Turning recurrent uses of e-learning tools into reusable pedagogical activities

a Meta-Modeling approach applied to a Moodle case-study

Esteban Loiseau[1], Pierre Laforcade[1] and Nour El Mawas[1]

[1]*LIUM (Computer Science laboratory), University of Le Mans, France*

{esteban.loiseau, pierre.laforcade, nour.el_mawas}@univ-lemans.fr

Keywords: Instructional Design, Visual Instructional Design Language, Learning Management System, Modeling and Meta-modeling

Abstract: The pedagogical expressiveness of designed courses using Learning Management Systems (LMS) is strongly dependent on the teachers' knowledge and expertise about how using the LMS at their disposal. The GraphiT project aims to help teachers in focusing on the specification of pedagogically sound learning scenarios that can be technically executable for automatically setting-up the LMS. We propose to provide teachers with LMS-specific instructional design languages and editors. Such objective requires, at first, to formally capture LMSs implicit learning design semantics. We already tackled this challenge by proposing a dedicated method of identification and formalization as a metamodel. Secondly, the LMS semantics has to be raised in order to enrich the pedagogical expressiveness of the produced models. This paper deals with the proposition of a specific LMS-centered approach for abstracting the LMS low-level parameterizations and turning them into higher-level pedagogical building blocks. We chose to present and illustrate our propositions about the Moodle LMS. We focus on the first abstraction level. It consists in identifying some pedagogical activities according to recurrent uses designers make by handling the Moodle activities.

1 INTRODUCTION

Nowadays, *Learning Management Systems* (LMS) are widely spread in academic institutions. These LMSs are not restricted to distant courses but are also useful during or in complement to face-to-face learning sessions (Garrisson & Kanuka, 04). Nevertheless, the results of a study we conducted with 203 teachers, put forward their heavy form-oriented human-interfaces and tools/services-oriented course design lead to reduce their uses. In order to set up complex learning activities, teachers must develop high-level skills on how to use the existing LMS: how and when managing and sequencing the available features and tools for pedagogical purposes. Such skills can be acquired through specific teacher education programs, often focusing on the features and technical aspects of the platform, but few courses deal with how to design pedagogically sound learning situations on the LMS (specific learning design). Because of the multiple educational theories (Ormrod, 11) 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 context, it is relevant to help teachers in focusing on pedagogical aspects and their instructional design setting-up for the specific LMS they have at their disposal. Whereas improving their know and know-how about the platforms features, a focus on the instructional design possibilities and how they can rely on the platform features should encourage individual and collective understanding about the pedagogical uses of the targeted LMS.

We on purpose propose an LMS-centered designing approach in opposition to the usual platform-independent approaches (Alario-Hoyos et al., 12)(Katsamani et al., 12). The GraphiT project (funded by the French Research Agency) is based on this approach. Its main objective is to investigate Model Driven Engineering (MDE) and Domain Specific Modeling (DSM) techniques to help specifying LMS-centered graphical instructional design languages and development of dedicated editors. This paper deals with one central challenge: raising the pedagogical expressiveness of the LMS learning design semantics by using MDE techniques. Nevertheless, the following propositions are mainly argued from an instructional design viewpoint. MDE techniques are only depected as concrete tools for applying our propositions.

To this end, we detail in Section 2 our research context, including the presentation of the GraphiT project, as well as a position of our current results. The section 3 is dedicated to a survey and series of interviews we conducted with designers in order to collect needs and requirements for the Moodle LMS. A global presentation of our abstraction proposition is done in section 4: a 4-levels abstract syntax, formalized as a metamodel, for the future learning design language. Section 5 focuses on the first-level. This section 5 includes the proposition of a specific method to identify the pedagogical activities and their bindings towards LMS's tools. We also use the concrete weaving language we developed to formaly capture these bindings. Section 6 validates the concrete use of our technical propositions.

2 RESEARCH CONTEXT

2.1 LMS and instructional design

LMSs development is usually based upon an educative theory rationale, or some specific pedagogical approach. For example Moodle claims a socio-constructivist pedagogy philosophy (Dougiamas & Taylor, 03). Most spread LMSs generally follow such an orientation because of the various production and communication tools provided. LMSs are the activity-centered evolution of former learning-objects-centered TEL-systems. Indeed, current LMSs provide designers with some numerous functionalities that can be used to realise various learning activities and are not restricted to provide some resources access to students.

Nevertheless, activity-centered standards like the *de facto* IMS-LD fail to integrate existing LMSs. Experiments on extending Moodle to import IMS-LD learning scenarios proved that adapting existing LMSs requires some complex and heavy re-engineering (in particular integrating a dedicated runtime-engine) in order to overcome the limits of the platform features and semantics (Burgos et al., 07). Educational Modeling Languages (Berggren et al., 05) fail to provide a support for operationalizing EML-conformed learning scenarios into existing LMSs. For now widely spread LMSs like Moodle still do not propose an IMS-LD compliance.

Moodle proposes its own format for importing questions into quizzes. Our idea is to generalise it to the whole instructional design aspects. Similarly to the SCORM compliance about Learning Objects, the rationale of the GraphiT project is based on the idea that LMSs should make explicit their learning design format in order to ease the import/export of compliant learning scenarios, and, in addition, to encourage LMS-dedicated instructional design editors.

2.2 Overview of the GraphiT Project From an MDE and DSM Perspective

The project main goal is to study the possibilities and limits about the pedagogical expressiveness of operationalizable languages. The project methodology consists in exploring how *Model Driven Engineering* and more particularly *Domain Specific Modeling* techniques and tools can be relevant and useful to achieve this goal.

Similar research works about pedagogical sound and executable learning desigh editors follow different approaches. For example the *Glue! architecture*, including the *Glue!PS* editor (Alario-Hoyos et al., 12), and the CADMOS editor (Katsamani et al., 12) are LMS-independent solutions offering an LMS deployment feature towards the most spread and used Moodle platform (Moodle, 14). They both achieve 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 semantics adaptations and semantics losses during their internal mappings because of the gap between the instructional design language and the specific learning design capabilities and features of the targeted LMS. Other works (Abdallah et al., 08) shows that transformation models techniques from the MDE theories and tools can be useful to translate a designer-centered and LMS-independent learning scenario to a specific LMS one. Nevertheless, they also highlighted the complex transformation model to specify, the LMS metamodel to capture, the semantics losses during translation, and the requirement of an LMS- dedicated tool for embedding the scenarios into the LMS.

Our approach is different: we propose an LMS-dependent architecture that 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. Our idea is to conduct the platform abstraction in accordance with the formalisation of future learning scenarios. We do not aim at extending the LMS semantics with new add-ons/plugins enriching it with more pedagogical-oriented features. Our objective is to support learning scenarios specification in conformance with the LMS 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. Past experiments in (Loiseau & Laforcade, 13) showed that the best solution (expressiveness / LMS compliance ratio) consists in extending the LMS metamodel. However, it requires a strong metamodeling expertise to reduce the developing cost while restoring the LMS compliance. This solution 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.

By extending the LMS metamodel we also extend the abstract syntax of the instructional design language and then losing the LMS-compliance format. We plan to restore it by DSM techniques (weaving and transformation models). We aim at guarantying that learning scenarios could be fully operationalized into the LMS without semantics losses. Obviously, our approach can take advantage of this LMS-dependance but it has also the inconvenience to be restricted to one LMS and one of its versions.

A global architecture of our solution is illustrated in Figure 1. The LMS instructional design semantics has first to be identified and formalized as a domain metamodel. This metamodel drives the elaboration of an XSD schema that will be used as a format reference for the API to develop. This API will be used through an import facility available to 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 derived. The editor will also be developed using the code-generation feature of DSM tools. The produced scenarios have to be compliant with the initial LMS meta-model to be deployed by the API. We propose then to run two kind of models transformations. The first one will consist of various, fine-grained transformations during design-time: it will show some LMS mappings to teachers-designers in order to help and guide them in the design process. The second transformation, unique and large, will be used as an export feature (after design-time).

The main challenge of this project is to abstract enough the LMS instructional design semantics to provide teachers with some pedagogically-sound

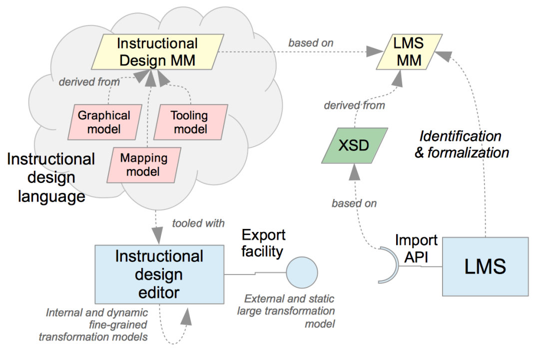

Figure 1: Global overview of the GraphiT architecture.

higher design blocks. The LMS expressiveness and limits have to be overcome in order to offer teachers some instructional design mechanisms closer to their practices and needs about specifying and sequencing learning activities.

Although the GraphiT project deals with different LMSs for guarantying the reproducibility of its results, we on purpose propose to focus on the Moodle platform which is the most popular open-source Learning Management System.

3 COLLECTING TEACHERS-DESIGNERS' REQUIREMENTS

At first, we conducted several theoretical, from literature sources (Conole et al., 04), and practical exchanges with pedagogical engineerers in order to sketch our proposition orientations. We then decided to conduct a larger survey with complementary interviews to verify our initial assumptions, to collect some feebacks about our project orientations and positions, and to identify more precisely end-users practices, needs and learning design tools requirements about the Moodle LMS.

3.1 Global overview of the survey

We conducted an online survey that was diffused through international french-speaking higher education institutions during a 4-weeks period. This survey addressed teachers and pedagogical engineers using existent LMSs. The survey was composed of 21 mandatory questions, most of them accepting multiple answers. Some questions were conditionned to the good selection of previous specific answers. For example the first 8 questions (relative to the global design of courses) are LMS-independant, whereas the other ones are only available to people using the Moodle LMS (the LMS we wanted to focus on). We received and analysed 208 results. Because of the multiple answers requiring to depict all results, we only sketch here the most noticable and relevant points in relation to the focus of this paper.

74% of answerers use an LMS in addition to their face-to-face courses (32% of them only do that), 52% for distant courses, 37% during the face-to-face sessions. Main uses of the LMS concern the document transmission (91%), collect of works (52%), support for collaborative activities (47%), (auto/teacher) evaluations (47%), and 58% consider that the LMS encourage them to put inro practice new pedagogical practices. On average, half of answerers considers having exploring the LMS alone. Those who not consider themselves as novices (56%) states having deepen their LMS knowledge by their own at 73%.

Although half of Moodle users consider that the global HMI of a course is easily understandable, only 33% consider that the form-oriented parameterization screens are understandable. From a learning design perspective, they sketch all (38%) or part (37%) of the learning scenario before setting-up the equivalent course upon Moodle. 43% of this sub-population have met some difficulties during this manual step and have been constraint to adapt their initial scenarios and intentions (12% fail to adapt the scenario). A majority of Moodle designers use the basic functionnalities like the move left/right (64%), the hide/show (84%) parameters. Half of answerers grade students productions and use Moodle's groups and groupings when required. More than half of them (62%) use the restrict access settings but only 34% the activity completion. 15 of 22 Moodle standard activities/functionnalities (note that institutions can remove/add these blocks) are misknown by an average of 50% (sometimes more) of answerers whereas the 7 others are regularly used. The *Forum* is largely prefered to the *Chat* to foster communication. For the realization of exercices, *Assignment* (47%) and *Quiz* (37%) are prefered to *Hot Potatoes* (15%) or *Lesson* (19%). The *Wiki* is the most preferred collaborative tool (23%) among others (*Journal* 8%, *Workshop* 8%).

3.2 Most relevant points from interviews analysis

From most relevant answerers that agree to be contacted we realize 20 one-to-one interviews, mostly by distant devices. interviewees were selected because of their instructional design expertise about the Moodle platform.

They agree that Moodle is useful for simple pedagogical objectives but is time-consuming for elaborating more complex learning situations. Parameterizations screens are considered too much complex and difficult to handle. These screens mix pedagogical and technical parameters, requiring to test and observe the pedagogical implications of all combinations. Some interviewees state that they encourage to use default parameters and then, hinder the setting-up of more complex activities.

A majority of interviewees accept the idea of both an external learning design editor dedicated to Moodle and a *import* bloc available through the Moodle internal design space to automatically set-up the course (the external feature allowing offline designs and the graphical notation helping in visualizing the scenario at design-time). They approve the approach emphasising its relevance if templates or concrete cases about pedagogical uses of Moodle tools can be handled within the editor. They highlight the need for a language/editor covering large pedagogical uses but without being too generic. Some of them consider important to continue using the editor for adapting the scenario after the import step although they agree that a round-trip use of both editor and Moodle can be an obstacle.

Most regrettable point highlighted by practitioners is that they do not really have pedagogical practices to capture, because of the heterogeneity of their Moodle expertises and pedagogical backgrounds. Nevertheless they have in common to think about Moodle tools according to their basic pedagogical uses. Indeed, they all point the heavy parameterizations of tools and resources and the need for having an abstract view of what are the pedagogical uses in order to help and guide them in selecting and configuring the right implementation activities.

3.3 Requirements for our language/editor

From all these practitioners feedbacks we listed some specific requirements for our Moodle language/editor to develop. First, they mentioned 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 (**specification and implementation concepts mix**). 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 design point highlighted (**declarative non-visible information**) is about the possibility to design and declare within the learning scenario some information that do not required to be mapped into LMS concepts or just mentioned as non-visible labels (for students/tutors) for the teacher him-self: information about the face-to-face sessions mixed up with the LMS-centered ones, about pedagogical strategies or pedagogical objectives, about activities to realize on the LMS at a specific runtime moment according to concrete data (enrolled students, dates, etc.). Finally, another design need was to help teachers in sequencing the course in more advanced structures (choices, sequences with elements showed one-by-one according to their progress (**advanced activity structures**). Indeed, these can be done manually but it requires to parameterize many low-levels and technical-oriented properties (achievements, restricted access conditions...) that they would appreciate not to have to set up by themselves.

4 ABSTRACTING THE LMS METAMODEL

According to practitioners'needs one first interesting point to drive the abstraction of LMSs semantics consits in raising the LMS uses supporting learners/tutors activities.

characteristic elements composing usual instructional design language and how they are inter-related: activities, resources, roles, objectives, etc.

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 (Eclipse, 14) 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 part of the general abstract syntax of the instructional design language to be developed. This part focuses on the abstraction of Moodle activities (tools)

4.1 Fine-grained Pedagogical Activities as First Abstraction

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

Because several LMS functionalities can be used for the same pedagogical purpose, we have to find the discriminatory criteria that can guide to identify the right tool and default configuration (as well as the relations to objectives, resources, groupings, etc. that are involved in the right setting-up of the pedagogical activities).

To be used appropriately, this first abstract block requires a name, a description, and specific properties (the former discriminants), set at design-time by practitioners, that will drive the default mapping. 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, it will also impact some other elements in relation with the tool/resource: grades, objectives, groupings, restriction access and achievements rules, etc.

4.2 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., 12; Heathcote, 06) 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 reflections and to guide designers towards new ways to realise their didactic and pedagogical objectives. This kind of pedagogical patterns 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.

In order to ease and assist the practitioners when assembling and setting-up combinations of activities

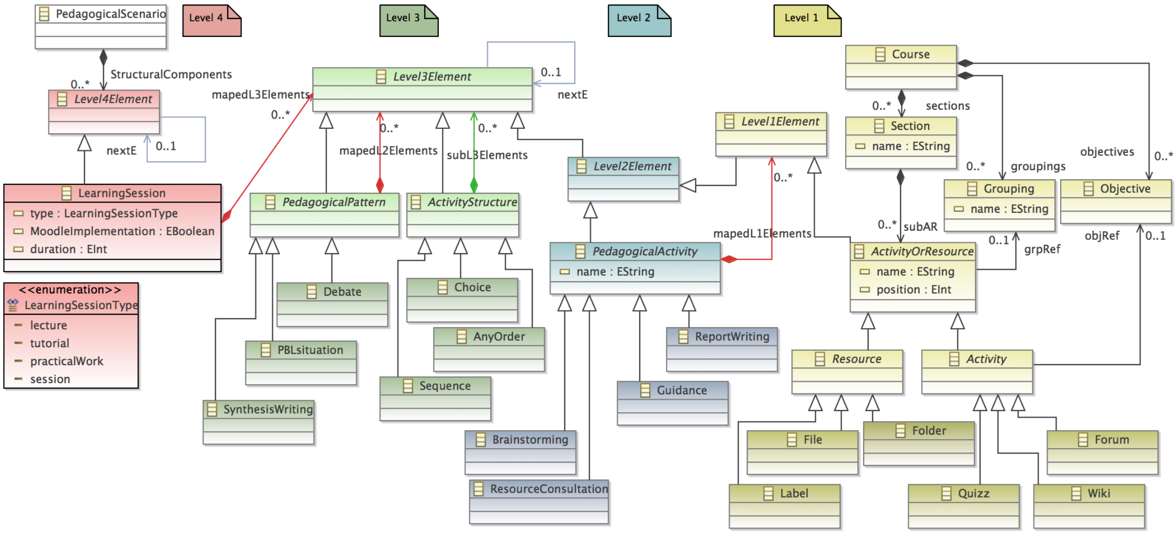

Figure 2: The 4-levels abstract syntax of an instructional design language on top of the Moodle metamodel.

or resources we propose usual structural elements (selection, sequence, conditional activities, etc.). These blocks will be composed of other blocks, from high or low levels, including themselves. Every instructional design language feature some of them. In the case of Moodle they will be concretely translated as 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.

4.3 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 with a graphical representation of the ecore domain model.

Level 1 fits the Moodle metamodel. Readers have to consider Figure 2 as an incomplete representation of the whole metamodel. Only important structural relations and concepts are depicted. *Level 1 elements* (restricted to the Moodle *activities* the Moodle name given to the tools - and resources) can be directly used by teachers-designers 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 to. In the extended metamodel they will be specified at first as child of *level 4 elements*. The model transformation, at post-design-time, will deal about restoring a model in full-compliance with the Moodle metamodel: creation of the global *Course* instance, Section instances, attachment of all the corresponding Moodle elements according to the orders and positions deductible from the source scenario.

Level 2 include our first high-level blocks about pedagogical activities. They are composed of *Level 1 elements*, i.e. Moodle activities and resources. Level 3 captures the second abstract blocks about pedagogical patterns and activity structures. The first one will be composed, after the design-time transformation model, by *Level 3 elements* that includes those 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 during the design-time. Finally, the fourth level 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 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 relations with a red composition 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. Differently, the green composition 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 2 (dark elements) sketches some examples of future elements. They are on purpose not showing their attributes (for ease of reading). However each of them owns specific properties in accordance with the different in-progress formal specifications we are studying 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 didactic objectives to the various *Level 4-to-1 elements*. 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 labour in the learning scenario. Mappings to the Moodle concepts of *Group* and *Grouping* will be studied.

5 FOCUS ON THE FINE-GRAINED PEDAGOGICAL ACTIVITIES

5.1 An identification method

In order to identify ... we followed the two next steps: (1) analysis for each Moodle tools of its recurrent uses, (2)

Rules:

R1 The pedagogical activity name is only from a teacher perspective, if no students are concerned (= with parameter *hide* on).

R2 Tools participating to the realization of the activity are placed in columns.

R3 Discriminating criterion are placed in lines.

R4 Discriminating criterion are expressed as much as possible as a pedagogical question designers have to answer by *Yes* or *No*.

R5 Cells intersecting a discriminating criterion and a tool must embed all answers that can implied to choose this tool (*Yes/No* are both possible if this criterion is not directly discriminant for this tool, i.e. the tool can support both pedagogical cases).

R6 A valid discriminating criterion must cause at least one different answers for one tool.

R7 The matrice is terminated if there is no similar combination of answers for two tools.

R8 *Default* indicators, using {/}, indicates that the related tool will be used by default for future mappings in case of multiple tool choices.

R9 Default indicators can allow to differentiate cells values and can be sued to verify R7; they are not mandatory.

An unachieved matrice indicates to experts that they have two options: add one more discriminating criteria and verify again the rule R7 or

Table 1: Example of identification matrice.

Answering poll	Quiz	Choice	Question -naire	Survey
More than 1 question?	Yes/No	{No}	Yes	{Yes}/No
Feedbacks?	Yes/No	No	No	No
Anonymous?	No	{Yes}	No	Yes/{No}
Standards surveys?	No	No	Yes	No

5.2 Formalization of mappings with our weaving language

5.3 A learning scenario example

We on purpose propose to illustrate our proposal by formalising a very simple but representative learning scenario for the Moodle LMS. We propose at first a brief textual description, then the equivalent specification as a model conformed to the dedicated meta-model we proposed in section 4 (Figure 3 is a screendhot of the EMF-tree-based model editor, annotated to highlight the elements' level).

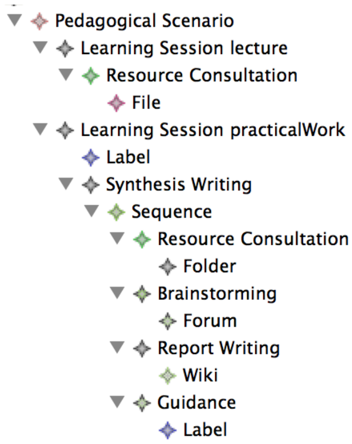

Figure 3: Example of learning scenario composed of elements from the 4 levels.

The learning scenario is composed of two learning sessions. The first one is a *lecture* session for which the teacher only want to propose learners with a *Resource consultation* corresponding to his face-to-face course 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 second learning session is a *practical work* that the teacher wants to realise in face-to-face within a computerized classroom. He would like to use the Moodle platform for supporting a pedagogical pattern "*Synthesis writing*". This pattern is automatically composed of a *sequence* activity structure embedding 4 sub-components. The first one is another *Resource consultation*. This time, the properties set to "many" (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 one is another pedagogical activity *Report writing* leading to a *Wiki* tool because of the *collaborative* property set to "true". Finally the fourth sub-component is a *Guidance* activity that aims at reminding the teacher to evaluate the synthesis in the wiki. Thanks to a *public* property set to "tutor" it leads the mapping process to set the corresponding *Label* to be invisible (*visible*="false") to students (it will be only visible to the teacher).

The teacher can change at any time the activities properties, leading to other mapping adaptations. He can also manually delete the mapping elements, rearrange their order, or add some other 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).

5.4 Formalizing the example

According to our Model Driven approach, we can use model transformations to achieve such mappings. The transformations will be run on demand at design-time, to add mapped elements to the model and populate the sub-diagrams. Such transformations are complex (proportionally to the mapping complexity) and numerous, thus costly to write.

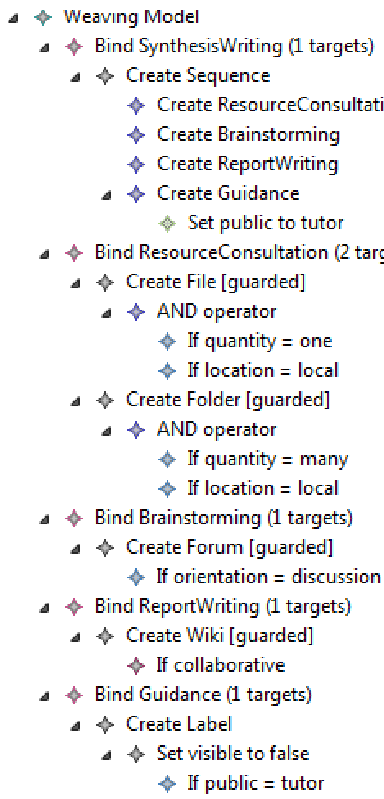

Figure 4: Example weaving model specifying mappings from Figure 3.

We on purpose propose to use model weaving to capture the mapping semantics in dedicated weaving models and automatically generate models transformations. From a practical point a view, thanks to representative specifications from the teaching community, an engineer will model the mappings in a weaving model, using a tree based editor. He can then run a generic *High Order Transformation* (HOT) that will generate the concrete "mapping transformations". These final transformations can then be integrated within the graphical editor to be run at design-time.

The weaving models can be expressed using a weaving language, based on a generic weaving metamodel we designed. This weaving metamodel defines the "syntax" of the mapping/weaving model. Each mapping (or binding) has one *source* element and one or several *targets* (chosen from the extended instructional design metamodel). Targets can have conditions on whether they have to be instantiated or not, attributes can be set to specific values (also with conditions)... Figure 4 is a weaving model, displaying the mapping strategy from the example in section 5.3.

We used languages and tools from the Epsilon project to build a software framework fulfilling our model weaving requirements. Weaving models are edited through ModeLink, a three pane editor displaying the source and target metamodels in side panels (which are the same in our use case). The final "mapping" transformations are expressed using Epsilon Object Language (EOL), and are generated through a Model-to-text transformation using EGL language. This last transformation replaces the HOT traditionally used in model weaving environments.

5.5 Current tooling overview

6 CONCLUSIONS

6.1 A first tools-centered abstraction

This paper proposes a specific LMS-centered approach for raising the pedagogical expressiveness of its implicit learning design semantics. We discussed how the LMS low-level parameterizations could be abstracted in order to build higher-level building blocks. Based on the Moodle application, we present and illustrate our approach by formalising the abstract syntax of a Moodle-dedicated instructional design language following a specific 4-levels architecture. Such abstraction of LMS 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 designs.

The complete version of our metamodel proposition will drive the definition of the concrete syntax model (graphical notation), the palette and the mappings models in order to develop and tool the authoring-tool. Because of our former experiences about the EMF/GMF frameworks, we will also have to pay attention to the abstract syntax adjustments required in order to realise some specific visual representations.

6.2 Other levels and abstraction points

We are also currently experimenting different frameworks about weaving and transforming models (more broadly about models composition). Indeed, the different default mappings during the design-time require a contextualised transformation model to perform. We are studying some weaving tools that will allow us to specify the mappings and automatically generate transformation rules (at design-time). First results have been illustrated within this article.

Also, in our approach the 4-levels extended metamodel will not allow to serialize future learning scenarios in conformance with the LMS format (source metamodel): a global transformation is required to restore this conformance. This transformation will be available as an export feature from our authoring-tool.

ACKNOWLEDGMENTS

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 different project members involved in the identification and formalization task that help us in this publication.

REFERENCES

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

Abedmouleh A., Oubahssi L., Laforcade P., Choquet C.: An analysis process for identifying and formalizing LMS instructional language. In: 7th International Conference on Software Paradigm Trends, pp. 218–223. ScitePress (2008)

Alario-Hoyos, C., Munoz-Cristobal, J.A., Prieto-Santos, L.P., Bote-Lorenzo, M.L., Asensio-Perez, J.I., Gomez-Sanchez, 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. In: 1st Moodle Research Conference, pp. 77–85. Moodle Research Conference, Heraklion (2012)

Bergin, J., Eckstein, J., Manns, M.L., Sharp, H., Chandler, J., Marquardt, K., Wallingford, E., Sipos, M., Völter, M.: Pedagogical Patterns: Advice For Educators. Joseph Bergin Software Tools (2012)

Berggren, A., Burgos, D., Fontana, J.M., Hinkelman, D., Hung, V., Hursh, A., Tielemans, G.: Practical and Pedagogical Issues for Teacher Adoption of IMS Learning Design Standards in Moodle LMS. Journal of Interactive Media in Education, special issue: Advances in Learning Design (2005)

Bertelsen, O.W., Bodker, S.: Activity Theory. In: Carroll, J.M. (eds.) HCI Models,Theories and Frameworks: Toward a Multidisciplinary Science, pp. 291–324.Morgan Kaufmann, San Francisco (2003)

Burgos, D., Tattersall, C., Dougiamas M., Vogten, H., Koper, R.: A First Step Mapping IMS Learning Design and Moodle. Journal of universal computer science 13, 924–931 (2007)

Conole, G., Dyke, M., Oliver, M., Seale, J.: Mapping pedagogy and tools for effective learning design. Computers & Education 4, 17–33 (2004)

Dougiamas, M., Taylor, P.: Moodle: Using Learning Communities to Create an Open Source Course Management System. In: World Conference on Educational Multimedia, Hypermedia and Telecommunications, pp. 171178. Association for the Advancement of Computing in Education, Waynesville (2003)

Eclipse Modeling Project Official Website, `http://www.eclipse.org/modeling/`

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

Heathcote, Elizabeth A.: Learning design templates - a pedagogical just-in-time support tool. In: Minshull, G., Mole, J. (eds.) Designing for Learning, pp. 19–26. JISC Development Group (2006)

Katsamani, M., Retalis, S., Boloudakis, M.: Designing a Moodle course with the CADMOS learning design tool. Educational Media International 49, 317–331 (2012)

Loiseau, E., Laforcade, P.: Specification of learning management system-centered graphical instructional design languages - A DSM experimentation about the Moodle platform. In: 8th International Joint Conference on Software Technologies, pp. 504–511. Scitepress (2013)

Moodle Official Website `https://moodle.org`

Ormrod, J.E.: Human Learning. Pearson College Division, Upper Saddle River (2011).