Improving the design of courses thanks to graphical and external dedicated languages: a Moodle experimentation

Authors: Pierre Laforcade, Aymen Abedmouleh

Le Maine University
LIUM laboratory
(Computer Science Laboratory)

Laval, France
What is it about?

- Stated purposes
  - Focusing on presenting and positioning our LMS-centered approach for allowing the specification and development of instructional design languages and tools
  - Introducing the GraphiT project

- Moodle as the concrete LMS for our experimentation
  - Our LMS University (UMTice) is based on it
  - Wide community of teachers-designers and pedagogical engineers, including within our university
Outline

• Context and general problematics
• Background about instructional design for LMSs
  ‣ Internal solutions
  ‣ External solutions
• Problematics and objectives
• Our approach
  ‣ Global overview
  ‣ About DSM
  ‣ First results about Moodle
  ‣ The GraphiT Project
• Summary
Context & general problematics

• 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

• 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?
Background about instructional design for LMSs: internal solutions

- By default: the form-based interfaces

- The LAMS for Moodle experience
  - Advantages: graphical representation
  - Inconveniences: requires a specific course & runtime engine

- Backup/restore functions
  - Not always accessible for teachers-designers
  - Database-centered semantics with several XML files
  - Embed more than instructional design informations
    - All resources (files, questions/answers for quizzes, etc.)
    - Enrollment and run-time data

- Development of internal editors
  - Constrained by the LMS architectural and programming languages
  - Often PHP client-server
• **Educational Modeling Languages (EML)**
  - Allow the specification of learning activities and units of learning
  - Formalization & binding => machine-readable scenarios
  - Focus on « abilities » (reusability, interoperability...)
  - Some standards (SCORM, IMS-LD)
  - Difficulties in providing a concrete operationalization support for LMSs

• **Visual Instructional Design Languages (VIDL) or Visual tools**
  - Favor imagination, creative thinking, communication
  - Semantics closer to practitioners
  - Offer visual notations => human-interpretable scenarios
  - Fail in providing a binding for exploiting the scenarios
    - Sometimes exports are proposed towards standardized EMLs (IMS-LD)
    - No direct binding and/or setting-up for LMSs

• Authoring-tools for EMLs/VIDLs often reduced to a programming effort...
Background: the crucial need of an operationalization support

- Operationalizing into LMSs
  - SCORM 1.3 success (dedicated runtime engine) but SCORM 2004?
  - IMS-LD-compliance attempts
    - Require a specific engine runtime
  - Models transformations towards LMS-specific scenarios
    - Partial and subjective LMS metamodels and semantics
    - Informations losses
    - Still require a manual setting-up or a partial automatic solution through a web-services setting-up
  - Web services oriented frameworks
    - Require hard modification of LMSs architecture

- Usual proposals (tools, languages) are first LMSs independant and then tackle the need of LMSs operationalization
  - By exportation services (with information losses), by extending LMSs with new semantics and/or with big new runtime engine and/or hard-modifications...
Our approach: LMS-dependant tools

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

• Original idea
  ▸ 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 research objectives and context

• Our specific objectives
  ▸ Specifying VIDLs and developing dedicated authoring-tools
  ▸ Centred on LMSs' semantics
  ▸ Directed towards practitioners needs and practices
  ▸ Within a Domain Specific Modeling methodological and Model Driven Engineering practical framework
    • Our research team background and skill

• Focus on exploring how DSM can be useful and relevant for helping specifying and developing such VIDLs and prototypes of editors
  ▸ Focus on the VIDLs semantics in relation to the LMS one

• NOT focus on user-friendliness/ergonomics/... of authoring-tools
Our approach: overview

1. Identification and formalization of the implicit learning design language

2. Addition of a specific API to the LMS for realizing the import/export facilities

3. Specification and tooling of external design languages and tools with a DSM approach
   3a. VIDL semantics close to the LMS one
   3b. VIDL semantics more abstract than the LMS one
DSM techniques for the specification of VIDLs and the development of dedicated editors

- **Domain Specific Modeling (DSM)**
  - A software engineering methodology for designing and developing systems
  - Involves the systematic use of a graphical DSM Language

- DSM tools
  - Propose meta-modeling techniques
    - Formalizing domain-specific vocabularies (abstract syntaxes)
    - Facilitating definition of notations (concrete syntaxes)
  - Generate domain-specific software code
  - Generate powerful and user-friendly editors for DSM languages
    - Kind of meta-CASE editors
    - Domain-designers can graphically specify models from their domain
    - Models persistence facilities in a machine-interpreted format

- DSM applications for the instructional design and TEL systems
  - Past results from our research team
Application to Moodle: Identification and formalization of its ID language

- A 3-viewpoints method
  - Users interfaces analysis (what designers see)
  - Functionnal analysis (what Moodle can do)
  - Databases & code analysis (how data/functions are stored/realized)

- Produce the Moodle meta-model and an XSD file
Application to Moodle: development of an import/export module
A very first VIDL and editor

- A prototypal concept-of-proof for using the DSM tooling
- Use of the Eclipse EMF/GMF modeling frameworks
- The notation, mapping, palette, editor code, scenarios persistence, etc. are driven by models starting from the identified Moodle domain one (no hand-coding)
A very first VIDL and editor (2)

• Practitioners' requirements
  ‣ Driven by our constraints and verification purposes
  ‣ Limited to the graphical design of courses, sections and Moodle activities/resources without a definitive ordering
    • Weak added-value in comparison to directly use Moodle
    • But sufficient for our verification

• Results
  ‣ Issues detected when the metamodel semantics has to be extended
    • a new Erelation (and code modification) has been required to separate the ordering of sections from their creation order (the persistent one in the generated XML files)
  ‣ The acceptance of an external tool before importing & completing the course within Moodle is dependent to its added-value
  ‣ Need for a VIDL semantics more abstract from the Moodle semantics
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/
  ▸ 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 expliciting needs and practices
  ▸ Pedagogical patterns, analysis&design patterns...
• Practices and needs from practitioners
  ▸ CoP currently being structured
    • Teachers-designers involved in distance & mixed courses
    • Pedagogical engineers from our University
  ▸ A pattern-oriented editor in development
• Some first prototypal results about a VIDL semantics relying on Moodle's tools usages rather than on tools (pedagogical activities)
Summary

• A Moodle-centered approach for improving the specification and development of external instructional design languages and tools
  ▸ Identification and explicitation of the instructional design language as a specific format to conform with
  ▸ An import/export API
  ▸ A first external exploitation based on this language

• A DSM/MDE methodological research frame

• The GraphiT project
  ▸ Exploring how DSM/MDE can be useful to raise the VIDLs semantics while taking into account the operationalization of produced models
Thank you!

...Any questions?

Contact emails:
pierre.laforcade@univ-lemans.fr
aymen.abedmoulehl@univ-lemans.fr