICSOFT-PT-2014), pages 72-80
ISBN: 978-989-758-037-6

requirements and needs that we collected during interviews. Section 3 is dedicated to the presentation of our abstraction approach. We also formalize as a metamodel extension, the application of our approach for the Moodle LMS. We then explain and illustrate the 4-levels architecture we propose and illustrate it with a representative example in Section 4. This first validation step is necessary in order to verify that pedagogically sound learning scenarios, that meet the designers' requirements, can be formally described with our model. Finally, we discuss how we plan to use specific DSM tools in order to elaborate the graphical instructional design language from this abstract syntax.

# 2 RESEARCH CONTEXT

The main issue we aim to tackle is studying the pedagogical expressiveness, in terms of possibilities and limits, of operationalizable instructional design languages to specify, i.e. languages allowing to formalize executable learning scenarios that can be automatically set-up into existing LMSs. After discussing the existing approaches and techniques, we present the original approach we chose for the GraphiT project.

## 2.1 Research on LMS Instructional Design

The Technology Enhanced Learning (TEL) research domain has provided many solutions to support instructional design: *Educational Modeling Languages* facilitate the specification of learning situations as formal learning scenarios for delivering and exchanges purposes (Koper and Manderveld, 2004), *Visual Instructional Design Languages* (VIDL) (Botturi and Stubbs 2007) support practitioners communities to communicate and imagine new learning situations and finally, *Learning Management Systems* (LMS) provide operational environments for delivering online learning situations (Muñoz-Merino et al. 2009). These solutions consider LMSs as a final generic environment, providing LMS-independent approaches that focus on the instructional design aspects rather than on how they can actually be operationalized with existing LMSs.

Unfortunately, most design languages do not propose direct binding and operationalization with existing LMSs. Standards such as IMS-LD (Koper, 2006) have therefore not succeeded in being integrated into widely spread LMSs (Burgos et al. 2007). Some researchers have proposed partial transformations from practitioners-centered scenarios towards LMS-centered models (e.g. from PPC to Moodle (Abdallah et al. 2008), from IMS-LD to Moodle (Burgos et al. 2007)). However, these models are based on a subjective and incomplete Moodle metamodel specified by researchers. Such transfor-mations attempts show a semantic gap leading to information loss or incomplete target models. Nevertheless, they have also highlighted the relevance of applying techniques and tools from the Model-Driven Engineering domain.

Recent attempts to operationalize LMS-compliant models have been tried by following a similar binding/translation approach. For example the *Glue! architecture*, including the *Glue!PS* editor (Alario-Hoyos et al., 2012), and the CADMOS editor (Katsamani et al, 2012) are LMS-independent solutions offering LMS deployment features towards the widely spread Moodle LMS (Moodle, 2014). They both realize the deployment by generating a Moodle course backup with all the information, mapping their own data model concepts to Moodle data model concepts. This backup is then imported and deployed within a Moodle course, using the Moodle restoration process. Such approaches result in semantic adaptations and losses during their internal mapping, because of the gap between the instructional design language and targeted LMS' learning design capabilities and features.

For now, the LMS-independent approach therefore reduces the operationalization issues but raises challenges such as specifying a transformation model, capturing the LMS metamodel, reducing the semantics losses during translation and providing a tool that can embed the scenarios into various existing LMSs.

## 2.2 Overview of the GraphiT Project from an MDE and DSM Perspective

Our approach in the GraphiT project, is different to current ones. Indeed, we propose an LMS-dependent architecture. It only focuses on one existing LMS in order to provide an instructional design language that will be specified and tooled according to the future mappings to realize (interoperability of generic learning scenarios is out of our scope). In other words, the main idea is to drive the design by taking into account, at first, the LMS semantics (and then the future mappings).

We do not aim at extending the LMS semantics with new add-ons/plugins, enriching it with more

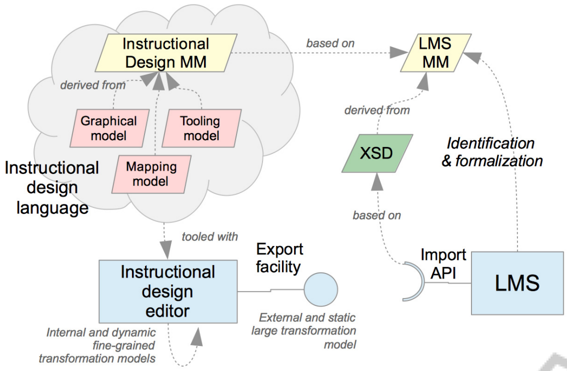


Figure 1: Global overview of the GraphiT architecture.

pedagogical-oriented features. Our objective is to support the design of learning scenarios in conformance with the LMS's semantics (its abilities as well as its limits). We also do not aim at only providing a notation layer, on top of the LMS metamodel. By extending the LMS metamodel we also extend the abstract syntax of the instructional design language, resulting in losing the LMS-compliance format. We plan to restore this format by DSM techniques (weaving and transformation models) we are currently experimenting. We aim at guarantying that learning scenarios could be fully operationalized, into the LMS, without semantics losses. Obviously, our approach has the advantage being LMS-dependence (operationalization support) but it also has the inconvenience of being restricted to one LMS and one of its versions (reengineering cost). We will particularly study how MDE/DSM tools can be useful to reduce that cost.

A global architecture of our solution is illustrated in Figure 1. The LMS instructional design semantics first has to be identified and formalized as a domain metamodel. Then, this metamodel drives the elaboration of an XSD schema that will be used to develop the API. This API will be used through an import facility, accessible by teachers-designers, in their LMS courses. It will take in charge the XML-based scenario parsing and the LMS's databases filling-up. The LMS metamodel will also act as a basis for the elaboration of the visual instructional design language. According to DSM techniques and tools (like the EMF/GMF ones for example), this language will be composed of an abstract syntax, from which the graphical, tooling and mapping models will be driven. The editor will be also developed using the code-generation facilities of DSM tools.

Past works have focused on the LMS meta-model formalization (Abedmouleh et al., 2012). We are currently focusing on the abstract syntax of the instructional design language.

## 2.3 Restoring the LMS Semantics

The produced scenarios need to be compliant with the initial LMS meta-model, in order to be deployed by the API. In order to reach this compliance, we propose to modify the model with two models transformations. The first transformation consist of various, fine-grained transformations during the design process: it will provide several LMS mappings to teachers in order to guide them through their design. They are endogenous transformations because source and target models will both be conformed to the instructional design metamodel. The second transformation will be realized as an export feature that can be used after the design process. This exogenous transformation will produce a scenario/model conformed to the LMS-metamodel.

Unlike other LMS-independent approaches, using transformation models techniques, we are particularly interested in making the underlying mapping models explicit. Indeed, these mappings models are at the center of our approach: their validation, *a priori* of their machine-translation, by experts of the considered LMS, will mainly participate in the learning scenario expressiveness. These explicit LMS bindings will control the translations at runtime. They will guarantee the semantics preservation.

## 2.4 Focus on the Instructional Design Abstract Syntax from a Metamodeling Perspective

The main challenge is to abstract the LMS instructional design semantics enough to provide teachers with higher, pedagogically-sound, design blocks. The LMS expressiveness and limitation therefore have to be overcome, in order to offer teachers instructional design mechanisms that are closer to their practices and needs for specifying and sequencing learning activities. Concretely speaking, the issue is to find a way to specifying the relations between the instructional design language metamodel (that we will refer to as MM-ID) and the LMS metamodel (that we will refer to as MM-LMS).

To this end, we already led several experiments (Loiseau and Laforcade, 2013) on three different approaches: 1/ MM-ID and MM-LMS are two different metamodels without any structural relations, 2/ MM-ID and MM-LMS are the same, the

ID-language being only built as a notation layer on top of the metamodel, 3/ MM-ID is an extension of MM-LMS. The first approach corresponds to the usual way of specifying instructional design languages with its main advantage (expressiveness) but also inconvenience (difficulty to operationalize). The second approach reveals the limits of the concrete syntax expressiveness, because it is only defined by derivation of the abstract syntax. Finally, the third approach is intermediate on all criteria: best expressiveness / LMS compliance ratio. However, it requires a strong metamodeling expertise to reduce the developing cost while restoring the LMS compliance. This approach also highlights the importance to drive the expressiveness (and semantics) extension of the initial metamodel with the binding capacity. This paper focuses on our further results and propositions about this issue.

# 3 EXTENSION BY METAMODELING

According to the practitioners' needs, presented in the next sub-section, we propose to direct the abstraction of LMSs semantics to the LMS uses supporting learners and tutors activities. The following sections present these abstractions in relation with their formalizations for the Moodle LMS (Figure 2). We used the Ecore metamodel format because it will be handled by the EMF and GMF metamodeling tools (EMP, 2014) in order to drive the specification of the instructional design language and the development of its dedicated graphical tool. The metamodel from Figure 2 can be considered as the general abstract syntax of the instructional design language to be developed.

## 3.1 A Practical Overview of Teachers' Requirements

Before tackling the LMS metamodel extension we first have to collect the LMS-specific pedagogical needs and practices. We therefore conducted several theoretical studies from literature sources (Conole et al., 2014), and practical studies with surveys and iterative interviews of 203 teachers and pedagogical engineers. These interviews covered a large variety of Moodle use contexts: online learning, local learning within universities, full distant courses as well as blended learning. Although the GraphiT project deals with different LMSs for guarantying the reproducibility of its results, we propose to focus on the Moodle platform which is the most popular open-source license free LMS. The analysis of these different sources aimed at collecting pedagogical practices or needs, and requirements about the languages and editors to specify and develop.

This study highlighted the fact that practitioners do not really have complex practices to capture, because of the heterogeneity of their Moodle expertise and pedagogical background. Nevertheless they all need to design their course by adapting Moodle's tools to their basic pedagogical uses. Indeed, 88% of of respondents point out the heavy parameterization of tools and resources; 46% requiring an abstract view of the pedagogical uses, in order to help them in select and configure the right implementation activities.

The advanced studies we conducted with pedagogical engineers, allowed us to identify several specific requirements concerning the language and the authoring-tool we will develop. First, they mention the need for the graphical authoring-tool to allow designers to select pedagogical blocks on top of the LMS semantics as well as with Moodle building blocks to compose with. In their mind, the editor will not have to strictly follow a top-down process from abstracted specification elements to implementation one expressed in terms of Moodle; abstractions from Moodle and its own concepts should be mixed up together according to practitioners' expertise about instructional design (**mix of specification and implementation concepts**). Secondly, they are interesting in the idea that mappings from pedagogical design blocks to Moodle concepts can be showed to practitioners (**default mapping**) and adapted if required (**mapping adaptation**). This design approach could help practitioners in the appropriation of the pedagogical constructs and guide them in designing more abstract learning scenario, while mastering the translations into LMS elements.

Another highlighted need is to declare information, within the learning scenario, that does not require to be mapped into LMS concepts point (**declarative non-visible information**). This would allow the designers to write information that is only visible by them and not by students or tutors such as information about face-to-face sessions mixed up with the LMS-centered ones, indications about pedagogical strategies or pedagogical objectives or information about activities to realize on the LMS at a specific runtime moment, according to concrete data (enrolled students, dates, etc.). Finally, another identified need is to facilitate the course sequencing with advanced structures (choices, sequences with

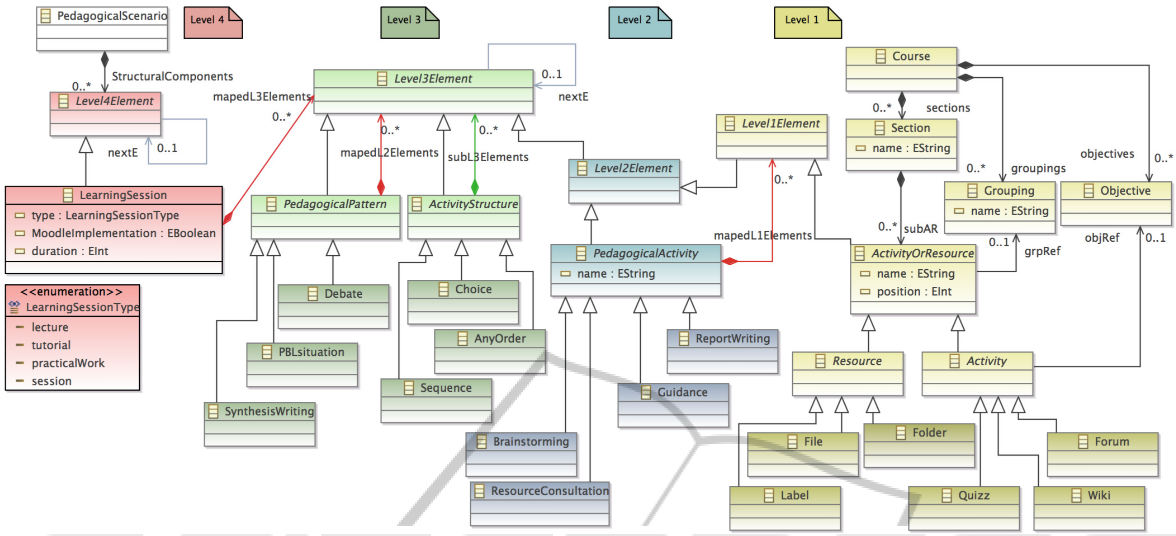


Figure 2: The 4-levels abstract syntax of an instructional design language on top of the Moodle metamodel.

elements showed one-by-one according to their progress (**advanced activity structures**). Indeed, these structures can be designed manually but it requires to parameterize many low-levels and technical-oriented properties (achievements, restrict access conditions...) that teachers would appreciate not to set up by themselves.

## 3.2 Fine-Grained Pedagogical Activities as First Abstraction

The first LMS-abstract building block we propose is the pedagogical activity. This activity is defined as an *abstraction of parameterizations one can realize when using a LMS tool or resource for a specific pedagogical usage*. For example from a single tool, for example a forum, one can design several pedagogical uses, depending on its configuration: provide information to students, set up group work, propose a peer reviewing activity etc.

To be used appropriately, this first abstract block requires a name, a description, and some specific properties that are set by the practitioner, during the design process. For example an exchange activity, involving student communication, could either rely on a forum or a chat, depending on a synchronous property. The mappings will not be limited to the parameterization of a tool. For example with Moodle it will also impact other elements in relation with the tool/resource: grades, objectives, groupings, restriction access and achievements rules, etc.

## 3.3 Large-Grained Pedagogical Activities as Second Abstraction

The second LMS-abstract building blocks are of two kinds. We propose to adapt and integrate some pedagogical patterns and templates from literature (Bergin et al., 2012) (Heathcote, 2006) (for examples as high-level blocks to use and combine for building learning sessions involving instructional strategies: inquiry, problem solving, role-playing, exploration, etc. Although practitioners from our studies do not use to compose with them, we aim at integrating them to encourage some pedagogical reflection and guide designers towards new ways of supporting their didactic and pedagogical objectives. This kind of pedagogical pattern will also have a description of their context, problem and solution uses. They will rely on a mix of structural activities, low-levels blocks (pedagogical activities) and LMS elements to be realized.

In order to ease and assist the practitioners, when assembling and setting-up combinations of activities or resources, we propose a set of usual structural elements (selection, sequence, conditional activities, etc.). These blocks will be composed of other blocks, from high or low levels, including themselves. In the case of Moodle they will be concretely translated into complex combinations of labels (stating the structure kind and use for users) and shifted content (move left/right Moodle feature) according to the activity structure components in the learning scenario. After various translations and mappings until reaching the LMS low-level elements, all its content parts will be parameterized (restrict access, visibility, achievement...) with appropriate properties in order to set up the desired behavior.

### 3.4 A 4-Levels Abstract Syntax

The global architecture we propose for the abstract syntax of the Moodle-centered instructional design language is composed of four levels. Figure 2 illustrates our proposition.

Level 1 fits to the Moodle metamodel. Readers have to consider the Figure 2 illustration (right part) as an incomplete representation of the whole metamodel. Indeed, in order ease the comprehension, only the important structural relations and concepts are depicted[1]. *Level 1 elements* (restricted to the Moodle *activities* – the Moodle name given to the tools - and resources) can be directly used by teachers and parameterized for building a learning session. From the Moodle metamodel point of view, these elements require a global *Course* and a *Section* container to be attached in. In the extended metamodel, they will be specified at first as children of *level 4 elements*. The large model transformation, after the design process, will deal with producing a model in full-compliance with the Moodle metamodel: creation of the global *Course* instance, of its *Section* instances, attachment of all the corresponding Moodle elements according to the orders and positions deductible from the source scenario.

Level 2 (part 2) corresponds to our first high-level blocks about pedagogical activities. They are composed of *Level 1 elements*, i.e. Moodle activities and resources. Level 3 (part 3) captures the second abstract blocks about pedagogical patterns and activity structures. The first one will be composed, after automatic design-time models transformations, by *Level 3 elements,* i.e. elements from levels 1-to-3, including structural activities and other *Pedagogical Activities*. The activity structures are also composed of *Level 3 elements* but their content will be specified by teachers-designers during the design process. Indeed, there is no default content to obtain by models transformation. Finally, the fourth level (part 4) is the contextual level, focusing on the global structure of the learning session, in relation to the different face-to-face, complementary or distant sessions.

Such *Level 4 elements* rely on the Moodle *section* concept. Indeed, Moodle only proposes some sections into the space of the course for aggregating the tools and resources. However, designers have at their disposal an *indentation* feature (*position* property in the Moodle metamodel) to shift activities and resources in order to visually indicate their collective relationships. This *position* property will be used by the dynamical mappings, in order to position the corresponding elements in accordance to the source element position in the global learning scenario.

The composition-relations, annotated with a (1), indicate that the content will not be showed in the future concrete syntax (notation) as nested elements but will be shown in another sub-diagram where the parent container will play the role of the root canvas. On the contrary, the composition annotated with a (2) symbol, indicates that content will be showed as nested elements of the parent container in the same diagram. Finally, the *nextE* reflexive relation allows, by inheritance, to provide a previous/next information to sequence the various elements within their dynamic pedagogical context (the ordering concerns the child elements sharing a same *Level Element* parent).

The future authoring-tool will directly propose to practitioners the *level-4 elements* in the tool palette. Indeed, these elements are necessary to map to Moodle sections in order to sequentially structure the course skeleton. Sessions that do not rely on Moodle features can also be described if designers need an overall view of a global module/course larger than the ones involving the use of an LMS. Other *level-4 elements* will then open an empty sub-diagram when double-clicked. It can then be used to arrange *levels 3-to-1 elements* from the new palette. Indeed, practitioners can then choose the method (top-bottom, bottom-up), the description level (specification versus implementation) and the elements to select, combine and adapt. Except activity structures, other *levels 3-to-2 elements* can be opened up as another sub-diagram containing the default mapping to *levels 2-1 elements*. Every mapping can be adapted and modified by deleting/adding new elements (according to those accepted under the parent element) or modifying the elements properties.

The leaf meta-classes from figure 1 (dark elements) sketches some examples of future elements. For ease of reading, we choose not to show these attributes. However, each of them owns specific properties in accordance with the different in-progress formal specifications we are studying

1 A global overview of the Moodle 2.4 metamodel we captured can be retrieved at http://www-lium.univ-lemans.fr/~laforcad/graphit/wp-content/uploads/2014/05/Moodle-2.4_GeneralMM.png

about the Moodle instructional design semantics, pedagogical activities and patterns, and activity structures.

The current abstract syntax proposition still has to be improved in order to allow the declaration of didactical objectives to the various *Level 4-to-1 element*s. Such objective will be mapped into Moodle *Objectives*, attached to the root *Course* and referenced by the direct or indirect corresponding *Level 1 elements*. Similarly, roles or groups have to be included in order to allow the division of labor in the learning scenario. Mappings to the Moodle concepts of *Group* and *Grouping* will be studied.

Our 4-Levels architecture meets the practitioners' requirements depicted in section 3.1. from a static perspective. The dynamical aspects will be tackled by the transformations models and are out of the scope of this article.

## 4 FIRST VALIDATIONS

The validation of our proposal requires several steps. First, we need to verify that our 4-levels metamodels can be used, in a declarative way, to formalize a set of learning scenarios, identified as relevant use-cases. Because we have not yet integrated the automatic execution of models transformation into the EMF-based editor, we specified the different default mappings by defining them directly using the “add child” service of EMF-based editor (manual definition). Then, we will still have to verify that the semantic meaning is maintained when the automatic weaving/ transformations models will be added. Finally, the graphical aspects of the editor (usefulness, user-friendliness, etc.) will have to be validated when the final concrete syntax will be specified and developed from our abstract syntax architecture proposal. Even though we have extended our research, this article is mainly focused on the first step of this project.

To this end, we propose to illustrate our proposal by formalizing one of the simple and representative use-cases for the Moodle LMS. First, we propose a brief textual description of this use-case and then, the equivalent specification with the dedicated metamodel we proposed in section 3 (Figure 3 is a caption-screen of the EMF-tree-based model editor, annotated to highlight the elements' levels).

The learning scenario is composed of two learning sessions. The first one is a *lecture* session for which the teachers simply want to provide *Resource consultation* activity that contains their lecture presentation material. This pedagogical activity has the *quantity* property set to “one” and the *location* one set to “local”. These properties will lead the dynamic mapping process to propose the *File* Moodle element. The learning scenario then continues with a face-to-face *practical work* sessions in a room with computers. The teachers would like to use the Moodle platform for supporting the pedagogical pattern “*Write a synthesis*” with the *collaborative* property set to “true”. This pattern is automatically mapped to be composed of a *sequence* activity structure embedding 4 sub-components. The first one is another *Resource consultation*. This time, the properties set to “several” (*quantity*) and “local” (*location*) by the teacher will lead the transformation process to add a *Folder* tool. The second sub-element is a *Brainstorming* pedagogical activity. Its *orientation* property, set to “discussion”, leads to propose a Forum tool. Similarly the third sub-element is *Report writing* activity leads to a *Wiki* tool because of the *collaborative* property set to “true”. Finally, the fourth sub-element is a *Guidance* activity that aims at reminding the teachers to evaluate the synthesis in the wiki. The *public* property set to “tutor”, leads the mapping process to make the corresponding *Label* invisible (*visible*=”no”) to students (it will be only visible to the teacher).

The teacher can change the activities properties at any time, leading to other mapping adaptations. For example, by changing the *collaborative* property of “*Write a synthesis*” to “false”, the default values for the sub-components 2 and 3 properties in relation to individual work will be changed to new mappings for individual-oriented Moodle tools. The teacher can also manually delete the mapping elements, re-arrange their order, or add extra elements. Figure 3 shows a global overview of the learning scenario elements, including all the automatic mappings, according to the various properties and values (not depicted within the figure).

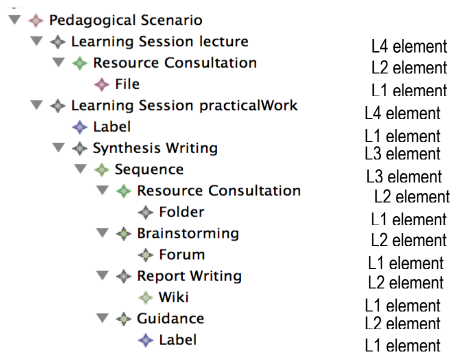


Figure 3: Example of learning scenario composed of elements from the 4 levels.

# 5 CONCLUSIONS

This paper proposes a metamodeling approach for raising the pedagogical expressiveness of learning design semantics of existing LMSs. To do so, we propose to extend the LMS metamodel with specific concepts, properties and relations, in order to meet practitioners' requirements. We discussed how the LMS low-level parameterizations could be abstracted in order to build higher-level building blocks. Based on the Moodle LMS, we present and illustrate our approach, by formalizing the abstract syntax of a Moodle-dedicated instructional design language, following a specific 4-levels architecture. Based on one illustrated use-case, we discuss how we validate, as a first step, our metamodel extension to formally describe Moodle-specific learning scenarios. Such abstraction of LMSs semantics may be a promising approach to develop a new generation of LMS-centered learning design languages, enabling teachers to develop pedagogically sound and technically executable learning scenarios.

The complete version of our metamodel proposition is currently exploited to specify a concrete syntax (graphical notation), a palette and mappings models, in order to develop the final authoring-tool. Because of our former experiences with EMF/GMF frameworks, we will also have to pay attention to the abstract syntax adjustments, required in order to realize specific visual representations.

We are also currently experimenting different frameworks for weaving and transforming models. Indeed, the different default mappings to realize during the design, require a contextualized transformation model to perform. We are studying weaving tools that will allow us to specify the mappings and automatically generate these transformation rules (during the design process). Also, in our approach, the 4-levels extended metamodel will not allow to serialize the future learning scenarios in conformance with the LMS format (source metamodel): a global transformation is required to restore this conformance. This transformation will be realized as an export feature from our authoring-tool.

# ACKNOWLEDGEMENTS

This article is part of the GraphiT project, a 42-months funded project of the French research agency. We thank the various individuals that participated to the interviews and surveys. We also thank the project members involved in the identification and formalization task that helped us for this publication.

# REFERENCES

Abedmouleh A., Oubahssi L., Laforcade P., Choquet C., 2012. An analysis process for identifying and formalizing LMS instructional language. In Proceedings of the 7th International Joint Conference on Software Technologies (*ICSOFT 12)*, Rome.

Abdallah, F., Toffolon, C., Warin, B. Models transformation to implement a Project-Based Collaborative Learning (PBCL) Scenario: Moodle case study. In *Proc. 8th IEEE International Conference on Advanced Learning Technologies (ICALT 08)*. Jul. 2008. pp. 639–643.

Alario-Hoyos, C., Muñoz-Cristóbal, J.A., Prieto-Santos, L.P., Bote-Lorenzo, M.L., Asensio-Pérez, J.I., Gómez-Sánchez, E., Vega-Gorgojo, G., Dimitriadis, Y. GLUE! - GLUE!-PS: An approach to deploy non-trivial collaborative learning situations that require the integration of external tools in VLEs. *Proceedings of the 1st Moodle Research Conference.* 77-85. Heraklion, Crete-Greece, September 2012.

Bergin J., et al., Pedagogical Patterns: Advice For Educators, Joseph Bergin Software Tools, 2012.

Botturi, L., and Todd, S., (Eds.). 2007. *Handbook of Visual Languages for Instructional Design: Theories and Practices*. Information Science Reference.

Burgos, D., Tattersall, C., Dougiamas, M., Vogten, H., and Koper., R., 2007. A First Step, Mapping IMS Learning Design and Moodle. In *Journal of Universal Computer Science* 13 (7). pp. 924-931.

Conole G. et al., "Mapping pedagogy and tools for effective learning design", *Computers & Education*. Vol 4. 2004. pp. 17–33.

Eclipse Modeling Project. Official Website http://www.eclipse.org/modeling/ Retrieved from april 2014.

Garrisson, D.R., Kanuka, H. Blended learning: Uncovering its transformative potential in higher education. *The Internet and Higher Education.* Vol. 7. 2004. pp. 95–105.

Heathcote, Elizabeth A. Learning design templates – a pedagogical just-in-time support tool, in Minshull, Geoff and Mole, Judith, Eds. *Designing for Learning*. Chapter 4, pages pp. 19-26. JISC Development Group. 2006.

Katsamani M., Retalis S. and Boloudakis M. Designing a Moodle course with the CADMOS learning design tool, *Educational Media International*. 49:4. 2012. pp. 317-331.

Koper, R., and Manderveld, J., 2004. Educational modelling language: modelling reusable, interoperable, rich and personalised units of learning.

In *British Journal of Educational Technology* Vol 35 N°5. Pp 537-551.

Koper, R., 2006. Current Research in Learning Design. In *Educational Technology & Society*, 9 (1), pp.13-22.

Loiseau E., Laforcade P. Specification of learning management system-centered graphical instructional design languages - A DSM experimentation about the Moodle platform. Proceedings of the 8th International Joint Conference on Software Technologies (*ICSOFT 13)*, Reykjavik (Iceland), 29-31 juillet 2013.

Moodle. Official Website, https://moodle.org/ Retrieved from april 2014.

Muñoz-Merino, P.J., Kloos, C.D., and Naranjo, J.F., 2009. Enabling interoperability for LMS educational services. In *Computer Standards & Interfaces* 31 (2009). pp. 484-498

Ormrod, J.E. *Human Learning*. Pearson College Division. 2011.