ICSOFT EA PT
ICSOFT-PT 2014
9th International Conference on Software Paradigm Trends
29 - 31 August, 2014 Vienna, Austria

A Meta-modeling approach for extending the Instructional Design Semantics of Learning Management Systems & Model Weaving and Pedagogy

Mapping Abstraction Levels in Instructional Design Languages

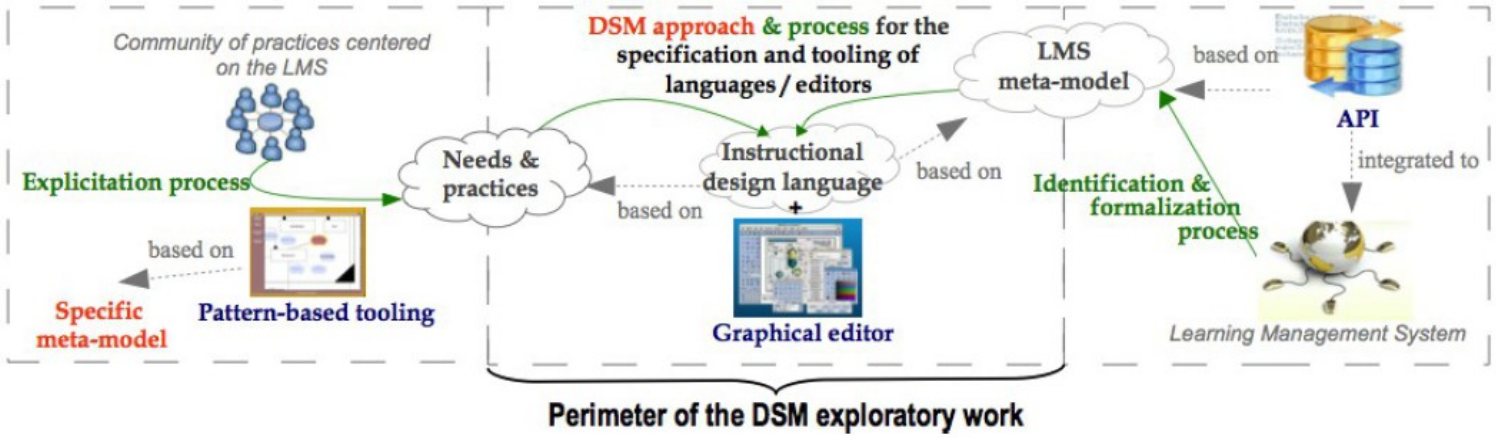
Esteban Loiseau, Pierre Laforcade, Sébastien Iksal

LIUM Université du Maine
esteban.loiseau@univ-lemans.fr, pierre.laforcade@univ-lemans.fr

lium
Université du Maine
l'université nantes angers le mans
PÔLE DE RECHERCHE ET D'ENSEIGNEMENT SUPÉRIEUR
graphiT Graphical VIDLs 4 Teachers

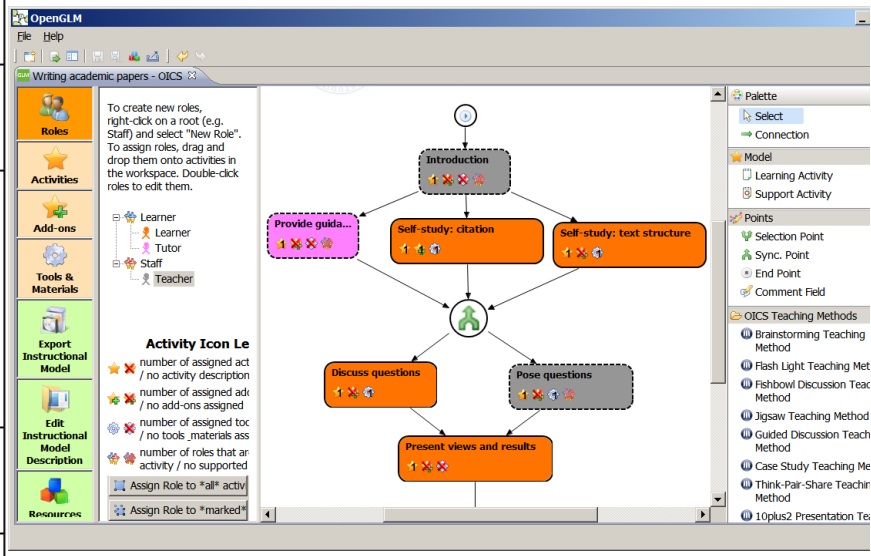
Research context

- ✗ Laboratoire d'Informatique de l'Université du Maine (Le Mans, France)
- ✗ TEL engineering team
- ✗ PhD Thesis
- ✗ GraphiT project: http://www-lium.univ-lemans.fr/~laforcad/graphit/
  - ➔ Funded by French research agency (ANR)
  - ➔ Study expressiveness limits of operationalizable learning design language
    - ➔ Learning scenarios automatically deployed on LMS



What is a learning scenario ?

Learning Goals	Assessment Activities	Learning Activities
Recognize and understand common C# syntax and semantics (foundational knowledge)	*Online study guides (quizzes that students can repeat until right)*	*Read textbook; classroom guided practice*
Design, develop, test, and document custom C# Windows computer applications (application)	*Forward-looking individual C# homework assignments; team project; final exam*	*Forward-looking application design discussions; demos; code-writing activities; student demos*
Apply computer programming solutions to business and personal interests (integration)	*Forward-looking individual C# homework assignments; team project*	*Forward-looking application design discussions; demos; code-writing activities*
• *Discover personal interest in a career as an application developer (human dimension/Self)* • *Develop ability to perform effectively as a member of a work-team (human dimension/Other)*	*Reflective self-evaluation* *Peer feedback & evaluation*	*Discussions (classroom and/or forums); research* *Work on project teams*
Find passion to use computer programming technology to help people and society (caring)	*Team project; reflective self-evaluation*	*Classroom discussions; forums*
Learn how to learn about new [program codes] when they are [issued]. (learning how to learn)	*Treasure hunt homework assignments (students document their learning process)* *(Revise?)*	*Explanations; role-playing; practice treasure hunts; video demos* *(Revise?)*

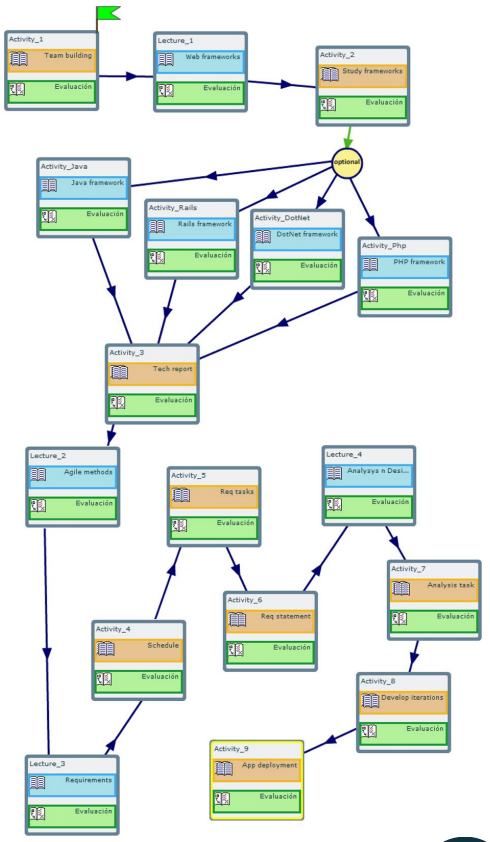


Credits to Jeff Straw retrieved from designlearning.org

OpenGLM screenshot retrieved from http://edutechwiki.unige.ch/

Visual Instructional Design Language

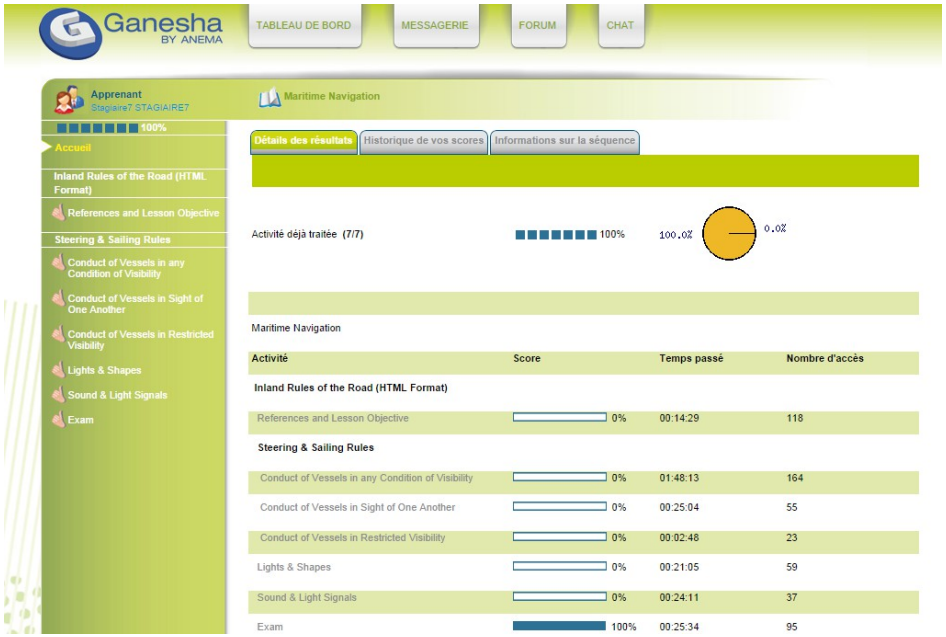
- ✗ To design learning scenarios
- ✗ Support creative thinking and human communication
- ✗ Do not systematically provind binding
  - ➔ Or through IMS-LD (LD standard)



Dodero, J., Martinez del Val, A., Torres, J. 2010. An extensible approach to visually editing adaptative learning activities and designs based on services. In Journal of visual languages & computing 21 : 332-346.

What is a LMS?

Moodle



Ganesha LMS

What is the problem?

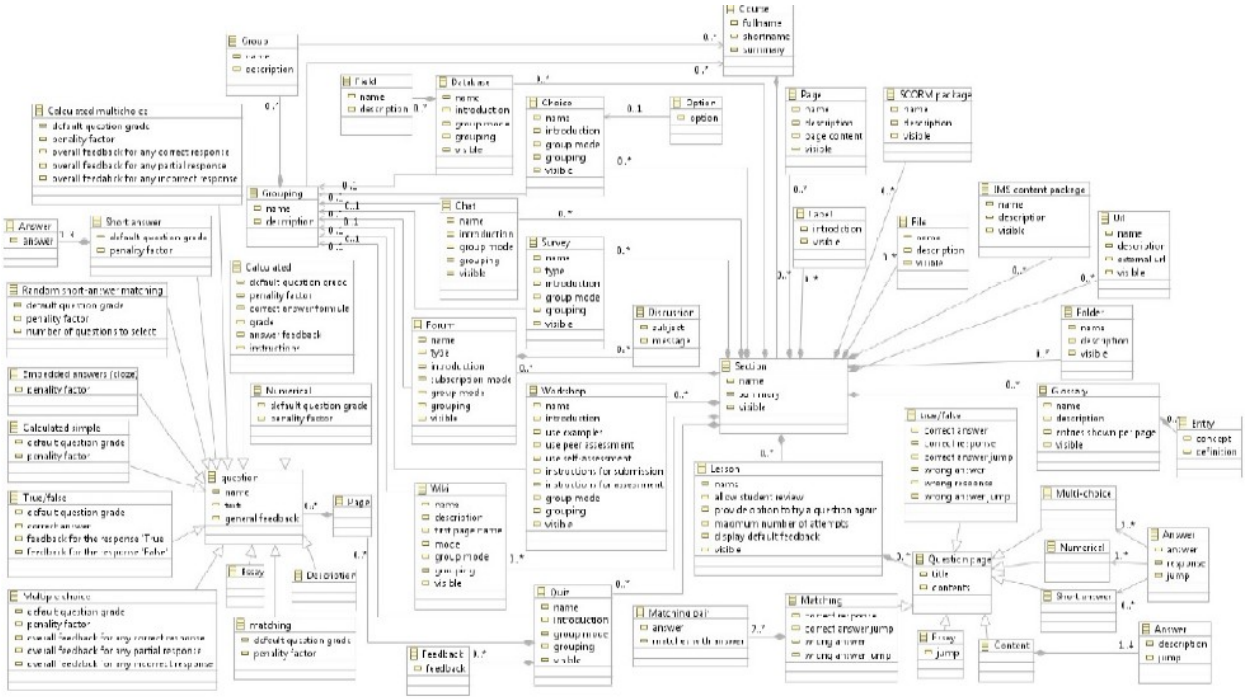
- ✗ Institutions provide LMS to teachers and students
  - ➔ Rarely up-to-date software
- ✗ Teachers are (sometimes) taught how to use it
    - ➔ Not how to design learning situations on the LMS
- ✗ Binding between LD standard and LMS not successful
  - ➔ Loss of pedagogical information
  - ➔ Hard-coded / non flexible mechanisms

What we want to do

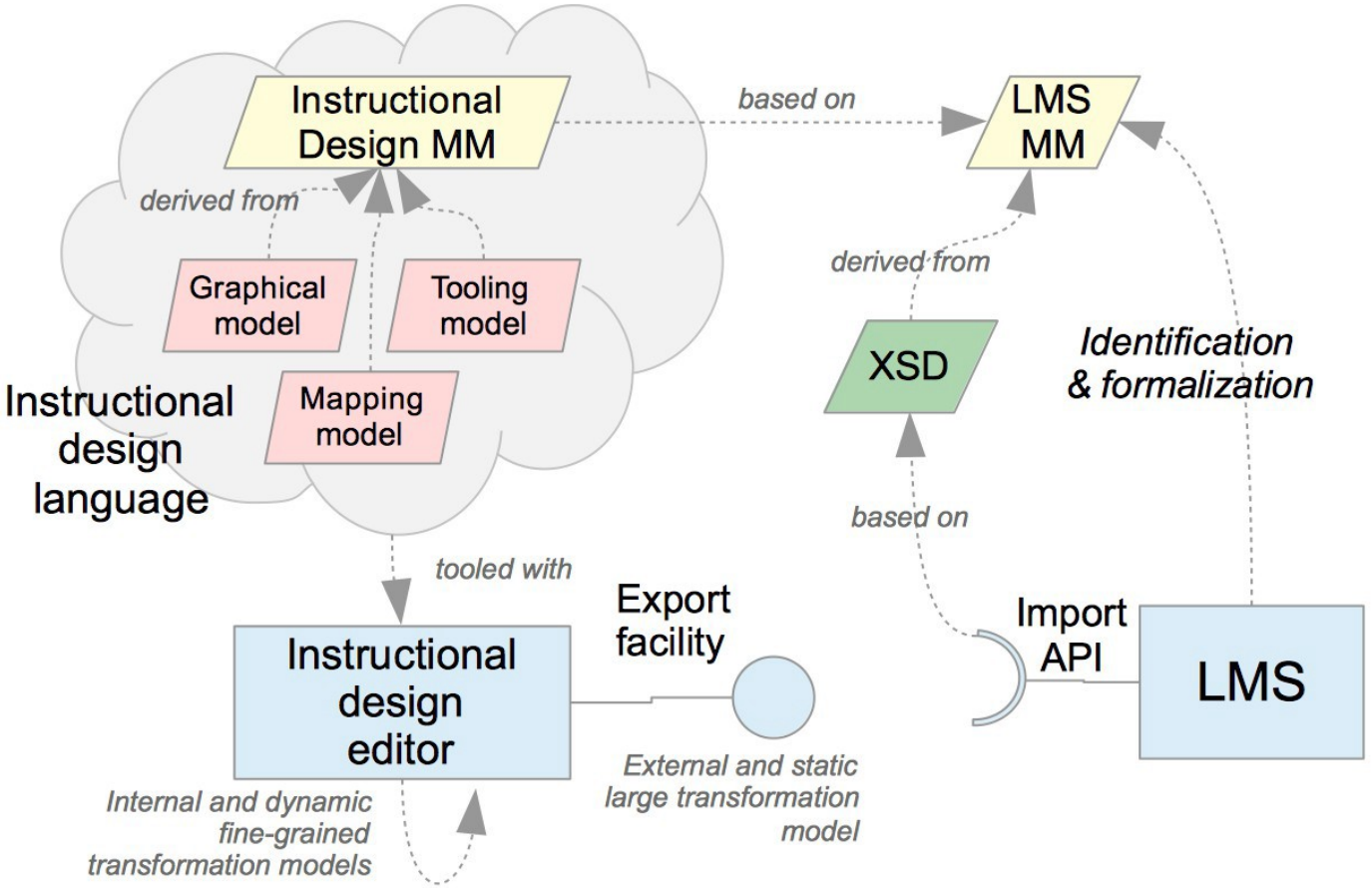
- ✗ Provide teachers with graphical learning design language
  - ➔ “compatible” with LMS
- ✗ Help to focus on the pedagogical aspect of the scenario
- ✗ Foster individual reflection about learning design
- ✗ Improve use of existing LMS

What is already done

- ✗ LMS-compatibility layer:
  - ➔ LMS Metamodel: Moodle and Ganesha (WIP) (conceptual model)
  - ➔ LMS learning scenario file format: XML Schema
  - ➔ Learning scenario deployment plugin



Overview


Instructional Design MM
based on
LMS MM
derived from
Graphical model
Tooling model
Mapping model
Instructional design language
derived from
XSD
Identification & formalization
based on
tooled with
Export facility
Import API
LMS
Instructional design editor
External and static large transformation model
Internal and dynamic fine-grained transformation models

What I do

- ✗ Only one design language: at a higher abstraction level
- ✗ Keep the LMS compatibility
- ✗ Focus on one platform : Moodle
  - ➔ Open-source , modular: easy to extend, customize if needed
  - ➔ Large community of users
  - ➔ “Used at home”
- ✗ Domain Specific Modeling approach
  - ➔ Code generation tools
  - ➔ Model = Abstraction

How ?

- ✗ Designing a new language with pedagogical concepts
- ✗ Extending the Moodle Metamodel
  - ➔ To be able to use the file format / deployment plugin
- ✗ Making sure every pedagogical concept can be implemented in Moodle
- ✗ Asking teachers what they want

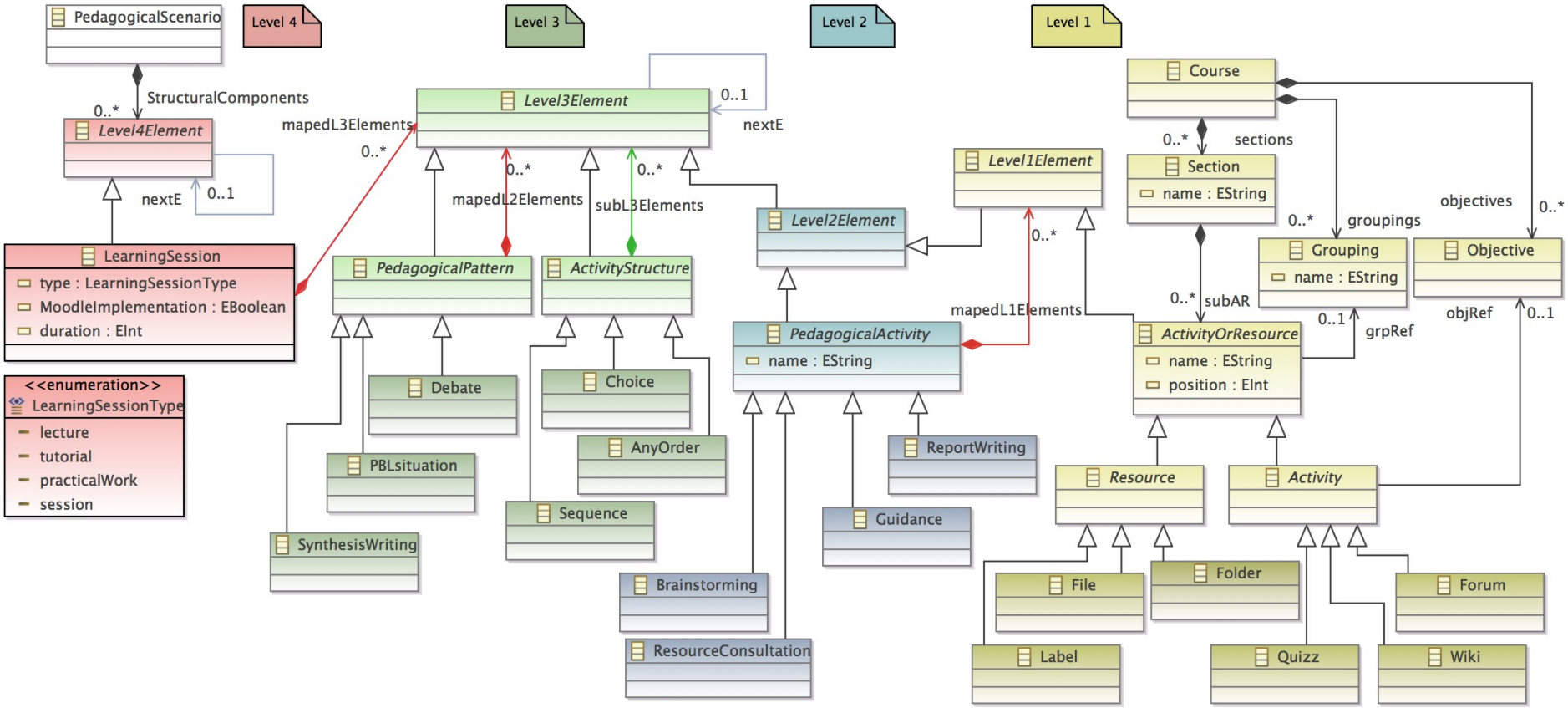
Teachers requirements

- ✗ Collected through
  - ➔ Survey (200+ teachers)
  - ➔ Interviews
  - ➔ Study of existing courses
- ✗ High level pedagogical building blocks
  - ➔ And LMS specific ones too!
- ✗ Have **default** LMS implementations for high level concepts
  - ➔ And be able to modify it!
  - ➔ Or not having it
- ✗ Have several types of structures
  - ➔ Sequences
  - ➔ Choices ...

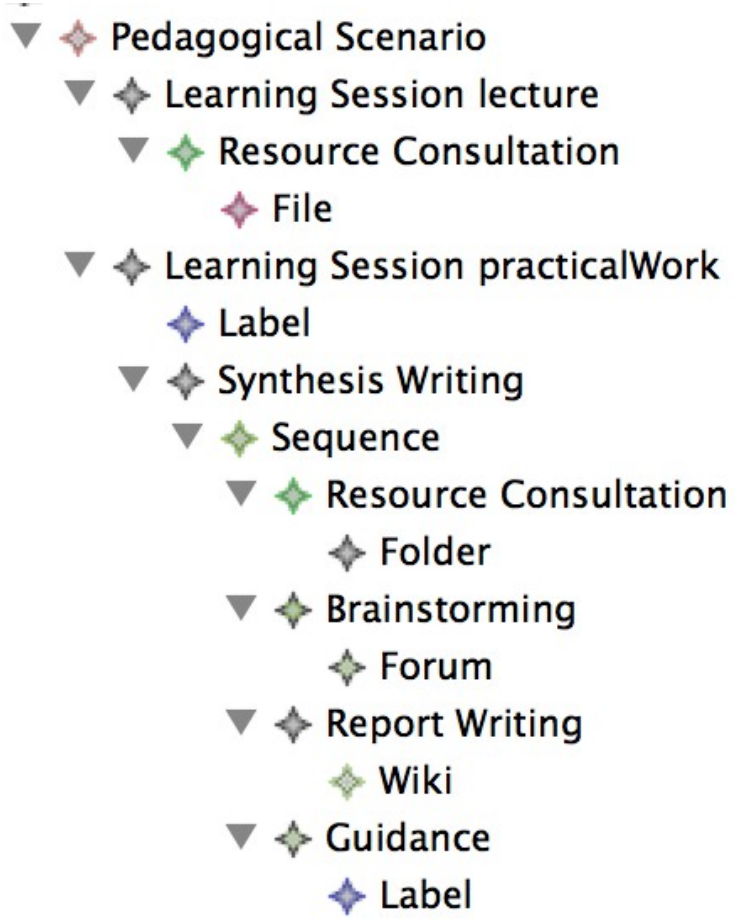
Abstract concept examples

- ✗ Pedagogical activity:
  - ➔ Exchange activity: students communicating with each other
  - ➔ Using the chat or forum feature in Moodle
    - ➔ How to decide? Synchronous property
- ✗ Pedagogical pattern:
  - ➔ Higher level
  - ➔ Synthesis writing: sequence of pre-defined pedagogical activities
    - ➔ Resource consultation
    - ➔ Brainstorming
    - ➔ Report Writing
- ✗ Activities structures:
  - ➔ Sequence: students have to complete all the tasks

Metamodel


PedagogicalScenario
Level 4
Level 3
Level 2
Level 1
StructuralComponents
0..*
Level4Element
nextE
0..1
mapedL3Elements
0..*
Level3Element
0..1
nextE
mapedL2Elements
0..*
subL3Elements
0..*
Level2Element
Level1Element
LearningSession
type : LearningSessionType
MoodleImplementation : EBoolean
duration : EInt
PedagogicalPattern
ActivityStructure
mapedL1Elements
0..*
PedagogicalActivity
name : EString
<<enumeration>>
LearningSessionType
lecture
tutorial
practicalWork
session
Debate
Choice
AnyOrder
PBLsituation
Sequence
SynthesisWriting
Brainstorming
ResourceConsultation
ReportWriting
Guidance
Course
0..*
sections
Section
name : EString
0..*
groupings
Grouping
name : EString
objectives
0..*
Objective
0..*
subAR
0..1
grpRef
objRef
0..1
ActivityOrResource
name : EString
position : EInt
Resource
Activity
File
Folder
Label
Quizz
Forum
Wiki

Learning scenario example



Automatic mapping

- ✗ “Default implementation” requirement
  - ➔ Automatically add instances to the models
  - ➔ Automatically set properties values
  - ➔ Automatically add implementation instances to the composition relationship
- ✗ For levels 2 & 3 only
- ✗ Using hand written model transformations
  - ➔ It works but...
  - ➔ A lot of rules
  - ➔ Complex business logic
  - ➔ Hard to maintain
  - ➔ Hard to understand

Why not modeling?

Mapping examples

- *Exchange activity: if synchronous then use chat else use forum*
- *Report writing activity: 3 boolean criteria and 4 possible implementations*

	Journal	Wiki	Assignment (file upload)	Assignment (online text)
Online	Y	Y	N	Y
Collaborative	N	Y	/	N
Iterative	Y	/	/	N

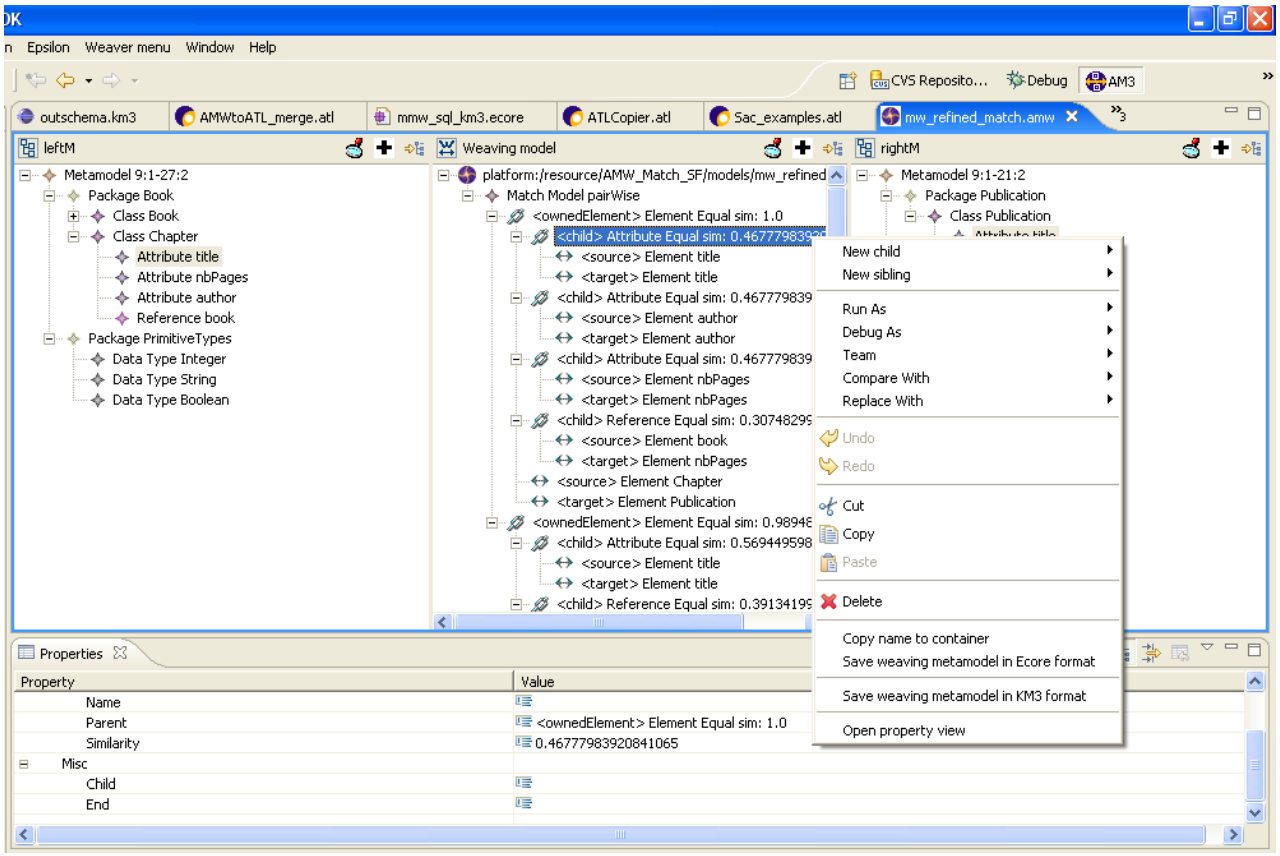
- *Some implementations only differ from the settings of the feature*
- *Also depending of the pedagogical element properties*

Model weaving

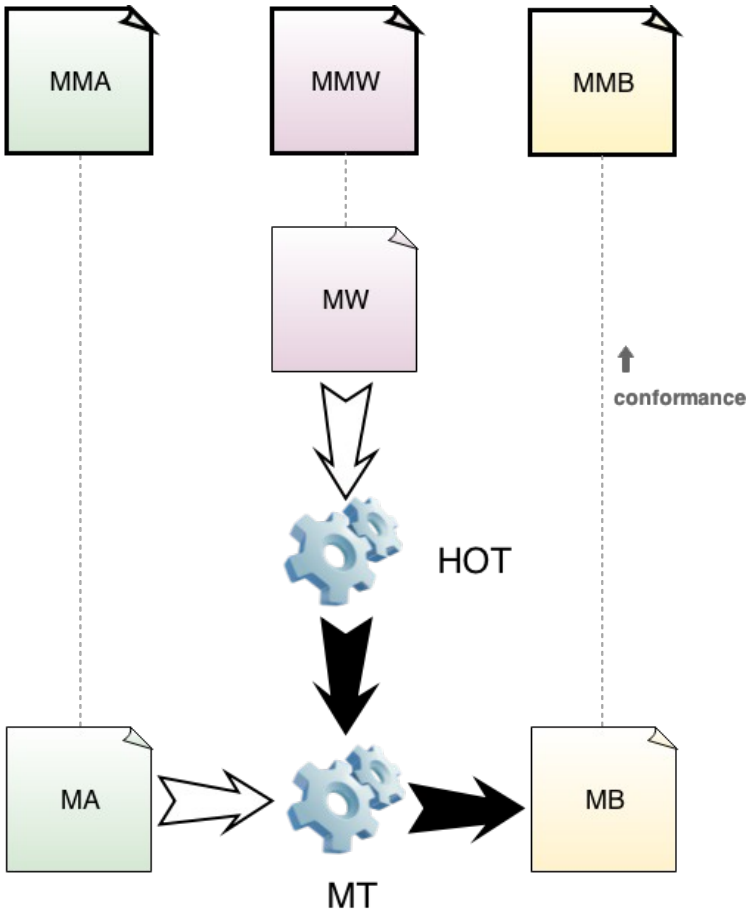
- ✗ Weaving model = referencing other models (woven models) elements
- ✗ Possible usage:
  - ➔ Model checking through pairing
  - ➔ Lazy loading of elements (in large models)
  - ➔ Non obtrusive refining of models
  - ➔ **Mapping formalization**
- ✗ Challenge: how to make a weaving model executable?

AMW

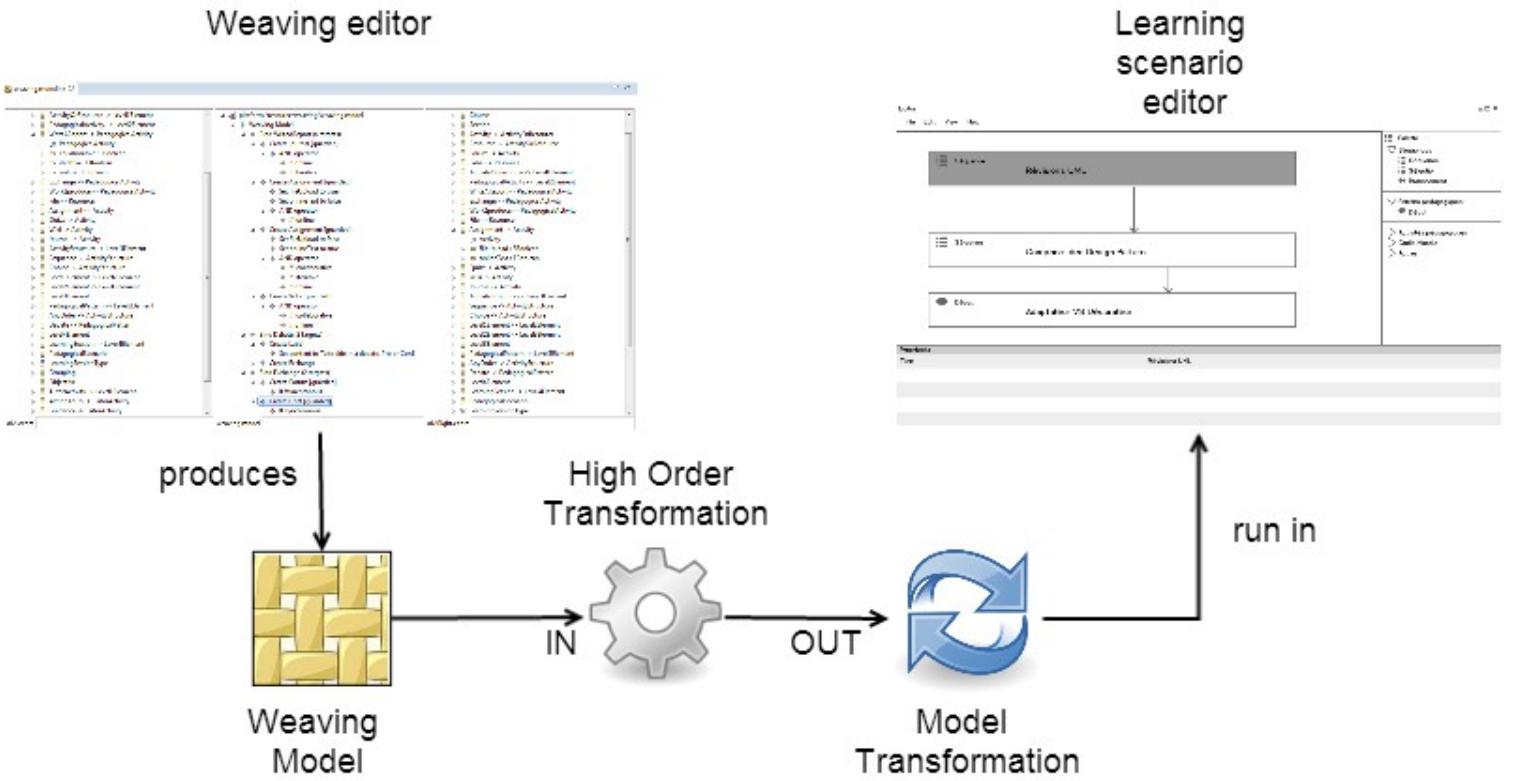
- AtlanMod Model Weaver
- Basic weaving metamodel
- Graphical tree editor
- Matching transformations support
- Several use cases and examples
- EMF compatible
- ATL based
- Outdated



AMW architecture



Our model weaving use case



Epsilon project

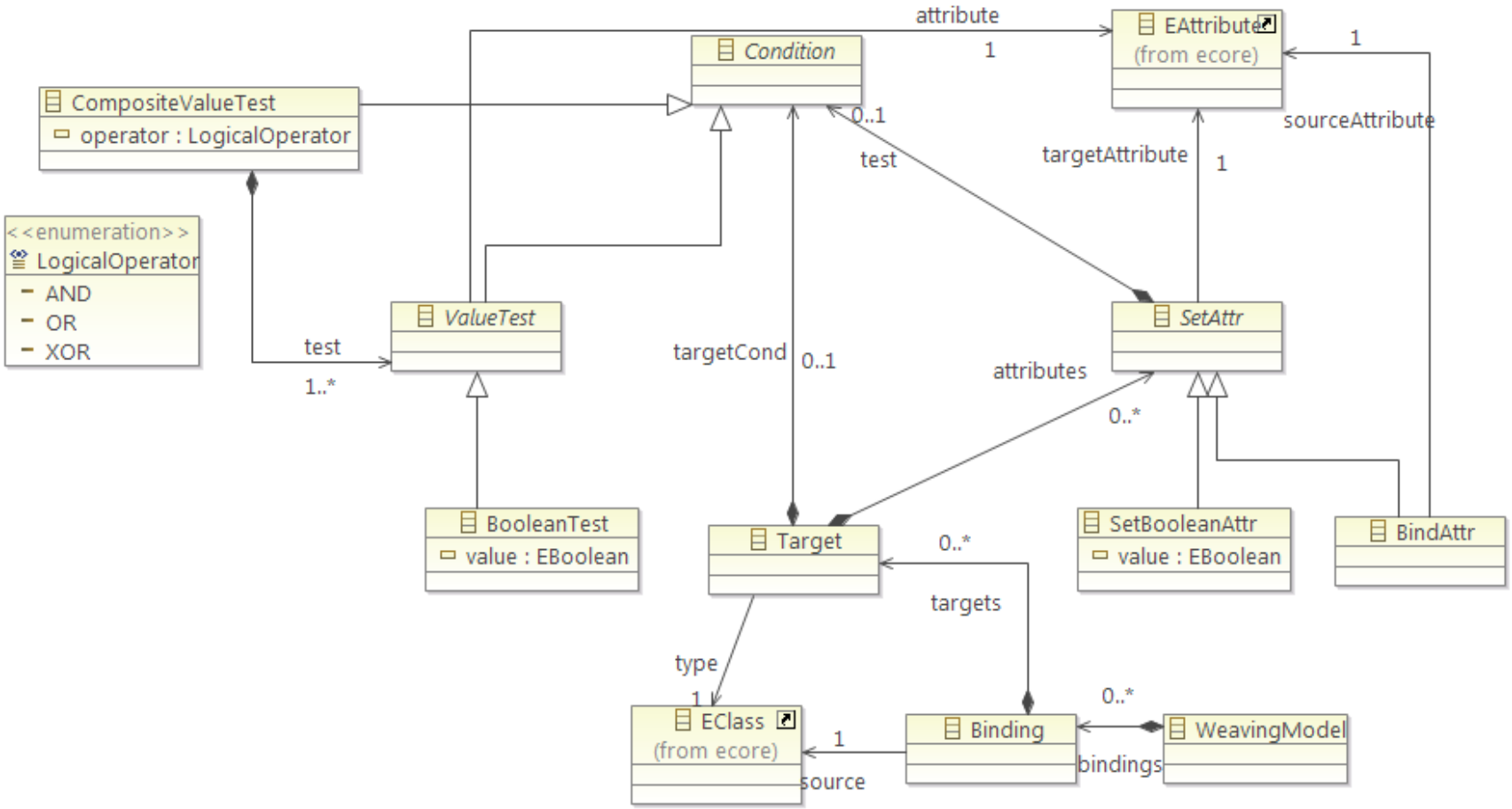
- ✗ Eclipse foundation project
- ✗ Collection of tools and language for MDD
  - ➔ Merging
  - ➔ Migrating
  - ➔ Transforming (M2T, M2M)
  - ➔ Validating
  - ➔ Comparing
  - ➔ …
- ✗ Active community
- ✗ Up-to-date compatibility with EMF

http://www.eclipse.org/epsilon/

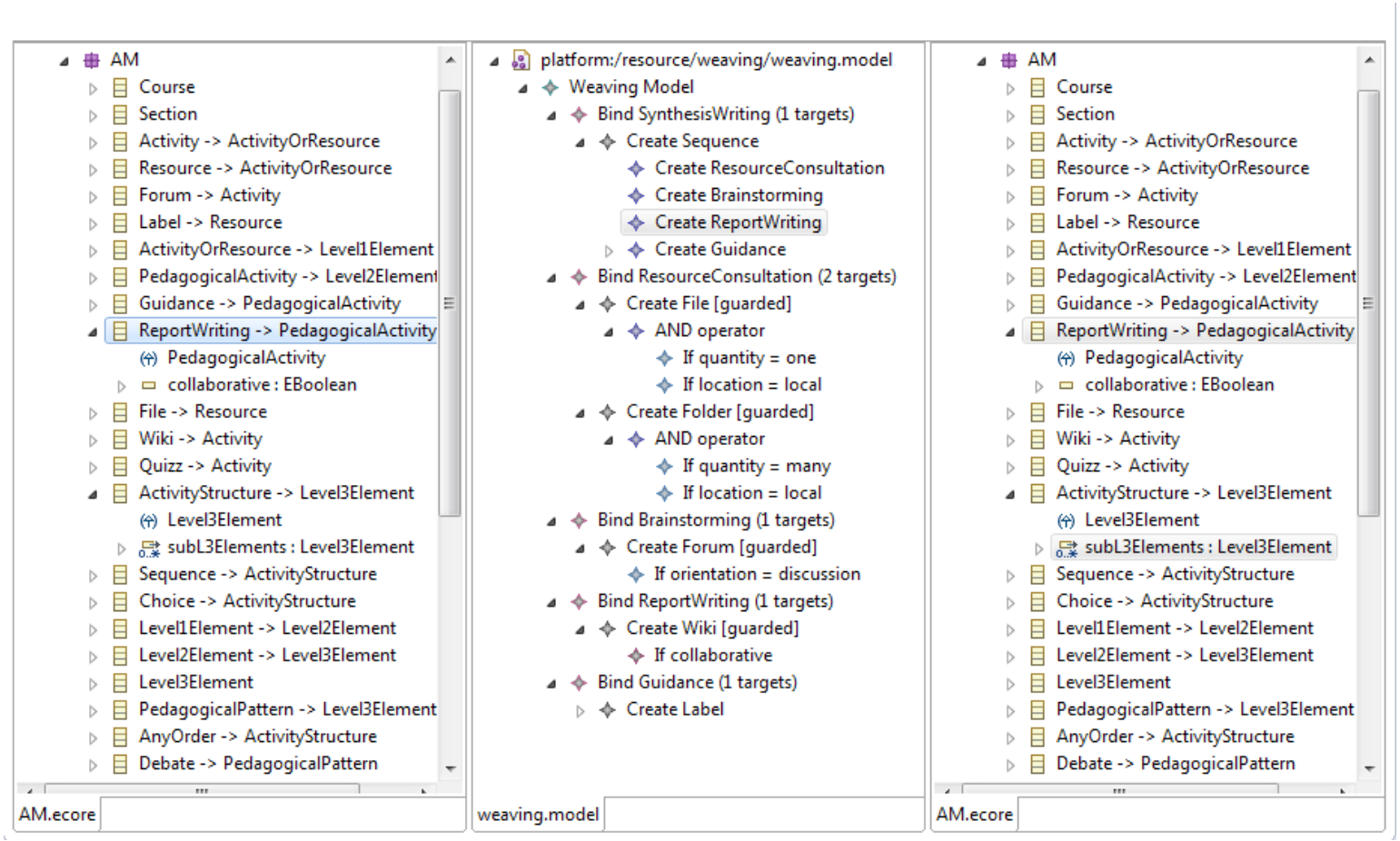
How we use it

- ✗ Custom weaving Metamodel (Ecore)
- ✗ Model weaving editor: ModeLink
- ✗ HOT through M2T template: EGL
- ✗ Model transformations with EOL

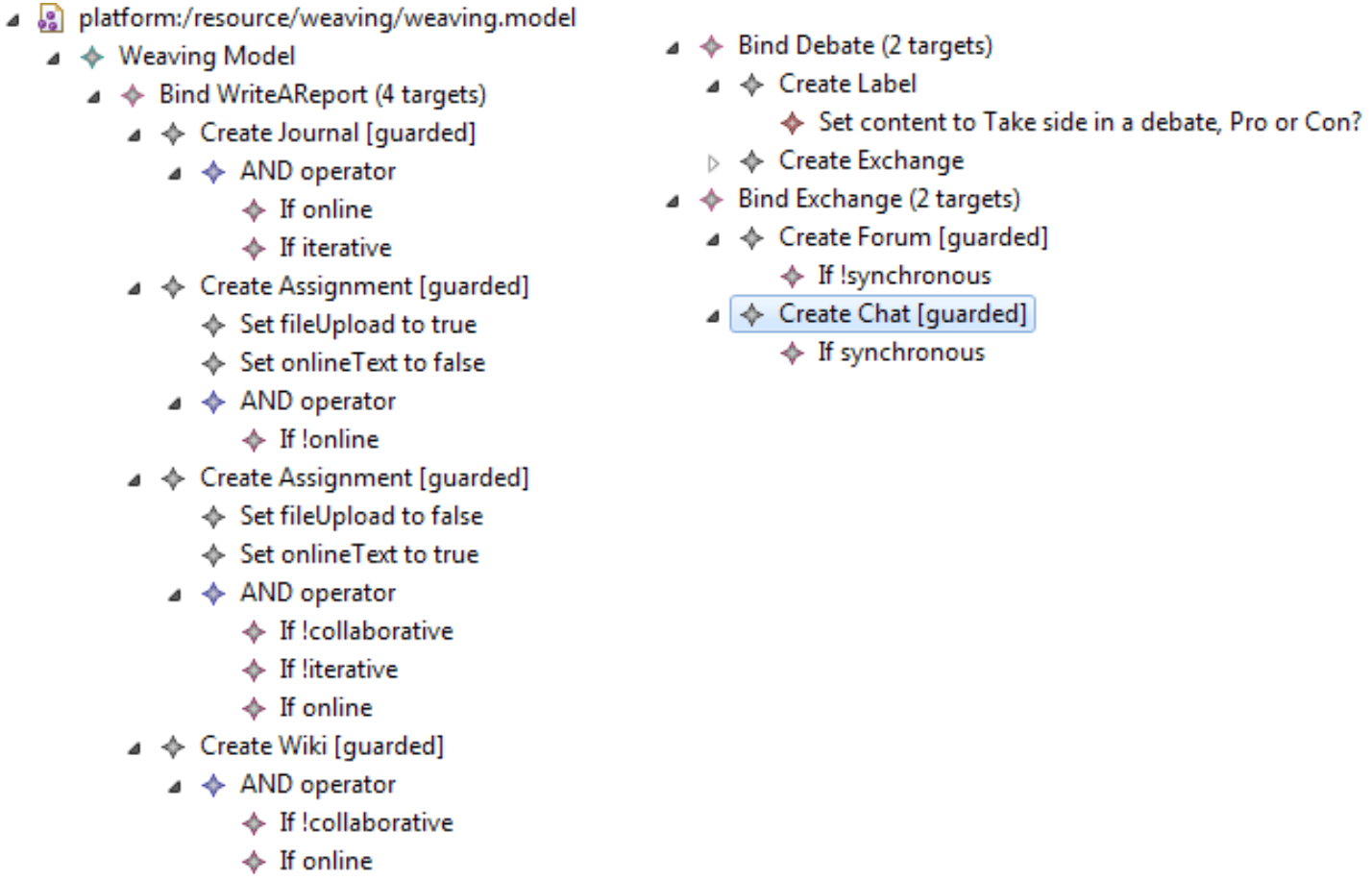
Weaving metamodel



Example weaving model (UI)



Example weaving model



EGL HOT

```
[%
import "./hotOperations.eol";
var sourceClassName:String = binding.source.getName();
var sourceVarName:String = sourceClassName.firstToLowerCase();
%]

operation source![%=sourceClassName%] addMapping(element:source!Level3Element) {
    if(element.isKindOf(source!Level1Element)) {
        self.mapedL1Elements.add(element);
    }
    else if(element.isKindOf(source!Level2Element)) {
        self.mapedL2Elements.add(element);
    }
    else {
        self.mapedL3Elements.add(element);
    }
}

operation source![%=sourceClassName%] bind[%=sourceClassName%]() {

    [% for(targetElement:Target in binding.targets) {
        var cond:Boolean = targetElement.targetCond.isDefined();
        var targetClassName:String = targetElement.type.getName();
        var targetVarName:String = targetClassName.firstToLowerCase();
        if(cond) {%]
            if([%=targetElement.targetCond.formatCondition()%]){
        [%}%]
                var [%=targetVarName%]:[%=targetClassName%] = new source![%=targetClassName%];
                self.addMapping([%=targetVarName%]);
        [%if(cond){%]
            }
        [%}
    }%]
}
```

Model transformations

```
operation source!SynthesisWriting addMapping(element:source!Level3Element) {
    if(self.isKindOf(source!Level2Element)) {
        self.mapedL1Elements.add(element);
    }
    else if(self.isKindOf(source!Level3Element)) {
        self.mapedL2Elements.add(element);
    }
    else if(self.isKindOf(source!Level4Element)){
        self.mapedL3Elements.add(element);
    }
}

operation source!SynthesisWriting bindSynthesisWriting() {

    var sequence = new source!`Sequence`;
    var resourceConsultation = new source!`ResourceConsultation`;

    sequence.subL3Elements.add(resourceConsultation);
    var brainstorming = new source!`Brainstorming`;

    sequence.subL3Elements.add(brainstorming);
    var reportWriting = new source!`ReportWriting`;

    sequence.subL3Elements.add(reportWriting);
    var guidance = new source!`Guidance`;
    guidance.public = Public#tutor;

    sequence.subL3Elements.add(guidance);
    self.addMapping(sequence);
}
```

What we do have

- ✗ Requirements
- ✗ LD language metamodel
- ✗ Mapping solution through Model weaving
  - ➔ Weaving Metamodel
  - ➔ High Order transformation
  - ➔ Simple editor
- ✗ Ideas 

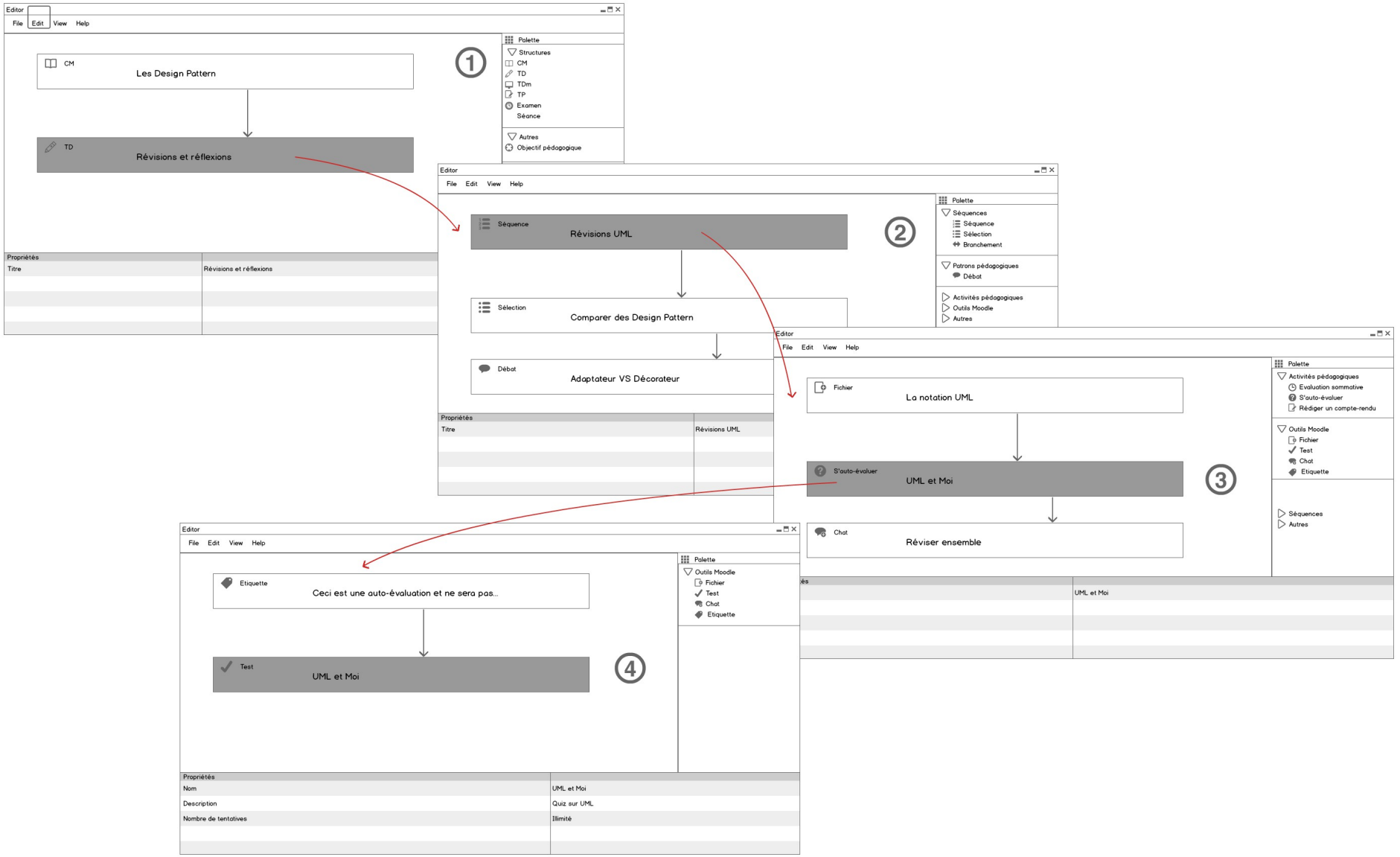
What we don't have (yet)

- ✗ A GMF diagram editor (WIP)
- ✗ Integrated model transformations
- ✗ More mapping use-cases
- ✗ A more user-friendly weaving model editor
- ✗ A cleaning transformation 
- ✗ More ideas 

Thanks for your attention

Questions ?

Diagram editor mockup



Practises analysis