FROM RESEARCH TO INDUSTRY

cea tech

list

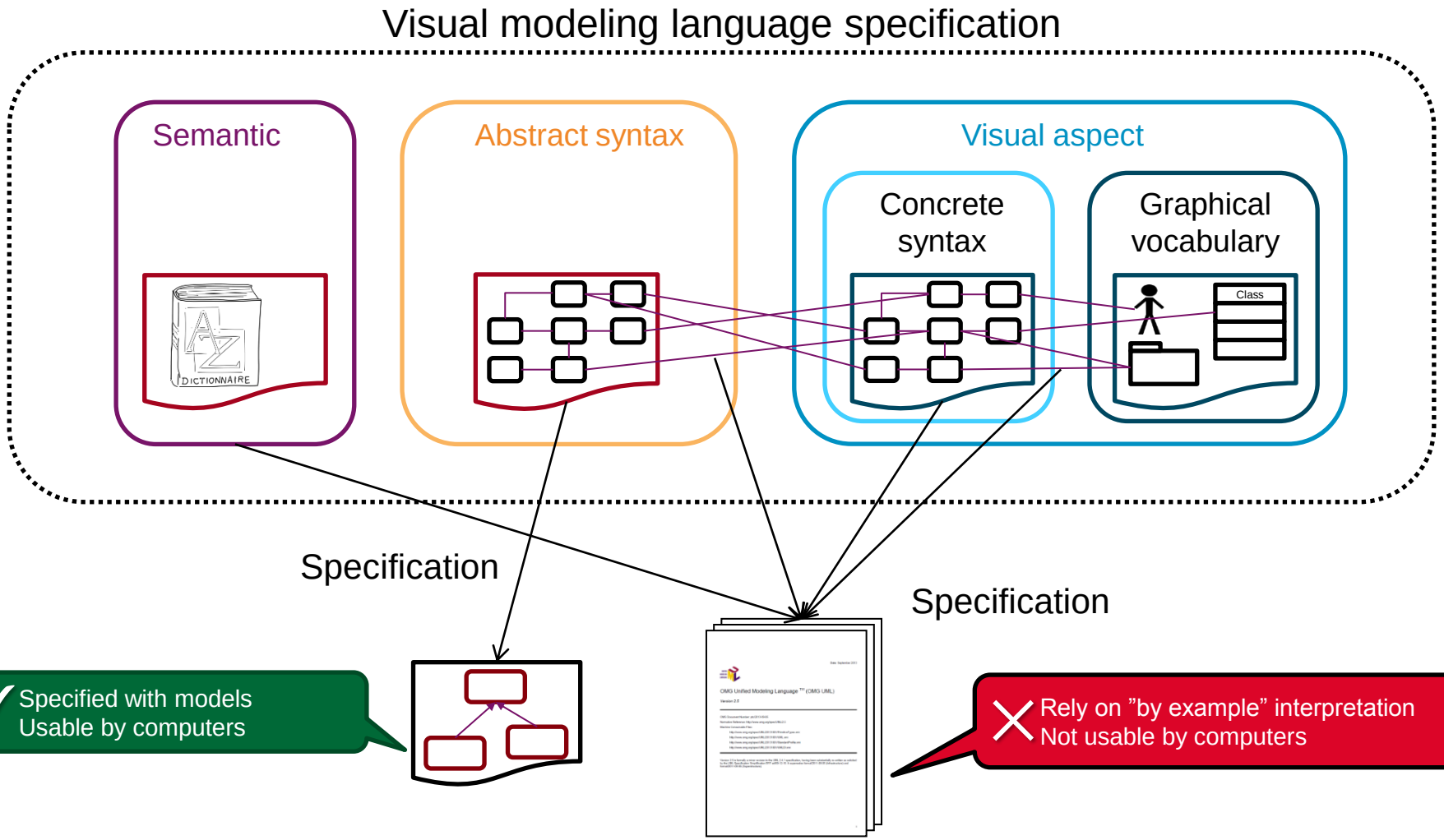
EVALUATION OF THE DIAGRAM DEFINITION STANDARD FOR VISUAL LANGUAGE SPECIFICATION

DRT/LIST/DILS/Laboratoire d'Ingénierie dirigée par les modèles pour les Systèmes Embarqués

June 22nd 2015

INSTITUT CARNOT CEA LIST digiteo

THE MOTIVATION



THE MOTIVATION

GMF-Tooling provides a formalism to specify a modeling language, and then generate the appropriate diagram editor

UML spec

Tool builder

Domain model (*.ecore)

Graphical definition (*.gmfgraph)

Tools definition (*.gmftool)

Mapping model (*.gmfmap)

Generator model (*.gmfgen)

Diagram editor (plugin Eclipse)

✕ Rely on ”by example” interpretation
Not usable by computers

THE MOTIVATION

GMF-Tooling provides a formalism to specify a modeling language, and then generate the appropriate diagram editor

UML spec

Tool builder

Generator model (*.gmfgen)

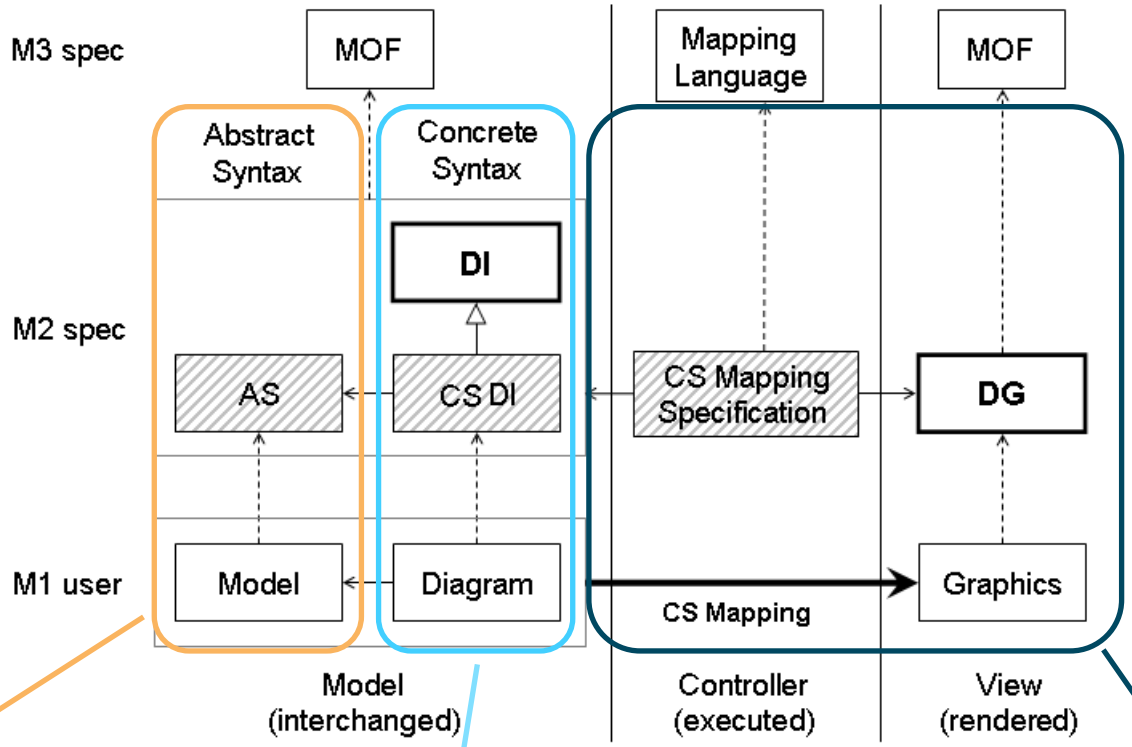
Diagram editor (plugin Eclipse)

✕ Rely on ”by example” interpretation
Not usable by computers

✕ This process has never been used because the GMFGen model had to be adapted anyway

THE DIAGRAM DEFINITION ARCHITECTURE

Diagram Definition support the MVC pattern required to provide tooling for a visual language [OMG12].



Abstract Syntax

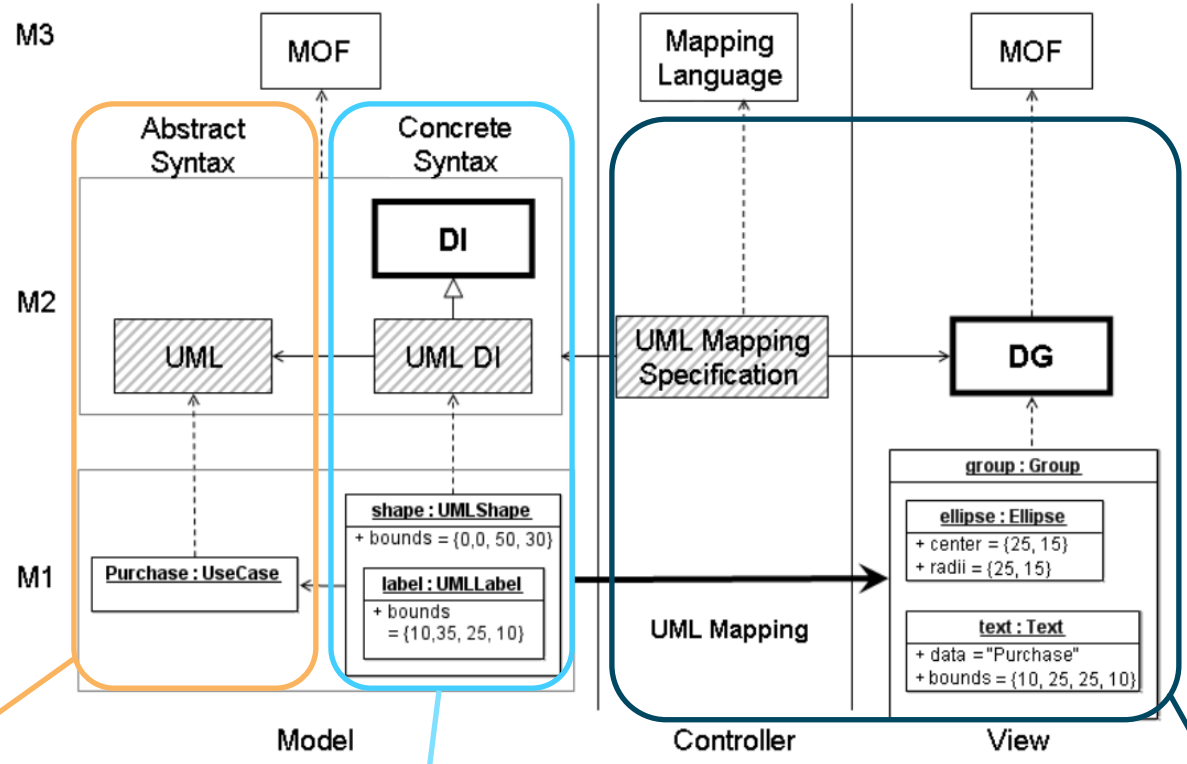
Concrete Syntax

Graphical Vocabulary

[OMG12] Object Management Group, *Diagram Definition 1.0 - formal/2012-07-01*, OMG, 2012.

THE DIAGRAM DEFINITION ARCHITECTURE

UMLDI exemplify the use of DD for the specification of the UML concrete syntax [OMG12].

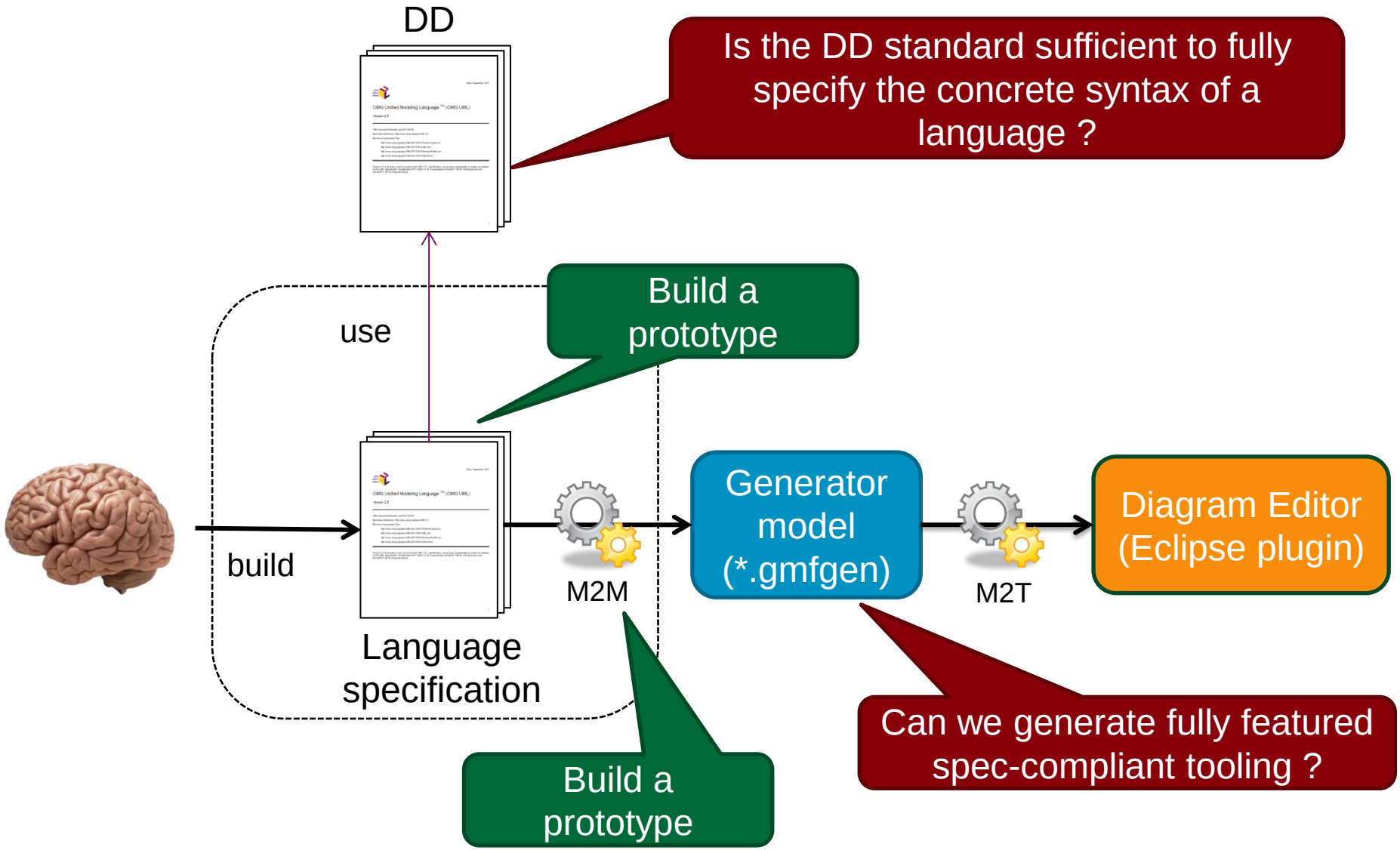


Abstract Syntax

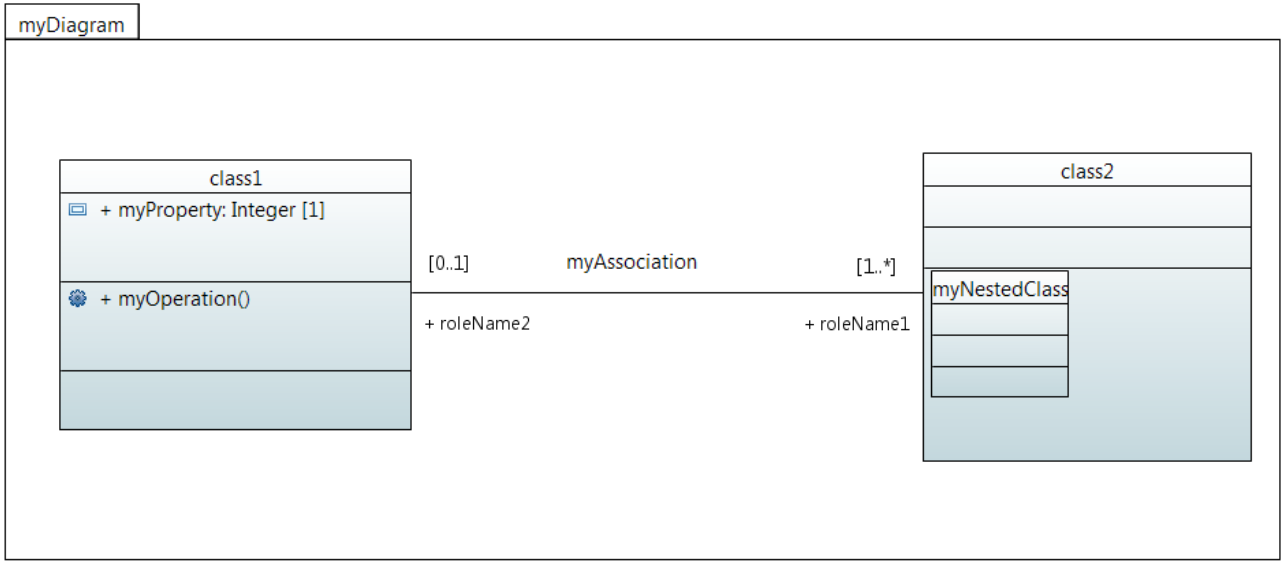
Concrete Syntax

Graphical Vocabulary

OBJECTIVES OF THE EVALUATION APPROACH



USE CASE



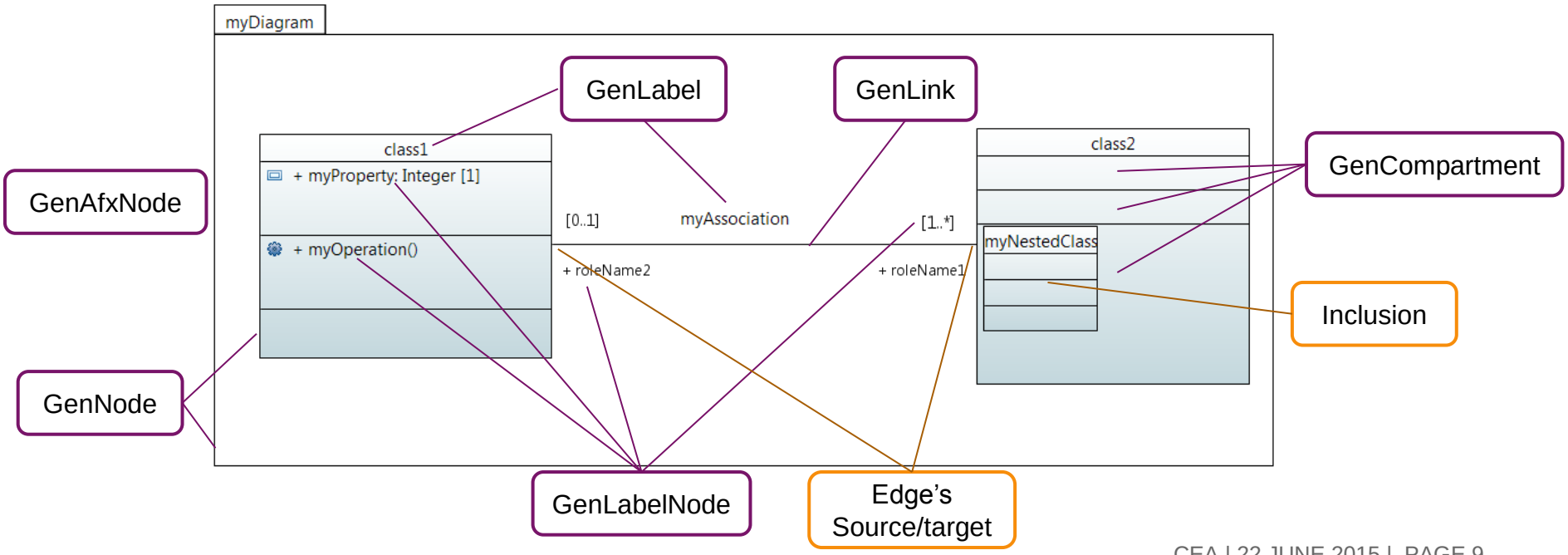
Restrictions:

- 1-1 mapping to the abstract syntax
- Reuse UML abstract syntax from Papyrus
- Reuse UML rendering implementation from Papyrus

ANALYSIS OF GMFGEN REQUIREMENTS

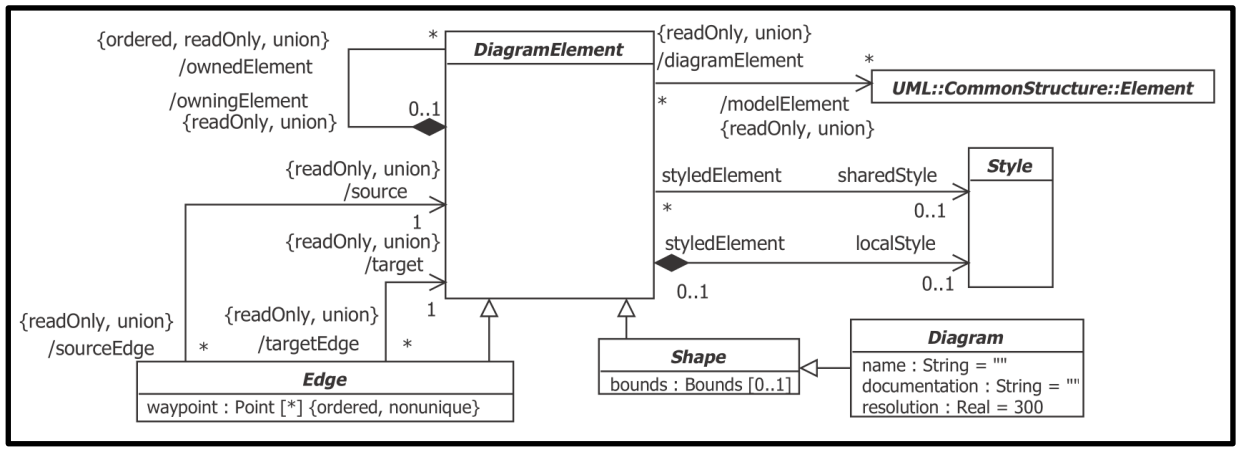
What we need to know to build the GMFGen model:

- The different kind of graphical element's behavior
- The different kind of graphical relations
- Connections with the abstract syntax (ModelFacet)
- Connections with graphical vocabulary (FigureViewMap)



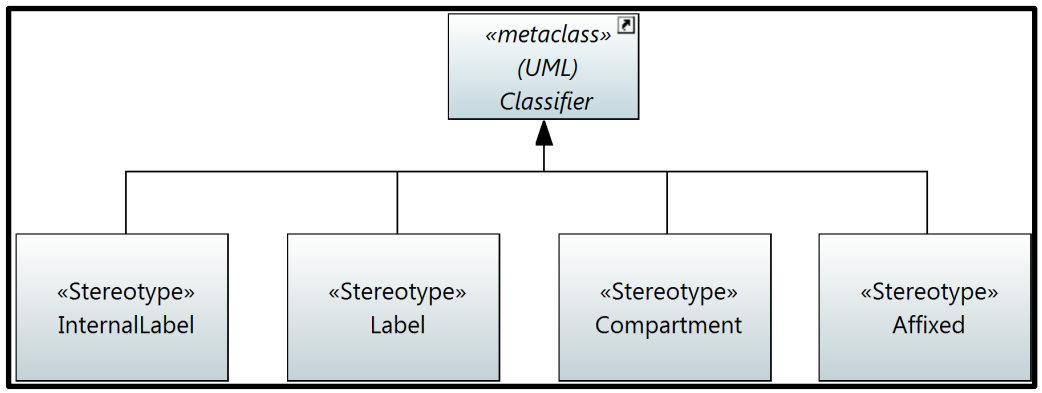
DOES DD ANSWER THE GMFGEN REQUIREMENTS ?

Do we know enough about graphical element's behavior distinction ?



DI meta-model

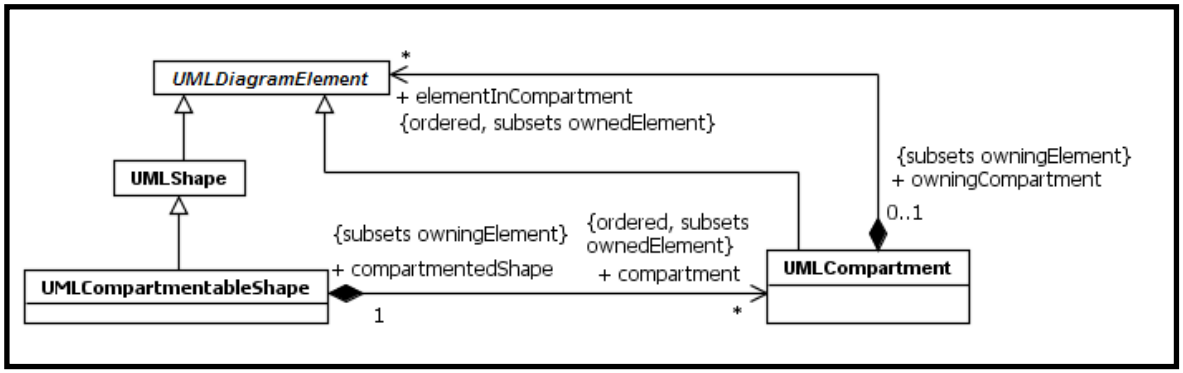
Only three kind of graphical elements



Use of a UML profile

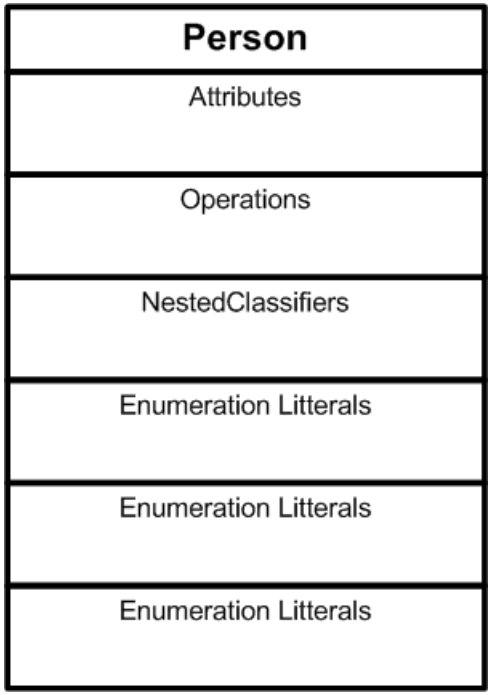
DOES DD ANSWER THE GMFGEN REQUIREMENTS ?

Do we know enough about graphical relations ?



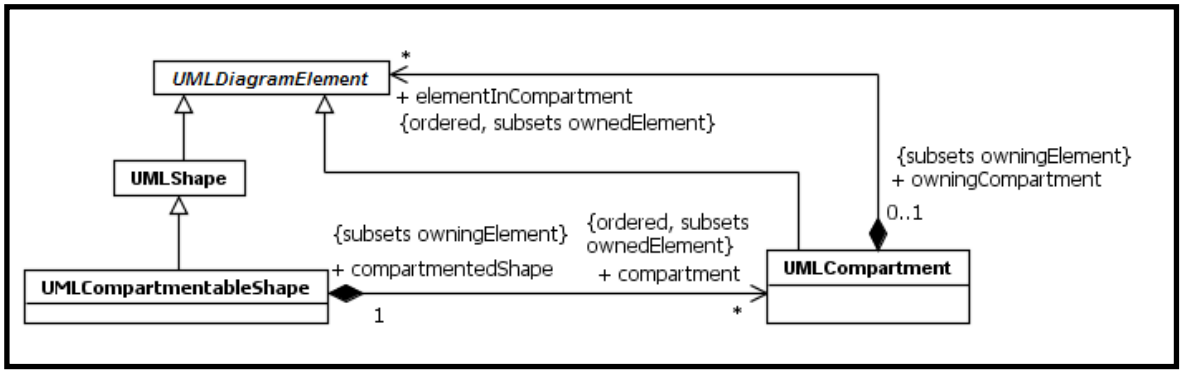
Extract of UMLDI meta-model

Not enough information on graphical relations



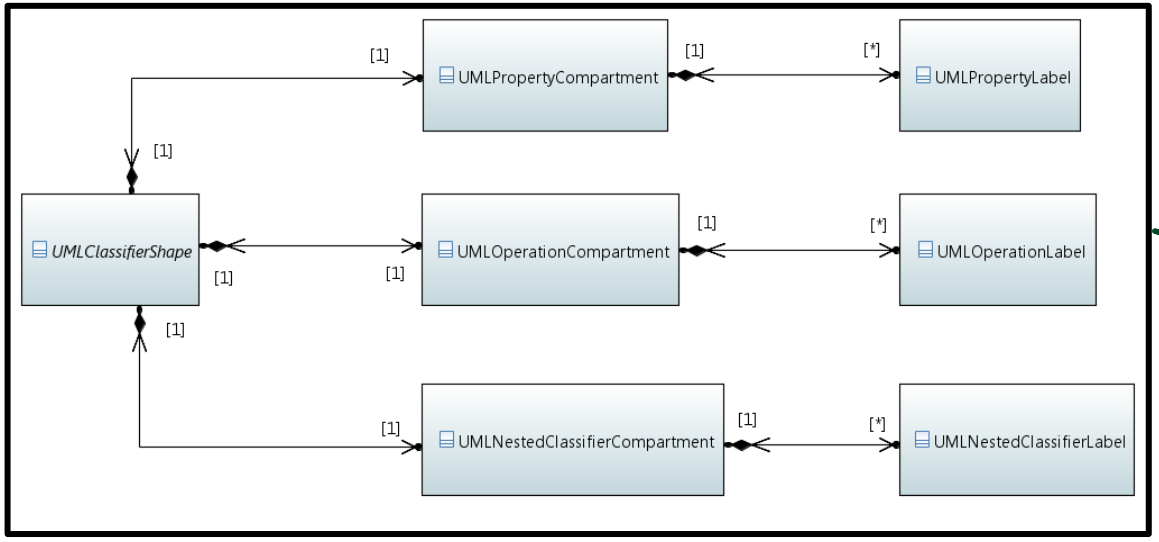
DOES DD ANSWER THE GMFGEN REQUIREMENTS ?

Do we know enough about graphical relations ?



Extract of UMLDI meta-model

Not enough information on graphical relations

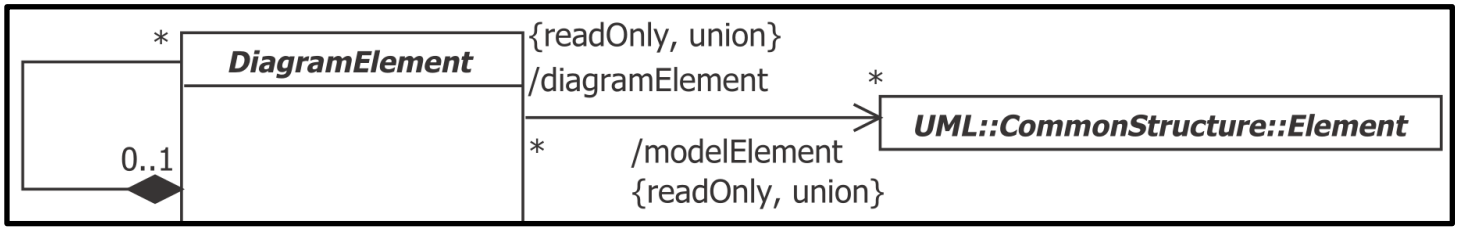


Syntax enrichment

DOES DD ANSWER THE GMFGEN REQUIREMENTS ?

Do we know enough about abstract syntax connections ?

Each graphical element can be linked to any abstract syntax elements

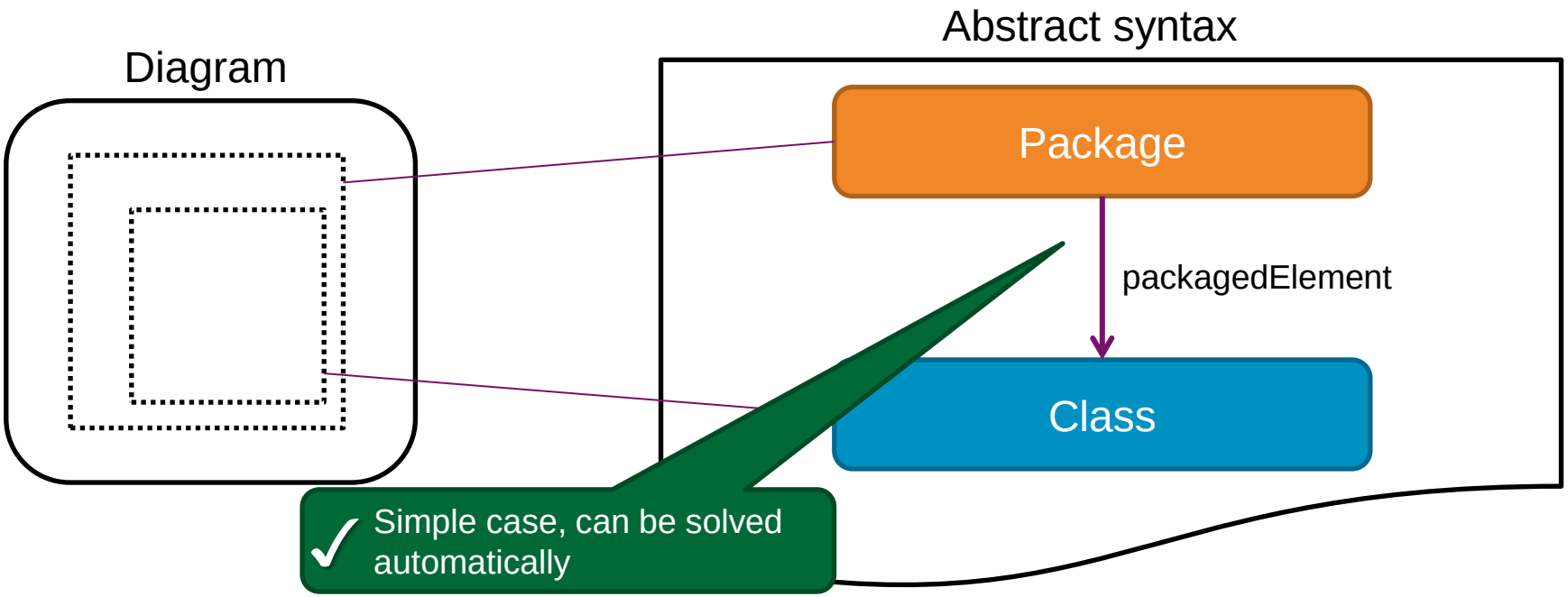


Extract of DI meta-model

From that, can we automatically deduce connections between graphical inclusion relations and abstract syntax containments ?

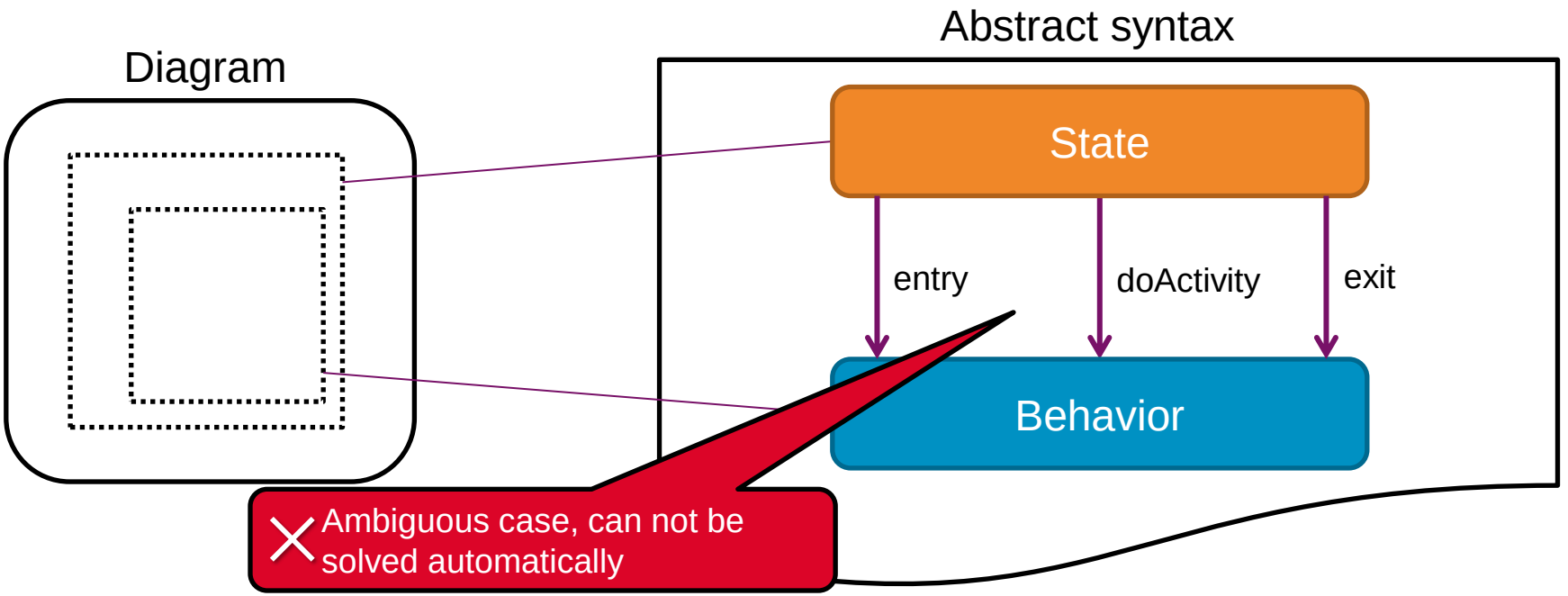
DOES DD ANSWER THE GMFGEN REQUIREMENTS ?

From that, can we automatically deduce connections between graphical relations and abstract syntax containments ?



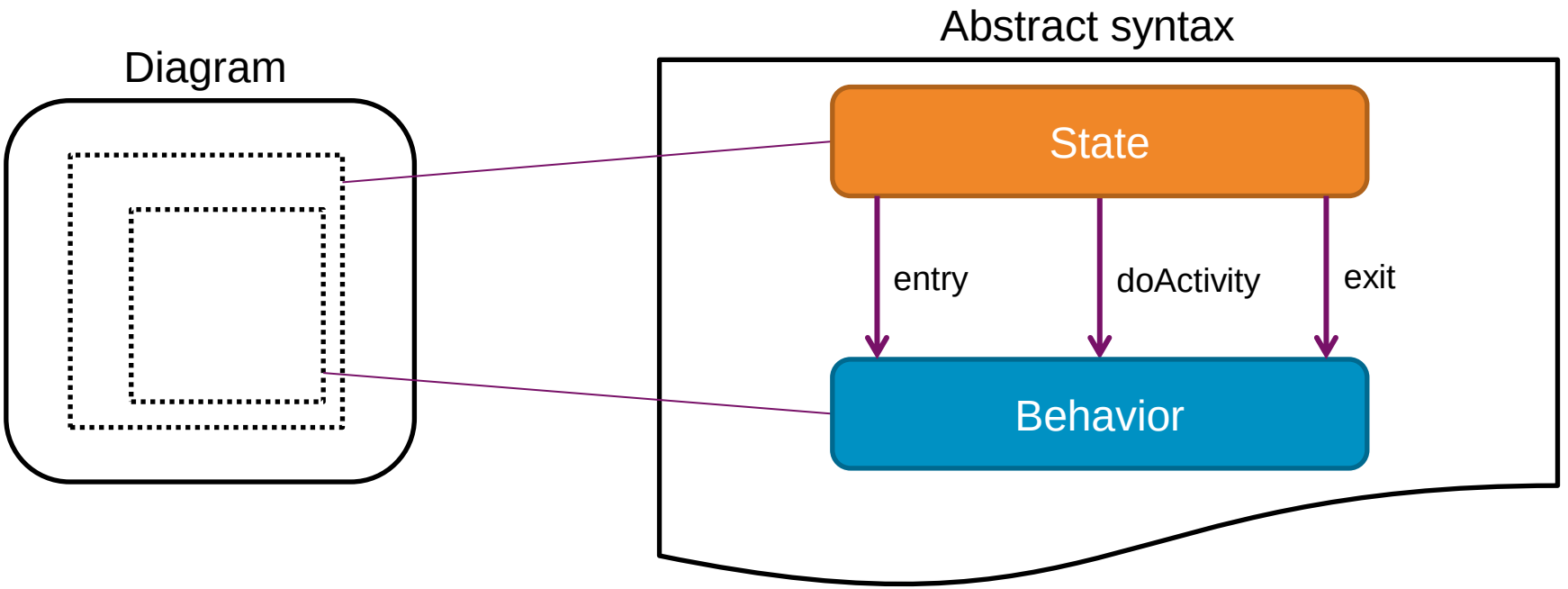
DOES DD ANSWER THE GMFGEN REQUIREMENTS ?

From that, can we automatically deduce connections between graphical relations and abstract syntax containments ?



DOES DD ANSWER THE GMFGEN REQUIREMENTS ?

From that, can we automatically deduce connections between graphical relations and abstract syntax containments ?



Development of an algorithm to analyze the whole UML meta-model.

→ 172 containments cases including 34 ambiguous.

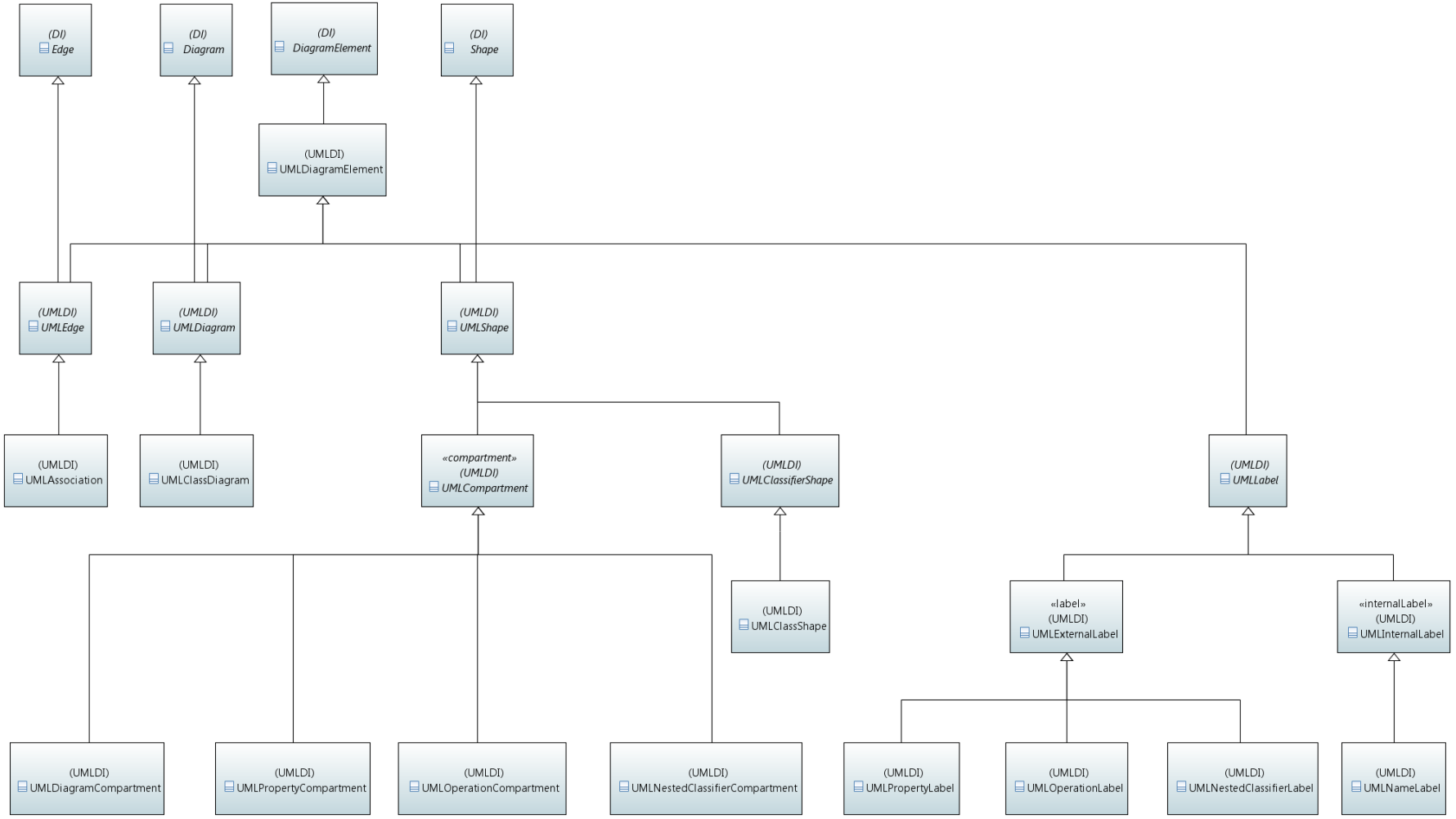
→ Considering the UML redefinition mechanism allows to solve only one case.

→ Language designer or user intervention is required.

THE SOLUTION PROPOSAL

UMLDI enriched

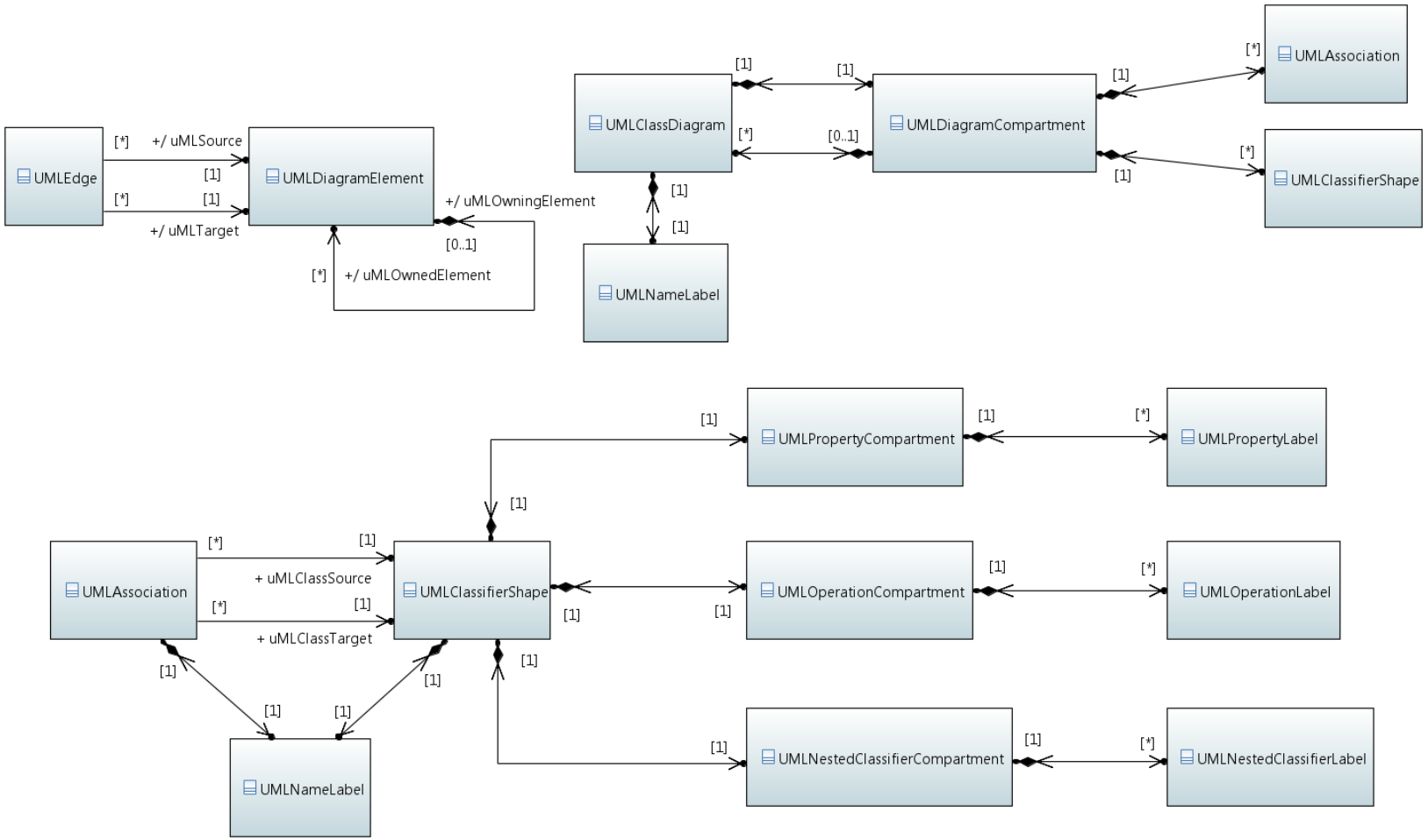
- Graphical element distinction



THE SOLUTION PROPOSAL

UMLDI enriched

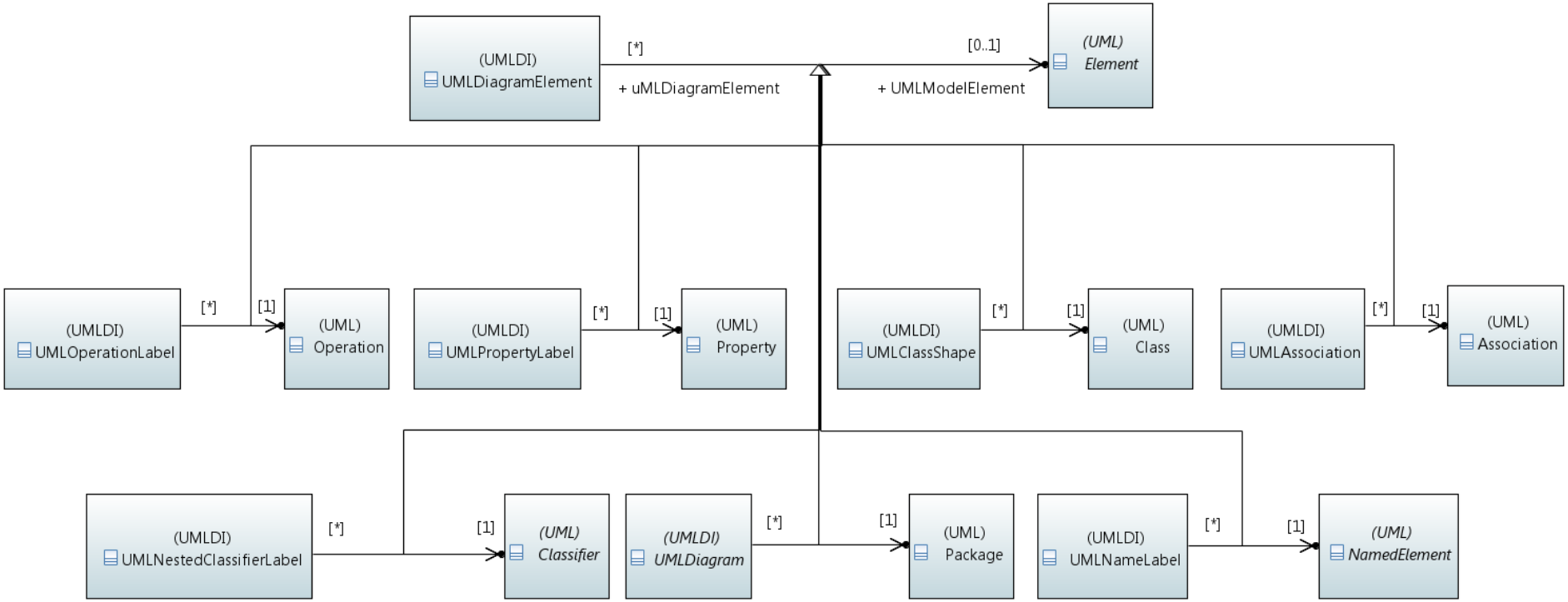
- Graphical element distinction
- Syntax enrichment



THE SOLUTION PROPOSAL

UMLDI enriched

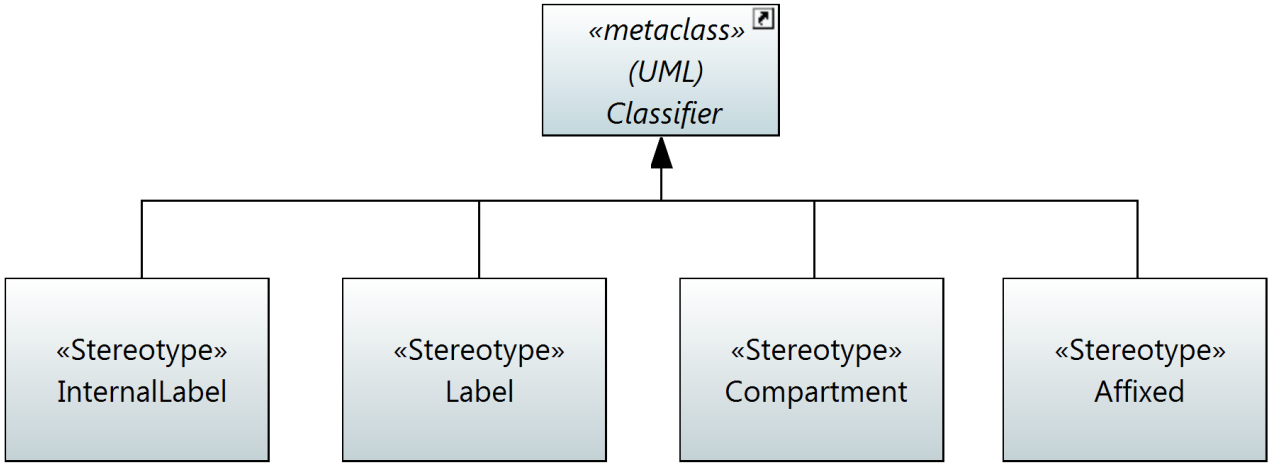
- Graphical element distinction
- Syntax enrichment
- Connections with abstract syntax elements



THE SOLUTION PROPOSAL

UMLDI enriched

- Graphical element distinction
- Syntax enrichment
- Connections with abstract syntax elements
- Graphical element's behavior distinction

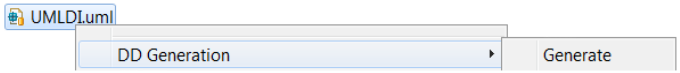


THE SOLUTION PROPOSAL

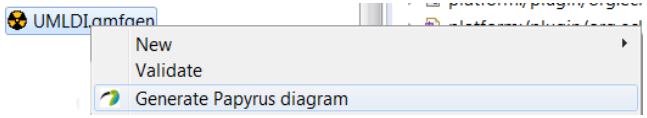
Tool generation process

Containments computation
+
M2M Transformation

Execution of the prototype on UMLDI

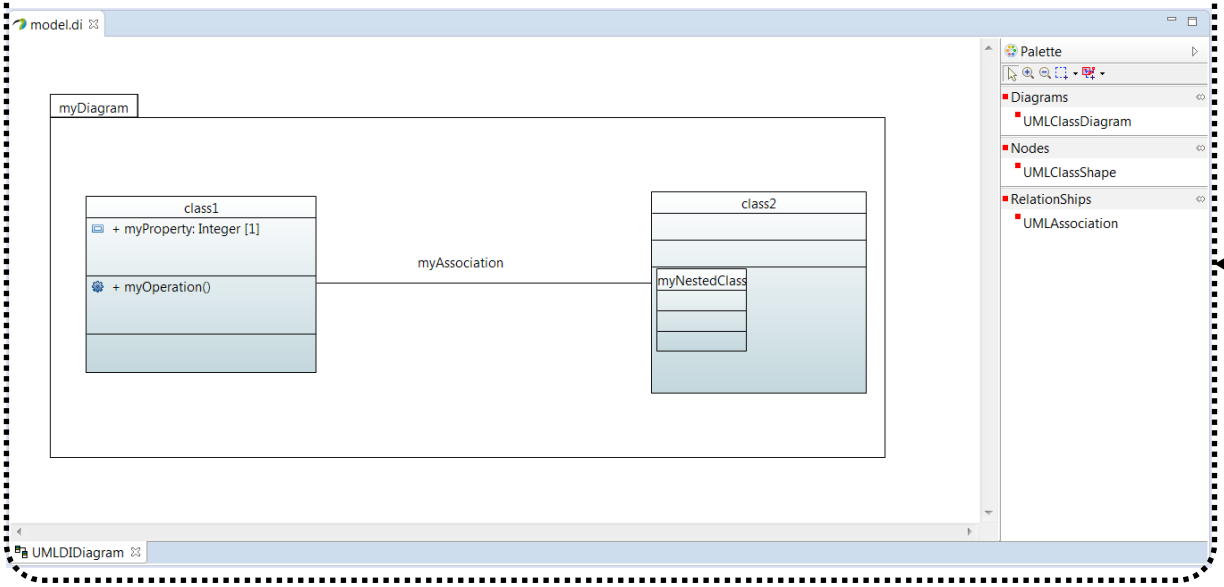


Execution of the GMFT M2T transformation



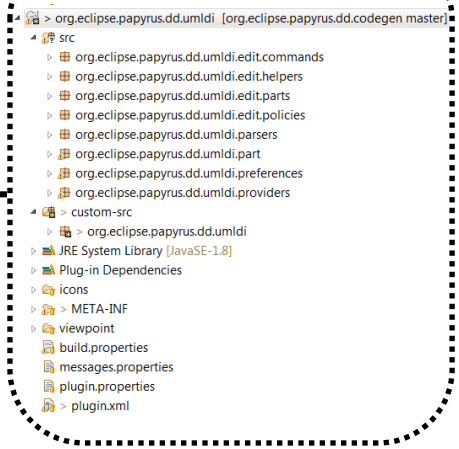
M2T Transformation

Diagram editor within Eclipse



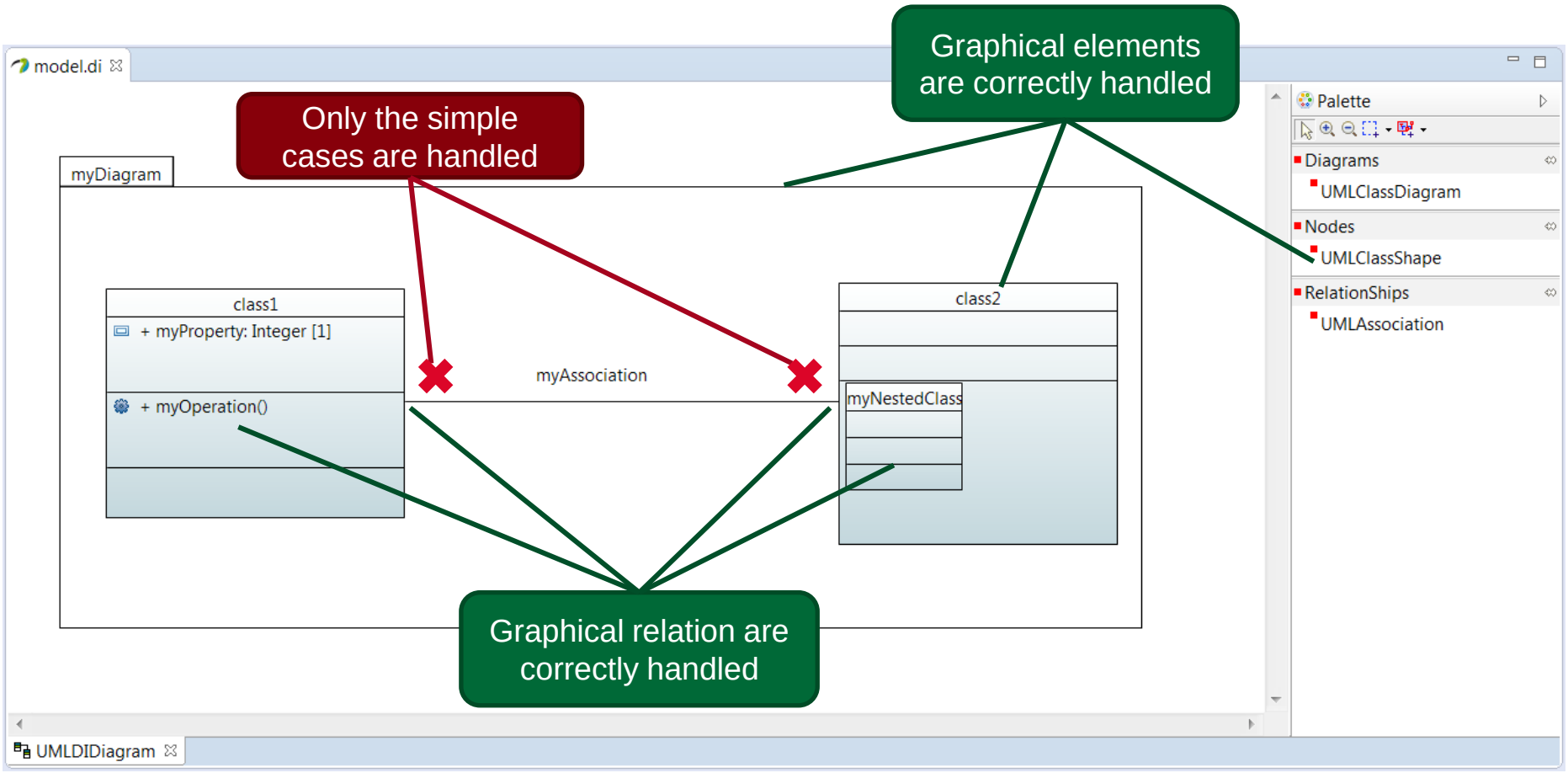
deployment

Eclipse plugin



THE SOLUTION PROPOSAL

Generated diagram editor



WHAT THE SOLUTION DO

- Improve our understanding of the DD standard : what we can do, what are the limitations.
- Proposal of a formalism to fully specify the graphical syntax of a language using the DD standard
- Development of a prototype to validate the proposal.

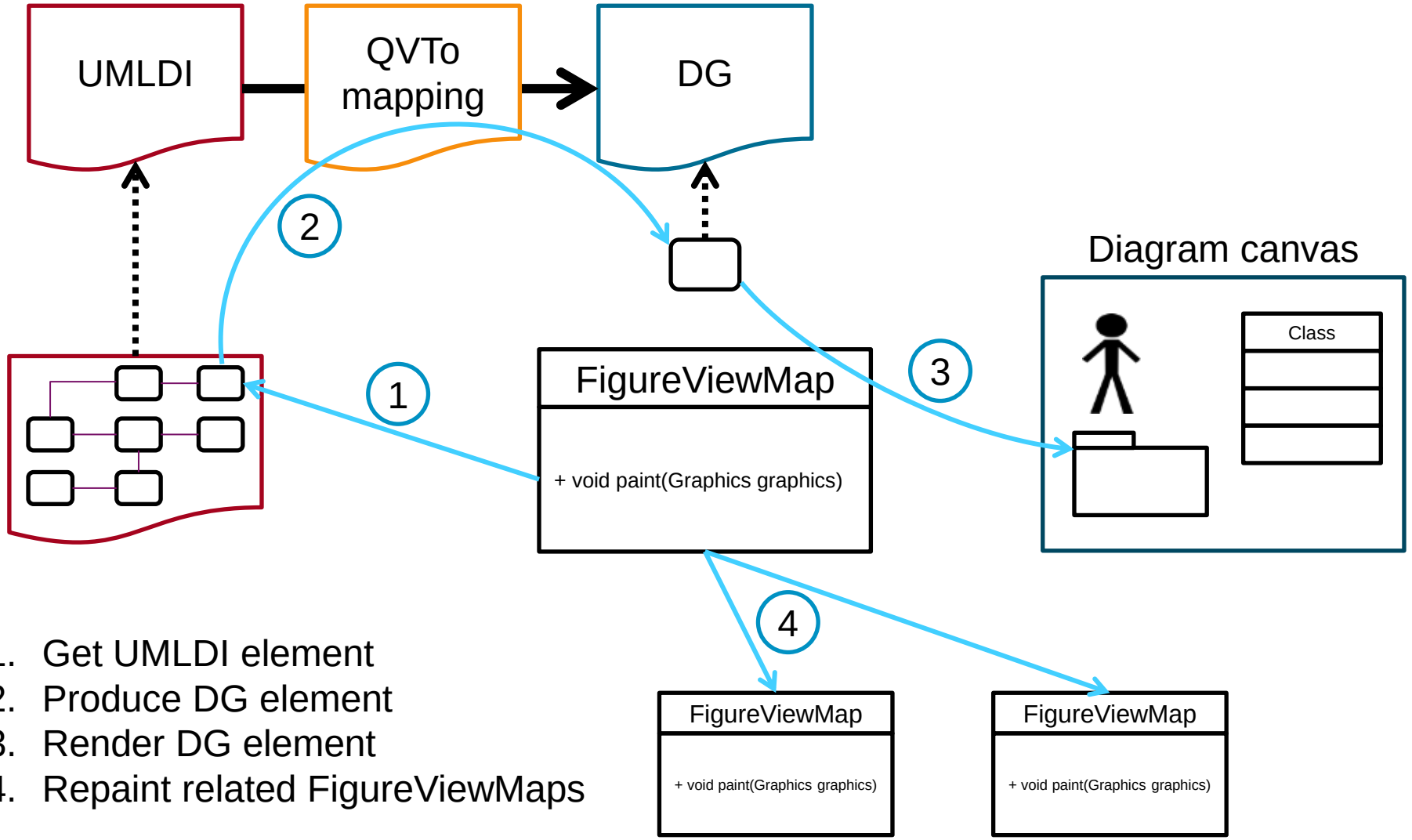
WHAT THE SOLUTION DO NOT DO

- Only the very simple cases are handled.
- The graphical vocabulary is not exploited.
- Ad-hoc graphical element’s behavior identification.

WHAT THE SOLUTION COULD DO

- Handle the graphical vocabulary using a modular QVTo DI→DG transformation.
- Handle n-m mapping to the abstract syntax using a framework for incremental transformation.
- Handle real label specification.
- Provide a more general and flexible mechanism for graphical element's behavior identification, or
- Provide a declarative framework to model UI interactions.

HANDLING THE GRAPHICAL VOCABULARY



1. Get UMLDI element
2. Produce DG element
3. Render DG element
4. Repaint related FigureViewMaps

instanceOf

Thank you for your attention

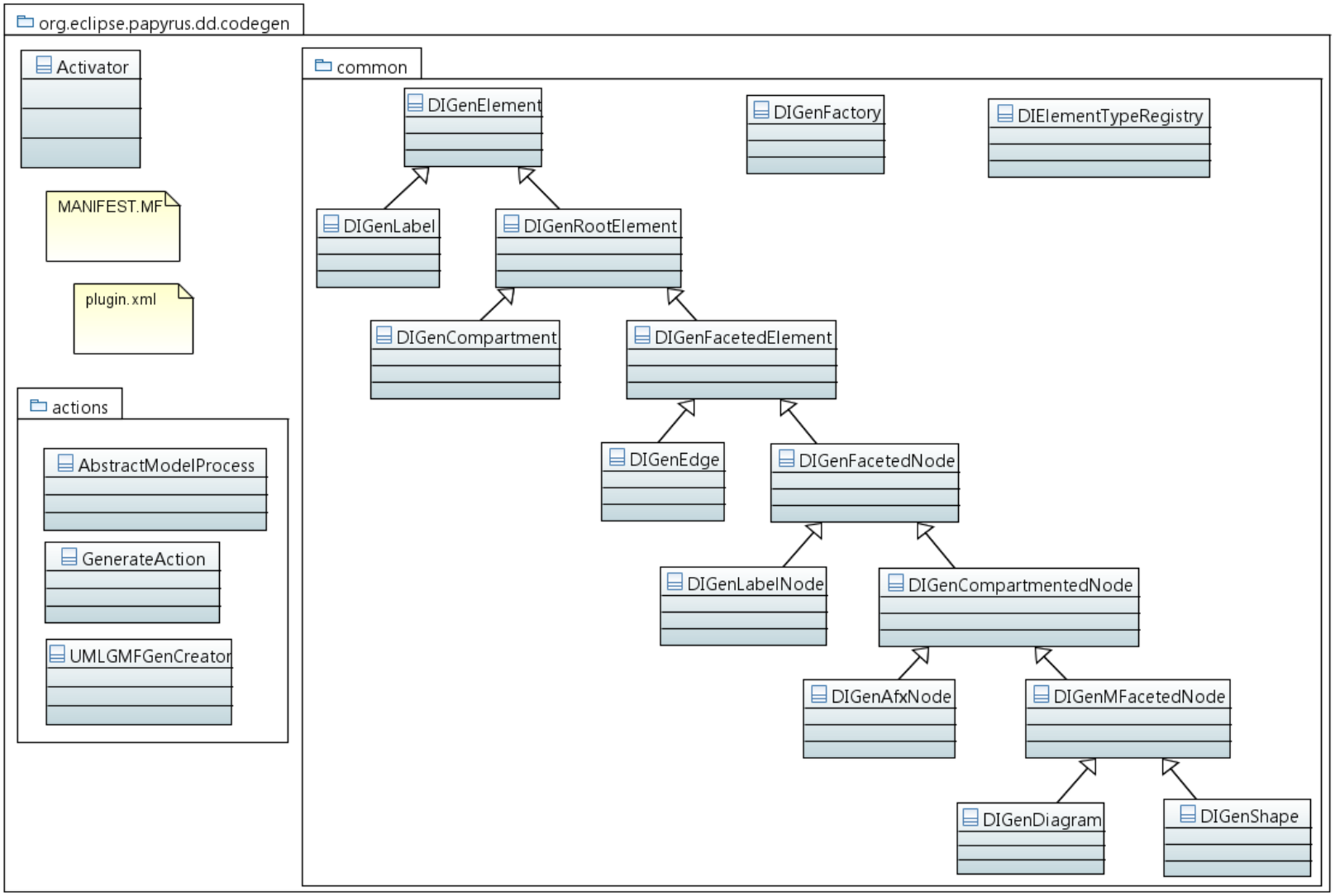
Any questions ?

Commissariat à l'énergie atomique et aux énergies alternatives
Centre de Saclay | 91191 Gif-sur-Yvette Cedex

DRT/LIST
DILS
LISE

Etablissement public à caractère industriel et commercial | R.C.S Paris B 775 685 019

PROTOTYPE OVERVIEW



PROTOTYPE OVERVIEW

Kind of element handled (UMLDI meta-model)	Java class	Kind of element generated (GMFGen model)
DI::Diagram	DIGenDiagram	GenMultiFacetedNode
DI::Shape	DiGenShape	GenMultiFacetedNode
DI::Shape « affixed »	DiGenAfxNode	GenAffixedNode
DI::Shape « compartment »	DiGenCompartment	GenCompartment
DI::Shape « label »	DIGenLabelNode	GenLabelNode
DI::Shape « internallabel »	DIGenLabel	GenLabel
DI::Edge	DIGenEdge	GenLink