Model-Driven Engineering (MDE) and Model-Driven Architecture (MDA) approach for learning scenarios

Pierre Laforcade

email: pierre.laforcade@lium.univ-lemans.fr
website: http://www-lium.univ-lemans.fr/~laforcad/

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

Overview

1. **MDE/MDA approach applied to learning scenarios**

2. Focus on the (re)design of learning scenarios from a MDE/MDA point of view

3. Summary and ongoing work

Recent trends for the design and runtime of learning scenario

- '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
  - Use of ontologies and semantic web principles and tools
  - Use of learning design patterns
    - Example Collage focusing on the building of collaborative designs
  - Development of Learning Design *Authoring* and *Content Management* Systems
  - Development of Learning Design *Players*

- Moving question
  - From “what is it?” to “who is it for and how can it be put to good use?”

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**
    - Inconveniences:
      - compliance in MOT+ is realized by the adding of LD concepts (marking-based) => MOT+ notation but pedagogical expressiveness of IMS-LD
      - compliance in CPM is limited to CPM activity diagrams and CPM concepts close to the IMS-LD ones

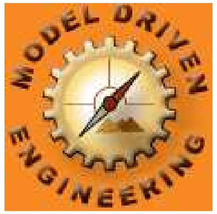
→ generalization of this second category

New proposal

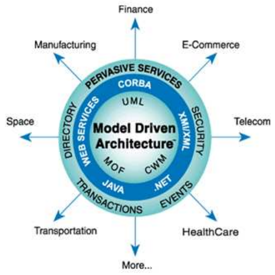
Application of theories and results from the Model Driven Engineering (MDE) and Model Driven Architecture (MDA) domains

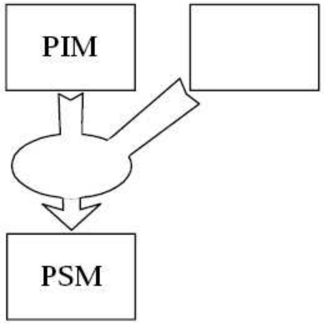
What is MDE/MDA?

- 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
    - Use to be out of scope of model transformations because not computer-readable but new trend against this (***Domain Specific Languages*-DSL**)
  - 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

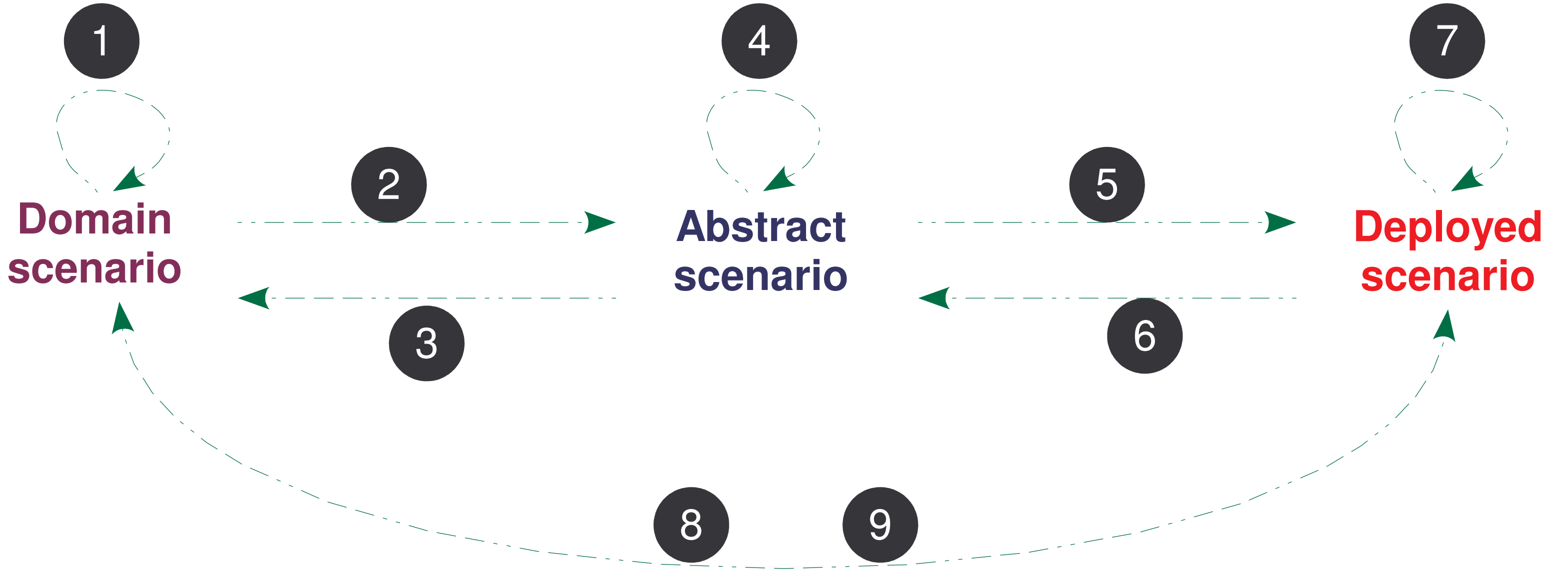
		Domain scenario	Abstract scenario	Deployed scenario
		CIM*	***PIM****	***PSM****
Models for what?	**Main objectives**	- Guide for the design and the reuse; - Ease the exchanges of learning scenario within a same learning design "community of practice"	- Describe the learning scenario in a formal way. - Describe the scenario in a LMS-independent form - Promote the exchange and inter-operability of LMS-independent scenarios	- Guide the configuration of the specific LMS or runtime-environment
	Learning theory-dependent	Yes/no	Yes/no (e.g. IMS-LD: independent because of the pedagogical flexibility objective)	Depend on the considered platform
Models for who?	**LMS-dependent**	Yes/no	no	yes
	Public for handling models	Human	Machine	Machine
	Examples of dedicated languages	CPM, MOT+	IMS-LD, LDL	The LMS' metamodels (some are given by LMS vendors, others need abstraction and modeling efforts in order to be specified)

Position and characteristics of this new approach

- LD point of view (from the [Griffiths et al.,2005] two dimensions axes
  - the approach is ecific or **general purp**

Roadmap for the possible learning scenario transformations in a mixed MDE-MDR context

MDR: just-another-3-letters-acronym for **Model-Driven Re-engineering**



- Transformations between 2 different languages
  - Engineering way: 2, 5, 8
  - Re-engineering way: 3, 6 ,9
- Transformations with the same language: 1, 4, 7

Existent works* about learning scenario transformations

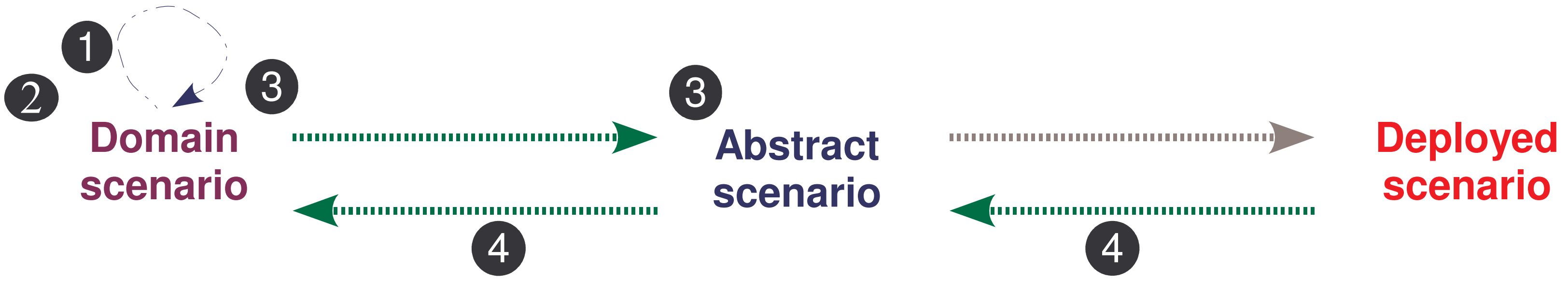
* from different French labs

Overview

1. *MDE approach applied to learning scenarios*
    1. *Recent trends for the design and runtime of learning scenario*
    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. *Position and characteristics of this new approach*
    7. *Roadmap for the possible learning scenario transformations*
2. **Focus on the (re)design of learning scenarios from a MDE point of view**
    1. **Context: the REDiM project**
    2. Why separating domain scenarios (CIM) from abstract ones (PIM)?
    3. From domain model to abstract one and *vice versa*
    4. Example: CPM to IMS-LD
    5. Example: IMS-LD to UML4LD
    6. Highlighting abstract and concrete syntaxes aspects of domain and abstract languages
3. Summary and ongoing work

Context: the 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
- Some of the specific objectives
  1. Considering “collective design” between various teachers/practitioners
  2. Providing them user-friendly languages and tools according to their specific preferences (pedagogical approach, etc.)
  3. Providing them facilities for the definition of “observation needs” and “observation means”
  4. Providing them facilities for the representation at “knowledge level” of abstract scenario and results from the analysis of users' tracks



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) (similarity with DSL)

- 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)

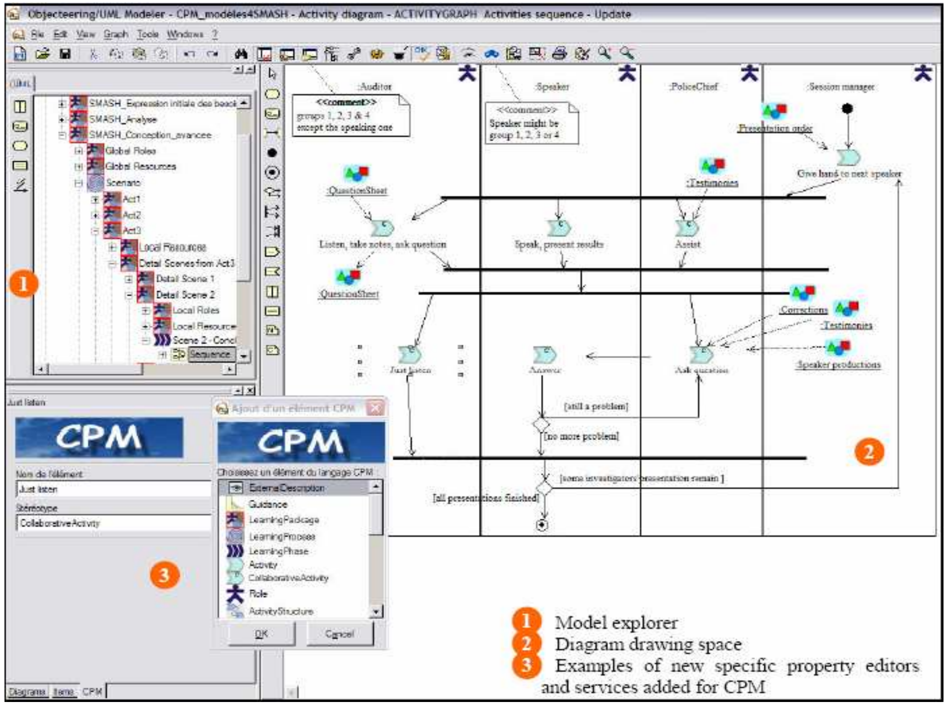
From domain model to abstract one and *vice versa*

- 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

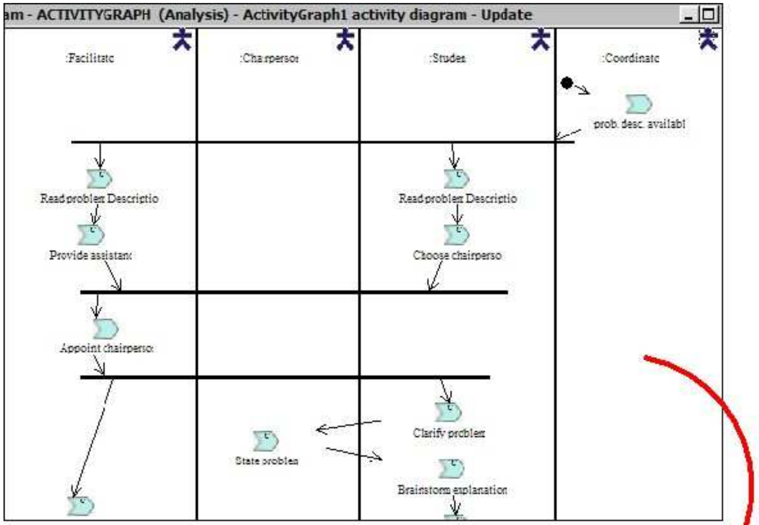
Example: CPM to IMS-LD

- 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

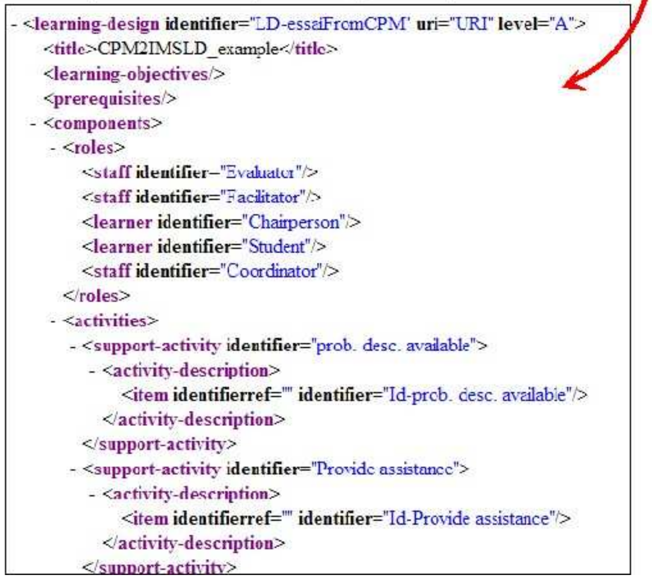


1 Model explorer
2 Diagram drawing space
3 Examples of new specific property editors and services added for CPM

From...



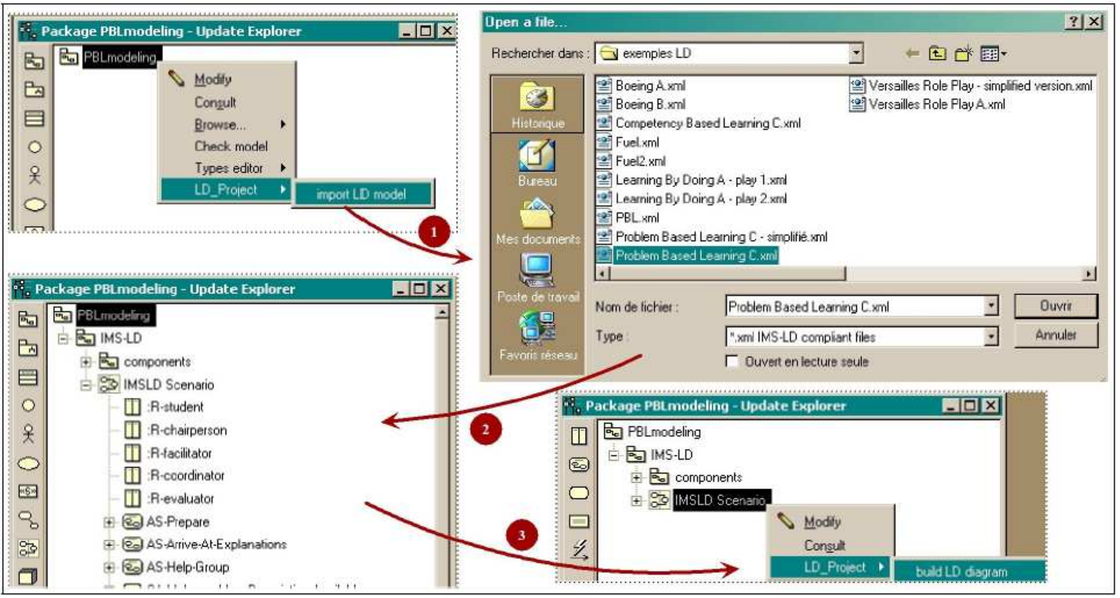
...to



```
- <learning-design identifier="LD-essaiFromCPM' uri="URI" level="A">
    <title>CPM2IMSLD_example</title>
    <learning-objectives/>
    <prerequisites/>
  - <components>
    - <roles>
        <staff identifier="Evaluator"/>
        <staff identifier="Facilitator"/>
        <learner identifier="Chairperson"/>
        <learner identifier="Student"/>
        <staff identifier="Coordinator"/>
      </roles>
    - <activities>
      - <support-activity identifier="prob. desc. available">
        - <activity-description>
            <item identifierref="" identifier="Id-prob. desc. available"/>
          </activity-description>
        </support-activity>
      - <support-activity identifier="Provide assistance">
        - <activity-description>
            <item identifierref="" identifier="Id-Provide assistance"/>
          </activity-description>
        </support-activity>
```

Example: IMS-LD to UML4LD

- Transformation details
  - UML4LD: UML profile dedicated to IMS-LD
  - Level A hard-coded facility
  - Imperative transformation

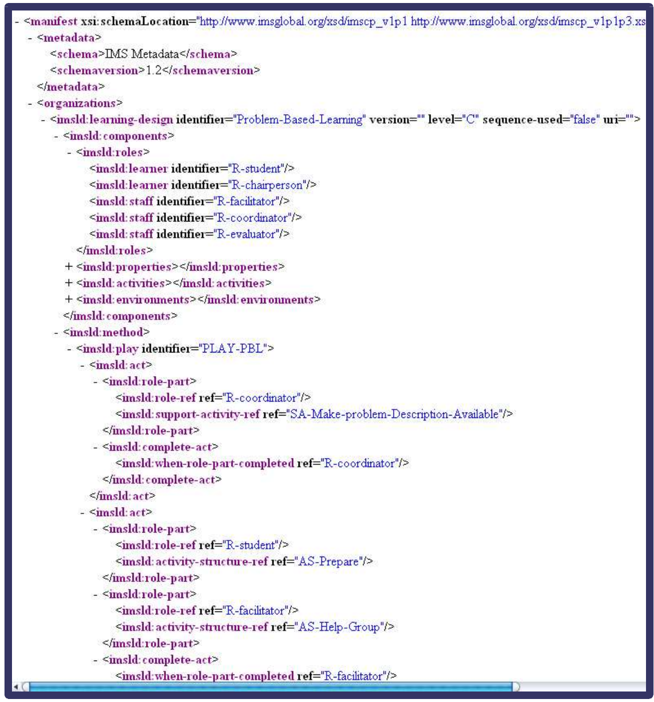


Nom de fichier : Problem Based Learning C.xml

Type : *.xml IMS-LD compliant files

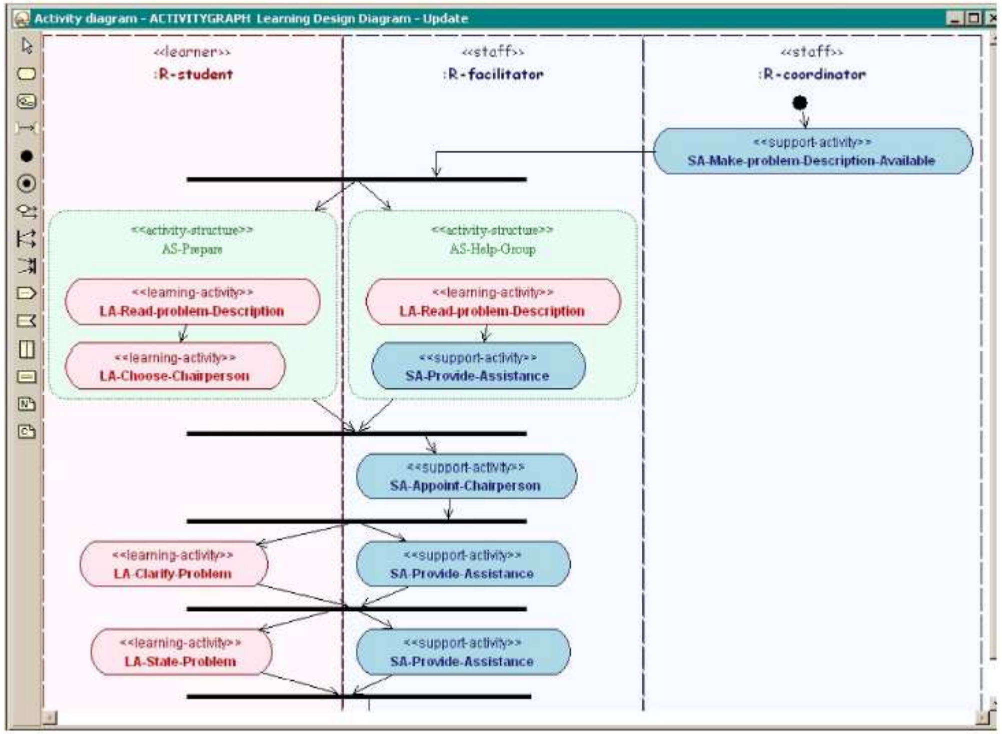
☐ Ouvert en lecture seule

From...



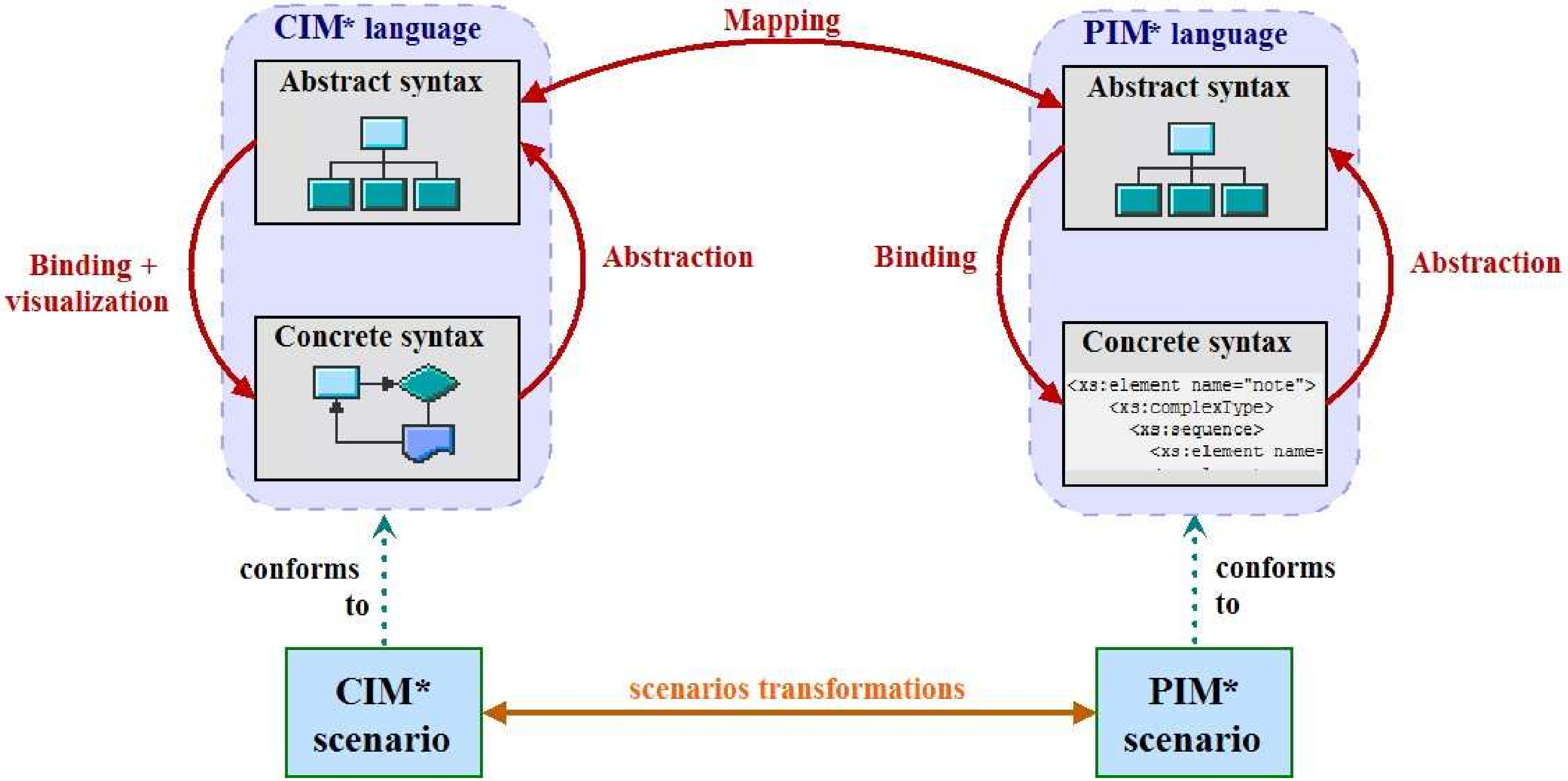
```
- <manifest xsi:schemaLocation="http://www.imsglobal.org/xsd/imscp_v1p1 http://www.imsglobal.org/xsd/imscp_v1p1p3.xs
  - <metadata>
      <schema>IMS Metadata</schema>
      <schemaversion>1.2</schemaversion>
    </metadata>
  - <organizations>
    - <imsld:learning-design identifier="Problem-Based-Learning" version="" level="C" sequence-used="false" uri="">
      - <imsld:components>
        - <imsld:roles>
            <imsld:learner identifier="R-student"/>
            <imsld:learner identifier="R-chairperson"/>
            <imsld:staff identifier="R-facilitator"/>
            <imsld:staff identifier="R-coordinator"/>
            <imsld:staff identifier="R-evaluator"/>
          </imsld:roles>
        + <imsld:properties></imsld:properties>
        + <imsld:activities></imsld:activities>
        + <imsld:environments></imsld:environments>
        </imsld:components>
      - <imsld:method>
        - <imsld:play identifier="PLAY-PBL">
          - <imsld:act>
            - <imsld:role-part>
                <imsld:role-ref ref="R-coordinator"/>
                <imsld:support-activity-ref ref="SA-Make-problem-Description-Available"/>
              </imsld:role-part>
            - <imsld:complete-act>
                <imsld:when-role-part-completed ref="R-coordinator"/>
              </imsld:complete-act>
            </imsld:act>
          - <imsld:act>
            - <imsld:role-part>
                <imsld:role-ref ref="R-student"/>
                <imsld:activity-structure-ref ref="AS-Prepare"/>
              </imsld:role-part>
            - <imsld:role-part>
                <imsld:role-ref ref="R-facilitator"/>
                <imsld:activity-structure-ref ref="AS-Help-Group"/>
              </imsld:role-part>
            - <imsld:complete-act>
                <imsld:when-role-part-completed ref="R-facilitator"/>
```

...to



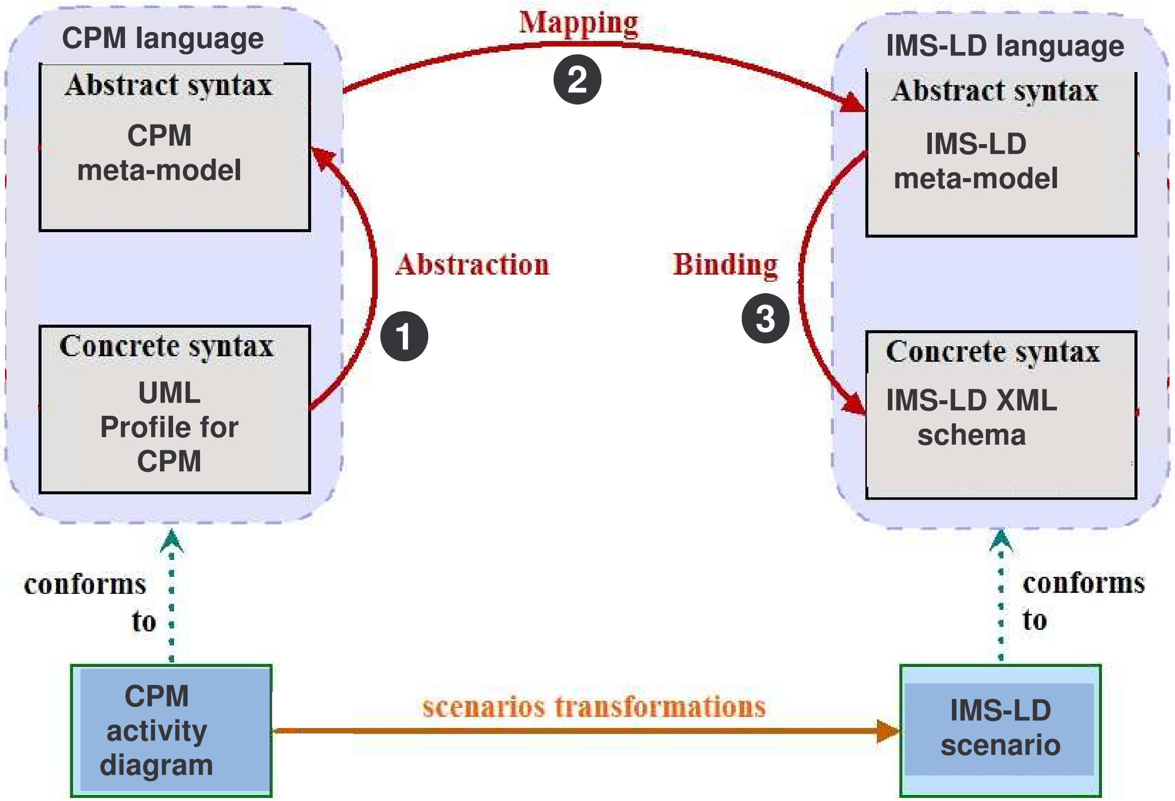
Highlighting abstract and concrete syntaxes aspects of domain and abstract languages

- Many obstacles for scenarios transformations
  - Abstraction from specific notation
  - Meta-models mapping
  - Binding (+ visualization for domain languages) towards concrete notation

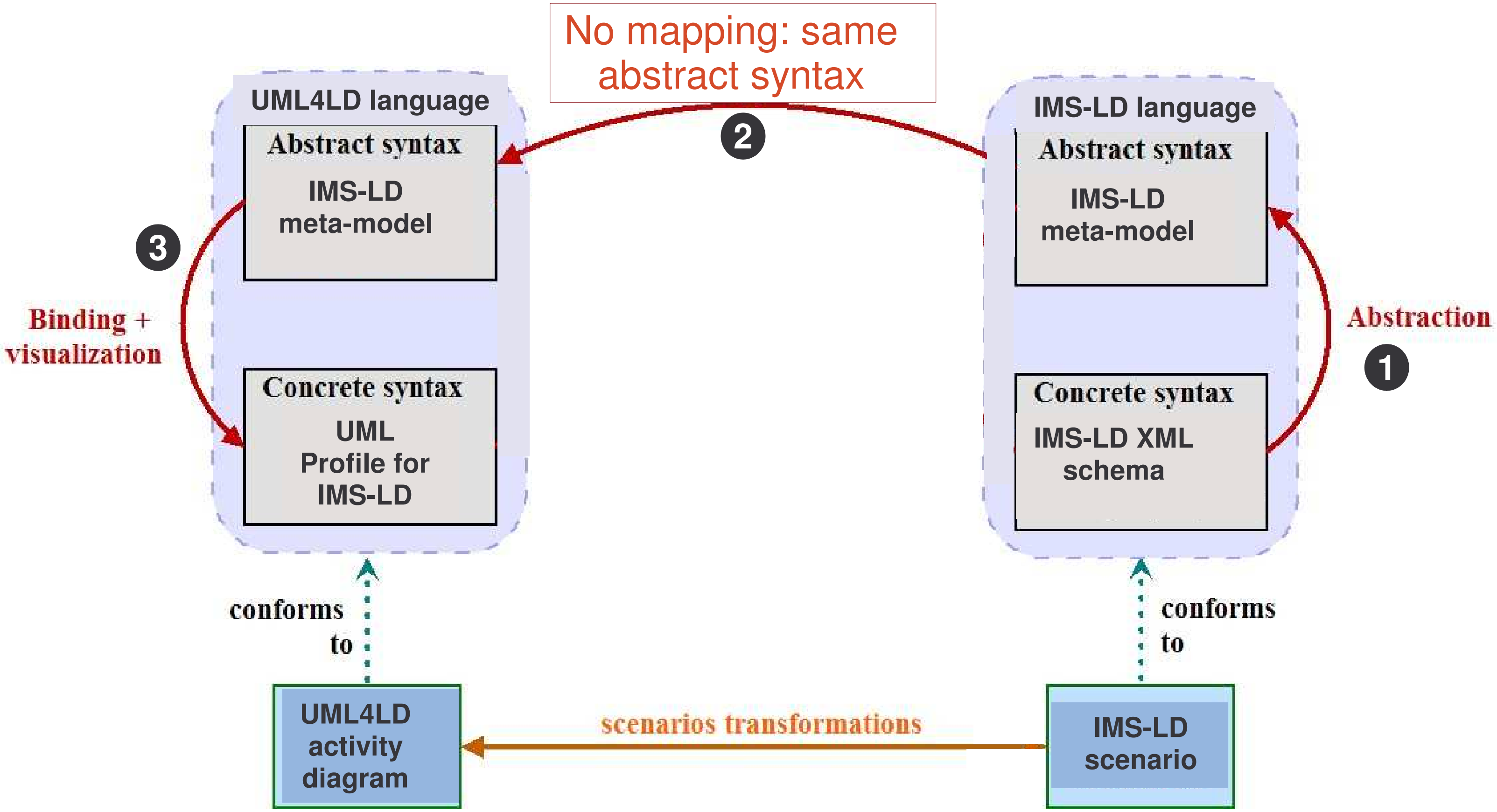


```
<xs:element name="note">
  <xs:complexType>
    <xs:sequence>
      <xs:element name=
```

Example: CPM to IMS-LD transformation



Example: IMS-LD to UML4LD transformation



Summary - Application of MDE-MDA-MDR theory 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**
- Transformation obstacles
  - Abstraction/Binding → Technological obstacles
  - Visualization/Graphical representation
  - **Mapping**: the **hot topic** from this new approach
- 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

This is the END

Thank you!

...Any questions?

Transformation obstacles

- Abstraction/Binding
  - Technological obstacles
  - Learning scenarios deal with various technological space: XML/XSD, UML/MOF, EMF/Ecore, XMI, Java/JMI

- Visualization/Graphical representation
  - Model element *versus* Representation element
  - Difficulty for automatic positioning of representation element
  - Must go further than simple class-association representation

- **Mapping**: the **hot topic** from this new approach
  - Must be carefully tackled
  - Many tools and techniques to test from MDE domain
  - Raises scientific obstacle:
    - The level of **pedagogical expressiveness** for the domain and abstract languages
    - The level of **completeness** and **pedagogical flexibility** claimed by abstract languages

Ongoing work

- **Bottom-up process**
  - **Testing tools and techniques from MDE/MDA research community...**
    - Graphical editors/ model visualizers
    - Transformation languages and tools
  - **... with reference learning scenario languages and models**
    - Need for abstract scenario language reference (IMS-LD, LDL, others ?)
    - Need for domain scenario language reference (CPM, others ?)