Next Step for Educational Modeling
Languages: The Model Driven Engineering and Reengineering Approach

Authors: Pierre Laforcade, Christophe Choquet

Contact email: pierre.laforcade@lium.univ-lemans.fr

Speaker: Noa Randriamalaka

REDiM Project
(Model Driven Reengineering of Technology Enhanced Learning)
LIUM lab
(Computer Science Laboratory of Le Maine University)
Outline

1. Research context
2. Personal analysis of current EML researches
3. New proposal
4. What is MDE/MDA?
5. Global overview of MDE/MDA applied to scenarios
6. Characteristics of this new approach
7. Summary and ongoing work

Presentation of a new approach for the design of learning scenarios
Research context

- REDiM Project: general objectives
  - Re-engineering of TEL (Technology Enhanced Learning) driven by the usage analysis
    - Strong interest about the design / analysis / re-engineering of learning scenarios

- 'LD'-community overall agreements (UNFOLD meeting)
  - Designers of education: instructional designers and regular teachers
  - Lack of user-friendly design tools (Users = teachers, training practitioners, ...)
  - Lack of runtime tools

- New initiatives based on closed research domains
  - Use of ontologies and semantic web principles and tools
  - Use of learning design patterns
  - etc.
Personal analysis of current EML researches

• 2 categories for EMLs design tools
  • LD-centered propositions
    • New (user-friendly) design facilities as a layer over an IMS-LD core
      ⇒ Produced models are IMS-LD ones
    • Examples: Collage, etc.
    • Inconvenience: pedagogical expressiveness limited to the LD one

  • LD-compliant propositions
    • IMS-LD independent propositions
    • LD-compliance by « exportation or « save as » services (only level A in practice)
    • Examples: MOT+, CPM

→ abstraction of this second category
Application of theories and results from the Model Driven Engineering (MDE) and Model Driven Architecture (MDA) domains to the learning scenarios design in a mixed engineering/re-engineering process.
What is MDE/MDA?

- **MDE (Model Driven Engineering) idea**
  - A system is developed by refining models starting from higher and moving to lower levels of abstraction until code is generated
  - Refinement is implemented by transformations over models

- **MDA (Model Driven Architecture) approach** (from OMG)
  - 3 classes of models
  - CIM (Computer Independent Model)
    - The domain model or enterprise model
    - Specify what the system is expected to do
  - PIM (Platform Independent Model)
    - The computerized domain model: computer-readable
    - Independent from specific platforms (EJB, Corba, SQL, etc.)
  - PSM (Platform Specific Model)
    - The PIM with the details of how it will be implemented on a specific platform
Global overview of MDE/MDA applied to scenarios

<table>
<thead>
<tr>
<th>Models for what?</th>
<th>Models for who?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain scenario</td>
<td>Abstract scenario</td>
</tr>
<tr>
<td></td>
<td>CIM *</td>
</tr>
<tr>
<td><strong>Main objectives</strong></td>
<td>- Guide for the design and the reuse;</td>
</tr>
<tr>
<td></td>
<td>- Ease the exchanges of learning scenario within a same learning design “community of practice”</td>
</tr>
<tr>
<td></td>
<td>- Promote the exchange and interoperability of LMS-independent scenarios</td>
</tr>
<tr>
<td><strong>Learning theory-dependent</strong></td>
<td>Yes/no</td>
</tr>
<tr>
<td></td>
<td>(e.g. IMS-LD: independent because of the pedagogical flexibility objective)</td>
</tr>
<tr>
<td><strong>LMS-dependent</strong></td>
<td>Yes/no</td>
</tr>
<tr>
<td><strong>Public for handling models</strong></td>
<td>Human</td>
</tr>
<tr>
<td><strong>Examples of dedicated languages</strong></td>
<td>CPM, MOT+</td>
</tr>
</tbody>
</table>
Characteristics of this new approach

- Not IMS-LD-centered
  - IMS-LD is just an example of language for abstract scenarios
- Model-centered approach
  - Separation of concerns according to the kind of learning scenario (domain/abstract/deployed)
- Important role supported by the scenarios transformations

Personal concerns about CIM ↔ PIM transformations
Already some existent research works and results
Summary - Application of MDE-MDA theories and techniques for the design of learning scenarios

• Interests
  • New trend for EMLs
    • Can improve the providing of dedicated languages and user-friendly design tools for regular teachers/practitioners
    • Support and ease the re-engineering of learning scenarios
  • Main characteristics
    • Model-centered approach → separation of concerns
    • Important role for scenarios transformations

• Ongoing works
  • Pragmatic/Bottom-up process
    • Testing tools and techniques from MDE/MDA research community...
    • ... with reference learning scenario languages and models
  • Experiments with teachers/practitioners
  • Prototypes development
Thank you!

...Any questions?

Email-me! → pierre.laforcade@lium.univ-lemans.fr
Example: CPM to IMS-LD

- Transformation details
  - Provided as a service of the CPM design tool
  - CPM activity diagram to LD-XML model
  - Level A hard-coded facility
  - Marking-based imperative transformation

From... to...

\[
\begin{aligned}
\text{CPM activity diagram} & \rightarrow \text{LD-XML model} \\
\text{Level A hard-coded facility} & \rightarrow \text{Marking-based imperative transformation}
\end{aligned}
\]
Example: IMS-LD to UML4

- Transformation details
  - UML4LD: UML profile dedicated to IMS-LD
  - Level A hard-coded facility
  - Imperative transformation
Why separating domain scenarios (CIM) from abstract ones (PIM)?

• The CIM focuses on the design → it addresses first Human
  ▪ Design = description semi-formal
  ▪ Learning scenario design at a “knowledge” level → concepts/relations are those of the teachers-designers (specific to their pedagogical approach, to the platform they usually use, etc.)
  ▪ Towards specific teachers/practitioners-adapted languages and user-friendly dedicated tools (community of practice)

• The PIM focuses on the “...abilities” (interoperability (*on different LMS*), formalization, exchange, reuse, etc.) → it addresses first Machine
  ▪ Design = formal specification (*automatic interpretation implies no ambiguities*)
  ▪ Learning scenario design at an “abstract” level → concepts/relations have to be platform-independent (and are those of a targeted community of teachers-designers)
From domain model to abstract one and *vice versa*

- Domain to abstract (CIM → PIM)
  - Need of abstraction level (formal LMS-independent level) for *reusing*, *exchanging*, etc.
    - Need of **playing** the scenario by means of PIM-compliant runtime tools
    - Example: CPM to IMS-LD

- Abstract to domain (PIM → CIM)
  - Need of representation at the « knowledge level » (or domain language) of:
    - An abstract scenario → in order to **ease the reuse by improving the understanding** of the formal scenario
    - A descriptive scenario or some “chunk” of concrete activities (obtained after the analysis of end-users tracks) → in order to **ease the re-engineering of the prescriptive scenario** in an iterative design process
    - Example: IMS-LD to UML4LD