A method for identifying and formalizing the underlying instructional design language of existent LMSs

Nour El Mawas, Lahcen Oubahssi and Pierre Laforcade
Université du Maine
LIUM (Laboratoire d'Informatique de l'Université du Maine) Lab
TEL (Technology-Enhanced Learning)Team
Presentation outline

• Context

• Motivation

• The GraphiT project

• Research questions

• Our approach / Moodle case study

• Conclusion & Perspectives
Context

• Learning Management Systems like MOODLE
  – Widespread within academics organizations
  – Not limited to distant courses
  – Provide many tools and services to teachers-designers

But

• They require for teachers
  – to understand the « way of thinking » of LMSs
  – to abstract from technical/low-levels details and form-based screens

• They directly use LMSs depending on their own expertise level

• Instructional design languages and tools are misknown or not at their disposal
Motivation

• Despite
  – many existing standards (Martinez-Ortiz et al., 2009) (Mekpiroona et al., 2008),
  – approaches (De Vries et al., 2006), languages (Baggetun et al., 2004),
  – architectures (Alario-Hoyos et al., 2013), and
  – tools (Baggetun et al., 2004) (Al-Ajan and Zedan, 2007) to facilitate the instructional design,
• they are often not compatible with existent LMSs,
• They require a costly reengineering of the LMS (new web service API, new runtime engines, etc.).
• They do not simplify the operationalization of the produced models. Some translations, leading to information or semantics losses, are still required to operationalize them into a targeted LMS.
The GraphiT project

• **General informations**
  – Funded by the french national research agency (ANR)
  – Start/End: February 2012 / September 2015
  – Website: [http://www-lium.univ-lemans.fr/~laforcad/graphit/](http://www-lium.univ-lemans.fr/~laforcad/graphit/)
  – Involved several research members from our LIUM laboratory

• **Objective**
  – To study how possible it is to specify and develop operationalizable VIDLs/editors starting from the LMS instructional design language and taking into account some practitioners' needs and practices DSM and Model Driven Engineering techniques for VIDLs
  – Metamodel composition, model weaving, specification of concrete syntaxes, models transformations Patterns-oriented approaches for making explicit needs and practices
  – Pedagogical patterns, analysis & design patterns...

<table>
<thead>
<tr>
<th>Context</th>
<th>Motivation</th>
<th>The GraphiT project</th>
<th>Research questions</th>
<th>Our approach/Moodle case study</th>
<th>Conclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>24/05/2015</td>
<td>The GraphiT project</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Research questions

• How helping teachers-designers (having an imposed LMS)
  – in improving their capacity of abstraction and understandability of the LMS?
  – in facilitating the design of learning scenarios in conformance to the LMS abilities?
  – In exploiting these scenarios for pre-configuring the courses?
Overview of our approach

• **Hypothesis**
  – LMSs embed an *implicit* instructional design language
  – It is possible to *explicit and exploit* it as a base for *external* design tools

• **Added value of our approach**
  – Specific LMS format (e.g. MOODLE format for quizzes) extended to the whole instructional design aspects
    ▪ without considering the resources => must be included into one XML file
  – Analogy with import/export standards (like SCORM)
    ▪ Without having to add a new runtime engine and semantics
    ▪ But an import/export facility (less complex as backup/restore)
  – Allow the designing and development of external specific languages and tools in conformance with the one identified
    ▪ For regulation activities, tracking, design of learning situations...
    ▪ For interoperability purposes between two LMSs, etc.
Our approach

(a) Define a process to extract the ID languages
(b) Develop an external editor
(c) Upload scenarios
(d) Develop an API

Context | Motivation | The GraphiT project | Research questions | Our approach/Moodle case study | Conclusion
---|---|---|---|---|---
24/05/2015 |  | | | |
The identification and the formalization process

• We define the necessary analysis and steps for the identification and formalization of an LMS instructional design language.

• It is specified according to three different viewpoints:
  - a viewpoint centred on macro-HMI
  - a functional viewpoint
  - a micro viewpoint.

• Formalism: the meta-model format
The identification and the formalization process: An overview
The macro IHM analysis

Objective: identify platform interfaces related to the Instructional Design (ID).
Moodle macro-HMI analysis

Context
Motivation
The GraphiT project
Research questions
Our approach/Moodle case study
Conclusion

24/05/2015
An extract of Moodle macro-HMI model
The factorization analysis

**Objective**: find common elements in pedagogical activities/resources and common relations between them.
Moodle factorization analysis

- Factorization is the process of finding common attributes shared between two or more pedagogical elements (classes) in the macro-HMI model and moving them into an existent or a new abstract parent element.

- This step aims to find common elements in pedagogical activities/resources and common relations between them. Factorization is applied on the Macro-HMI model. The macro model, resulting for the factorization, is clearer and more simplified than the Macro-HMI model.
An extract of Moodle Moodle Macro model
Objective: identify the functionalities dedicated to the course instructional design.
Moodle functional analysis

Context

Motivation

The GraphiT project

Research questions

Our approach/Moodle case study

Conclusion
An extract of Moodle functional analysis

Context
Motivation
The GraphiT project
Research questions
Our approach/Moodle case study
Conclusion

24/05/2015
Objective: takes into account two different viewpoints: micro-HMI and technical viewpoints.
The micro analysis
The micro HMI analysis (micro analysis)

**Objective**: identify all elements relevant to the instructional design, including their features (attributes, types, etc.).
Moodle micro HMI analysis

Context

Motivation

The GraphiT project

Research questions

Our approach/Moodle case study

Conclusion
An extract of Moodle micro HMI model
The technical analysis (micro analysis)

**Objective**: specify a reduced Conceptual Data Model from the one available by LMS providers if it exists.
Moodle technical analysis

• This technical analysis consists in
  - (1) looking over all database tables in order to sketch a first draft of the model,
  - (2) focusing on tables embedding elements in relation to instructional design concepts.
An extract of Moodle technical model

Context

Motivation

The GraphiT project

Research questions

Our approach/Moodle case study

Conclusion
The Confrontation & formalization (micro analysis)

**Objective**: concerns the confrontation of both micro-HMI and technical models, and the formalization of the final model.
The Confrontation and formalization (micro analysis)

• The micro-HMI and technical models are compared in order to
  -(1) refine the micro-HMI model
  -(2) detect and correct the difference between models
  -(3) ensure that the final model can be easily bind to a computer-readable format for the existent LMS.

• Some differences or ambiguities are so identified. They require a deeper and finer analysis of both HMI and technical analysis. At this step, other technical-centred analysis (source code, backup packages, etc.) can be useful.
The Confrontation and formalization

Context

Motivation

The GraphiT project

Research questions

Our approach/Moodle case study

Conclusion
An extract of Moodle final model

<table>
<thead>
<tr>
<th>Context</th>
<th>Motivation</th>
<th>The GraphiT project</th>
<th>Research questions</th>
<th>Our approach/Moodle case study</th>
<th>Conclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>24/05/2015</td>
<td>The GraphiT project</td>
<td>Research questions</td>
<td>Our approach/Moodle case study</td>
<td>Conclusion</td>
<td>31</td>
</tr>
</tbody>
</table>
Conclusion & Perspectives

- a meta-model-based approach and method for identifying and formalizing LMS languages.
- We apply our proposed method on the Moodle platform.
- The meta-model will be used as a basis for the development of the external editor by using a Model Driven Engineering tooling like EMF-GMF.
- This will facilitate the use of LMS and allow to teachers and pedagogical engineers (service information technology and communication for education) of becoming more familiar with the specific design upon this LMS.
A method for identifying and formalizing the underlying instructional design language of existent LMSs

Thank you!

Contact emails:
nour.el_mawas@univ-lemans.fr
lahcen.oubahssi@univ-lemans.fr
pierre.laforcade@univ-lemans.fr