Formalization of recurrent uses of e-learning tools as reusable pedagogical activities

Abstract—**The pedagogical expressiveness of designed courses using Learning Management Systems (LMS) is highly dependent on the teachers' expertise about the use of the LMS at their disposal.The LMS semantics has to be raised in order to enhance 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 choose to present and illustrate our propositions about the Moodle LMS.**

Keywords—Instructional Design, Learning Management System, Modeling and Meta-modeling, weaving models.

I. Introduction

Learning Management Systems (LMS) are widely spread TEL-system whose uses have been extended from distant courses to face-to-face learning sessions [1]. Nevertheless, the results of a study we conducted put forward the difficulty for teachers to appropriate such complex systems. It is then 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.

For this purpose, we propose within the GraphiT project, an LMS-centered designing approach in opposition to the usual platform-independent approaches [2][3]. The main objective is to investigate Model Driven Engineering (MDE) techniques for supporting the specification of LMS-centered graphical instructional design languages and the development of dedicated editors. This paper deals with one central challenge: raising the pedagogical expressiveness of LMSs learning design semantics using MDE techniques.

To this end, we detail in section 2 a survey and series of interviews we conducted with designers in order to collect needs and requirements for the Moodle LMS. Section 3 focuses on a first-level of abstraction: it includes the proposition of a specific method to identify the pedagogical activities and their bindings to LMS tools. We also use a specific weaving language we developed to formally capture these bindings. Section 4 illustrates our propositions by a concrete learning scenario.

II. Collecting teachers-designers' requirements

We conducted a 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.

A. Global overview of the survey

The online survey was relayed through international french-speaking higher education institutions over a 4-weeks period. This survey is aimed at 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 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 were only available to people using Moodle (the LMS we wanted to focus on). We received and analysed 208 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 are about: document transmission (91%), collecting works (52%), supporting collaborative activities (47%), evaluations (47%), and new pedagogical practices (58%). On average, half the answerers considers having explored the LMS on their own. Those who do not consider themselves as novices (56%) state having deepen their LMS knowledge by their own at 73%. Although half of Moodle users consider that the global user-interface 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 functionalities 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/functionalities 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%).

B. A need for abstraction

From most relevant answerers we realized 20 one-to-one interviews, mostly by distant devices. Interviewees were selected depending on 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. Settings screens are considered too 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 stated that they encourage to use default parameters and then, hinder the setting-up of more complex activities.

One issue highlighted is that practitioners 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.

C. Requirements for a Moodle dedicated 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 interested 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 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.

III. Abstracting the LMS metamodel

According to practitioners' needs, the abstraction could consist in raising the recurrent LMS uses supporting some learners and/or teachers activities. From an activity theory perspective[4][5], such activities should involve LMS's bindings of subject, objects, tools/artefacts, community, division of labor and rules. Because our survey and interviews highlight a special need to ease the parameterization of Moodle tools and resources for setting-up activities, we decided to focus at first on raising these tools and resource semantics, and to study later the other aspects.

A. Tool-or-resource-based pedagogical activities

We define an LMS-abstract pedagogical activity as an *encapsulation of parameters a teacher has to set-up when using a tool (or resource) for a specific pedagogical use*. From a single tool, for example a forum, a teacher 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 mapping (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. It will also impact some other elements in relation with the tool/resource: grades, objectives, groupings, restriction access and achievements rules, etc.

B. Activity structures

In order to ease and assist the practitioners when assembling and setting-up combinations of activities or resources we propose then usual structural elements (selection, sequence, conditional activities, etc.). These blocks will be composed of activities or other activity structures. 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 name, kind and use for users), shifted content (*move left/right* 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 will be parameterized (*restrict access*, *visibility*, *achievement*...) with appropriate properties in order to set up the desired behavior.

C. An identification method for pedagogical activities

In order to identify the most appropriate tool for a specific pedagogical activity, we followed these three steps: (1) analysis for each Moodle tools of its recurrent uses (bottom-up method), (2) identification of tools offering common uses (top-down method) and (3) specification of discriminating criteria to drive the selection of a suitable tool. Moodle 2.4 offers 7 resources (*Book*, *Page*, *Label*, *IMS content package*, *File*, *Folder*, and *URL*) and 13 activities (*Forum*, *Database*, *Glossary*, *Assignment*, *Lesson*, *Quiz*, *Workshop*, *SCORM package*, *External tool*, *Choice*, *Survey*, *Wiki*, and *Feedback*). We studied recurrent uses of these Moodle's activities and resources and noticed that some activities/resources can be diverted to support different uses. For example, everyone knows that the *Forum* can be used for discussions but it can also be used to allow students to introduce themselves in a course or to consult a Frequently Asked Questions (FAQ) or to share documents between learners. After looking at all Moodle's activities/resources uses, we identified those supporting the same ones. Three tools can be used to consult a FAQ: the Forum, the Wiki, and the Glossary.

We then specified discriminating criteria to help teachers decide which tool they must use if they have many choices. We chose the $m \times n$ matrix A format to present these discriminating criteria (A has m rows and n columns) according to seven rules :

- R1 The pedagogical activity name is only from a teacher perspective if no students are concerned (= with parameter *hide* on). For example, for a survey, we choose the expression answer a survey (students viewpoint) instead of create a survey (teachers viewpoint). Note that A_{11} presents this pedagogical activity.
- R2 Tools participating to the realization of the activity are the elements $A_{12}...A_{1n}$.
- R3 Discriminating criteria are the elements $A_{21}...A_{m1}$.
- R4 Discriminating criteria 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 matrix is terminated if there is no similar combination of answers for two tools.

An unachieved matrix indicates to experts that they have to add one more discriminating criteria and verify again the rule R7. Table 1 shows an example of identification matrix for

TABLE I. EXAMPLE OF IDENTIFICATION MATRIX.

PA	T1	T2	T3	T4
C1	Yes/No	No	Yes/No	Yes
C2	Yes/No	No	Yes/No	Yes
C3	No	No	No	Yes
C4	Yes/No	No	No	No
C5	No	No	Yes/No	No
C6	Yes	No	No	No
C7	Yes	No	Yes	No

the pedagogical activity (PA) "Answer a poll". Four Tools can support this activity: *Quiz* (T1), *Choice* (T2), *Feedback* (T3), and *Survey* (T4). Experts have found 7 discriminating criteria. Each criterion is presented in the form of a question:

- (C1) More than one question?
- (C2) Only multiple choice questions?
- (C3) Pre-populated with questions?
- (C4) Time limit?
- (C5) Anonymous?
- (C6) Graded?
- (C7) Feedback after submission?

In table 1, we have three different answers that can imply these four tools: Yes, No, and Yes/No. For example with a survey (T4), we can have more than one question (A_{25} = Yes),only multiple choice questions are allowed (A_{35} = Yes), it is a pre-populated survey with questions (A_{45} = Yes), it can not have a time limit/countdown timer for students' navigation (A_{55} = No), it is always nominative (A_{65} = No), it can not be graded (A_{75} = No), and students can not have a feedback after their submissions (A_{85} = No). Note that a designer can reply to C1, C2, C3, C4 , C5, C6, and C7 in any order. Some combinations cannot lead to a specific tool choice for two reasons: (1) a non-valid combination, or (2) a non-response to all questions. In the first case, the experts will be notified to adapt their pedagogical choices while in the second case they will be asked to precise more choices.

Such identification matrix has to be completed by additional information in order to precise the general (whatever the answers that guide the binding like a Tool's name) or contextualized (depending on some specific answers like a tool's format) parameters for the related LMS activity or resource.

D. A weaving language to formalize the mappings

According to our Model Driven Engineering research framework, we can use model transformations to achieve the mappings specified by experts. The transformations will be run 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.

We on purpose propose to use the model weaving technique we studied in [6] to capture the mapping semantics in dedicated weaving models and automatically generate models transformations. From a practical viewpoint, using the matrix and additional information from an LMS expert using our method and formalisms depicted in section III-C, an engineer will formalize 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 1 is a screenshot of the weaving editor. It is used to formally specify as a weaving model the corresponding binding specified by LMSs' experts.

An example of weaving model can be sketched in the middle part of figure 1. It is related to the binding example from the table I. Such weaving model is realized by following the matrix formalism, tool by tool. Data about the tools parameterizations are deduced from the additional information given by experts.

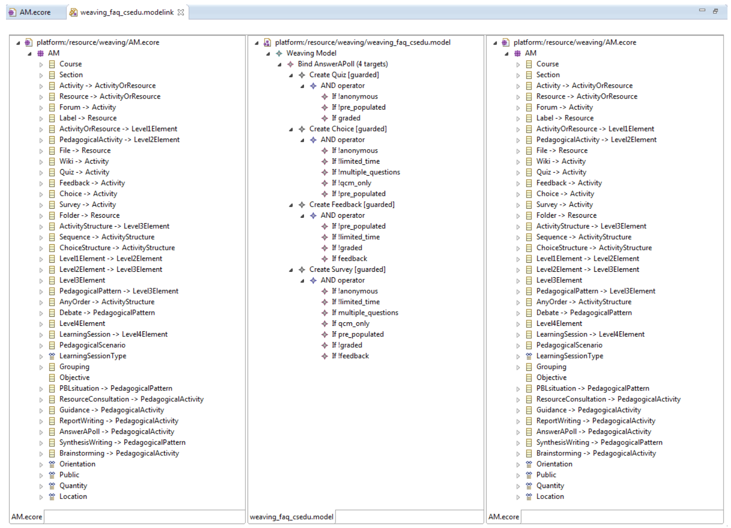

Fig. 1. Screen caption of our weaving tool for formalizing the bindings.

We used languages and tools from the Epsilon project to build a software framework fulfilling our model weaving requirements. This project is compliant with the Ecore formalization of metamodels we already used to formalize the metamodel of Moodle and the extension including pedagogical activities. This Ecore format is from the Eclipse Modeling Framework [7].

IV. A LEARNING SCENARIO EXAMPLE

We on purpose propose to illustrate our proposal by formalizing a very simple but representative learning scenario for the Moodle LMS.

A. *Scenario description and formalization*

The learning scenario is composed of two learning sessions. The first one is a *lecture* session for which the teacher only want to provide 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 set to ”local”. These properties will lead the dynamic mapping process to add the *File* Moodle element to the scenario.

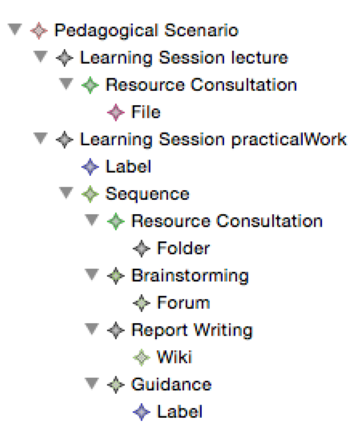

Fig. 2. Example of simplified learning scenario with generated contextualized tools and resources bindings (leaf elements).

The second learning session is a *practical work* that the teacher wants to realize in face-to-face within a computerized classroom. He would like to use the Moodle platform for supporting 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 only be displayed 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, re-arrange their order, or add some other elements. Figure 2 shows a global overview of the learning scenario elements (with no consideration) including all the automatic mappings according to the various properties and values (not depicted within the figure).

B. *Current tooling prototype*

We are currently working on the development of a prototype adding a notation layer on top of the abstract syntax we propose for the Moodle-centered instructional design language. We choose for now the Sirius tooling [8] because it allows to quickly define custom multiview for workbenches with less technical knowledge compared to the well-known GMF tooling (from [7]). The notation, or concrete syntax for our instructional design language, is derived from the abstract syntax formalized as an Ecore metamodel. Sirius also facilitates the development of dedicated graphical tools by generating most of features (diagrams, trees, tables, etc.) from the sirius-specific model we build when using it. It reduces

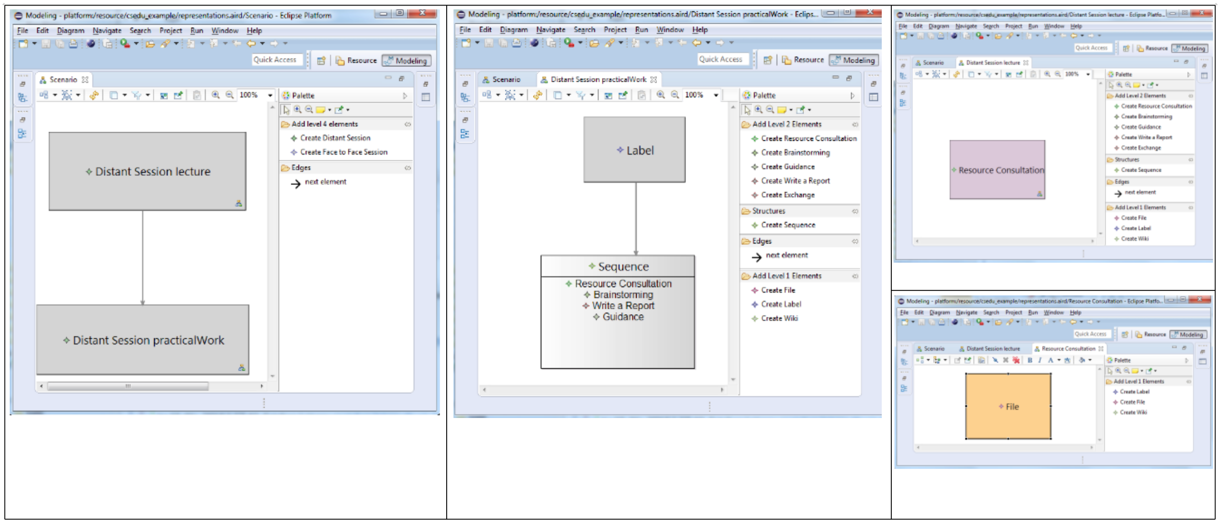

Fig. 3. Different screen captions of our current prototype.

the cost and complexity of developing graphical editors. We succedeed in integrating the mappings transformations within this prototype. For now, when a user double-clicks on a session within the first level diagram (left part of Figure 3), it opens a new diagram where he can mix elements from Moodle with pedagogicial or structural activities according to his Moodle expertise (middle part). Pedagogical properties of activities can be set at this stage. When pedagogical activities are double-clicked, a transformation process is launched for checking all transformation rules automatically generated from the weaving models we produced. The execution of an eligible rule modifies at run-time the current scenario by adding the corresponding binding towards Moodle elements. Right part of Figure 3 shows resulting mappings. The result is part of the pedagogical scenario: it can be modified by adding/editing/reoving new elements. Mappings can also be updated if a user changes the pedagogical properties of a pedagogical activity.

V. Conclusions

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 capturing some recurrent resources or tools uses into pedagogical activities. We also presented a specific method for helping and guiding LMS experts to describe how these activities should be binded to appropriate tools or resources. In addition we propose a specific model weaving approach for formalizing these mappings. The resulting weaving model will drive at run-time and in real-time the automatic translations when using the authoring-tool. Thanks to illustrative examples and an overview of our current prototyping editor, we concretely argued and verified our propositions.

The current abstract syntax proposition is still being improved in order to allow the declaration of didactic objectives to the various elements. According to the objective status (goals/competencies for teacher or activities objectives for learners understanding), the mapping could lead to Moodle *Outcomes*, attached to the root *Course* and referenced by the direct or indirect corresponding *Level 1 elements*, describing labels, or simply a description field filling-up. Similarly, roles or groups are in progress to be included in order to allow the division of labour in the learning scenario. Mappings to the Moodle concepts of *Group* and *Grouping* are realized. Also, in our approach the extended Moodle metamodel will not allow to serialize future learning scenarios in conformance with the Moodle 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.

References

[1] D. Garrison and H. Kanuka, "Blended learning: Uncovering its transformative potential in higher education," *The Internet and Higher Education*, vol. 7, no. 2, pp. 95 – 105, 2004.

[2] C. Alario-Hoyos, J. Munoz-Cristobal, L. Prieto-Santos, M. Bote-Lorenzo, J. Asensio-Perez, E. Gomez-Sanchez, G. Vega-Gorgojo, and Y. Dimitriadis, "GLUE! - GLUE!-PS: An approach to deploy non-trivial collaborative learning situations that require the integration of external tools in VLEs." in *Proc. 1st Moodle Research Conference*, Heraklion, Greece, 2012, pp. 77–85.

[3] M. Katsamani and S. Retalis, "Orchestrating learning activities using the cadmos learning design tool," *Research in Learning Technology*, vol. 21, no. 0, 2013.

[4] Y. Engeström, *Learning by Expanding: An Activity-theoretical Approach to Developmental Research*. Orienta-Konsultit Oy, 1987.

[5] A. Benson, C. Lawler, and A. Whitworth, "Rules, roles and tools: Activity theory and the comparative study of e-learning," *British Journal of Educational Technology*, vol. 39, no. 3, pp. 456–467, 2008.

[6] E. Loiseau, P. Laforcade, and S. Iksal, "Model weaving and pedagogy - mapping abstraction levels in instructional design languages," in *ICSOFT-PT 2014 - Proceedings of the 9th International Conference on Software Paradigm Trends, Vienna, Austria, 29-31 August, 2014*, 2014, pp. 81–86. [Online]. Available: http://dx.doi.org/10.5220/0005003600810086

[7] (2014) Eclipse modeling project official website. [Online]. Available: http://www.eclipse.org/modeling/

[8] (2014) Sirius project. [Online]. Available: http://eclipse.org/sirius/