A framework for Model Driven Production of Graphic Modeling Tools

Plan

- Problem
- Objective
- Proposal
- Experiments
- Comparisons
- Conclusion and perspectives

Plan

- Problem
- Objective
- Proposal
- Experiments
- Comparisons
- Conclusion and perspectives

Problem

- Modeling Environments offer the abstractions necessary to describe systems
- General-purpose abstractions → (use ++ / contribution to the development --)
- Domain-Specific Modeling (Adaptation to the domain needs)
  - The modeling environment must offer the concepts of the domain of application
  - Specific abstractions (use -- / contribution to the development ++)
- Many domains of applications
  - Generic environment + extension
  - Development of an environment for each domain of application

Plan

- Problem
- Objective
- Proposal
- Experiments
- Comparisons
- Conclusion and perspectives

Objective

- Support to the Domain-Specific Modeling
- provide an environment adapted to a field
  - Describe the needs
  - Produce automatically the environments from the needs
  - Independence between the need definition / technology of execution

MDE (Model Driven Engineering) approach

- MDE structure the application development in several models and model transformations
  - Good expression of problem (several concerns / integration)
  - How to go to the solution (several levels [business or technical] / iterative refinement)
  - Capitalization (specifications / know-how)
  - Evolution

Plan

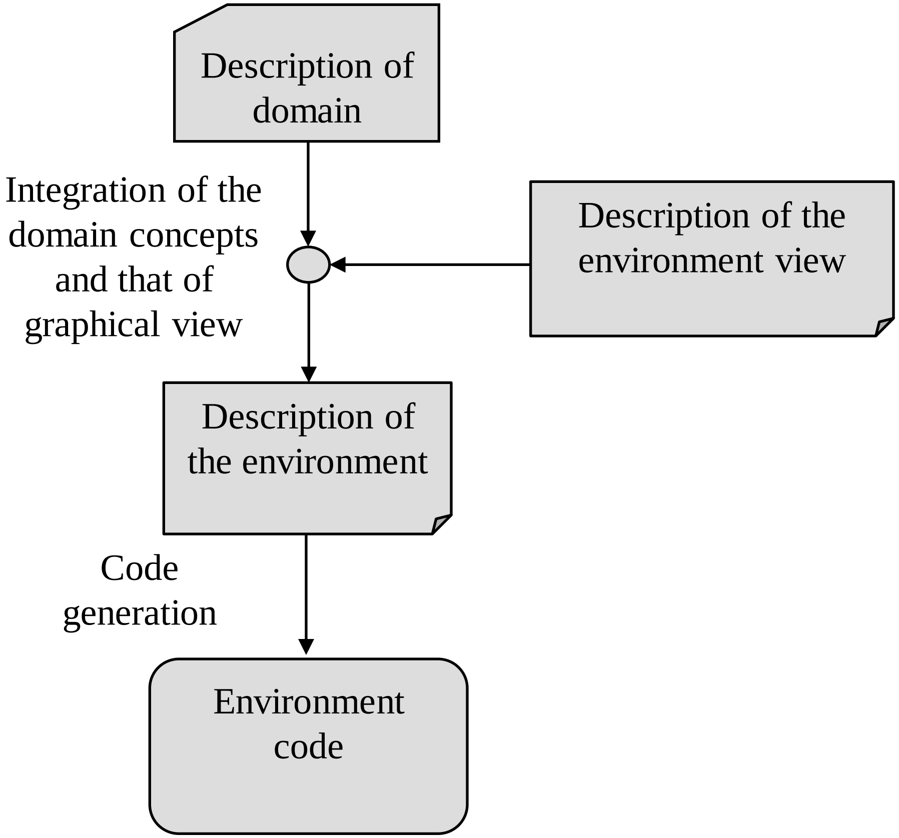
- Problem
- Objective
- Proposal
- Experiments
- Comparisons
- Conclusion and perspectives

Proposal

- Production of graphic modeling environments adapted to a domain of application
  - Graphic support to MOF repositories
  - Modeling of the graphic view
  - Automatic generation
- MDE Approach to produce these environments
  - Abstract Modeling of the environment needs
  - Separation between the needs (domain / graphical view)
  - Code production based on these definitions
    - Definition reusable
    - Use of a correspondence model between the domain model / the model of graphical view

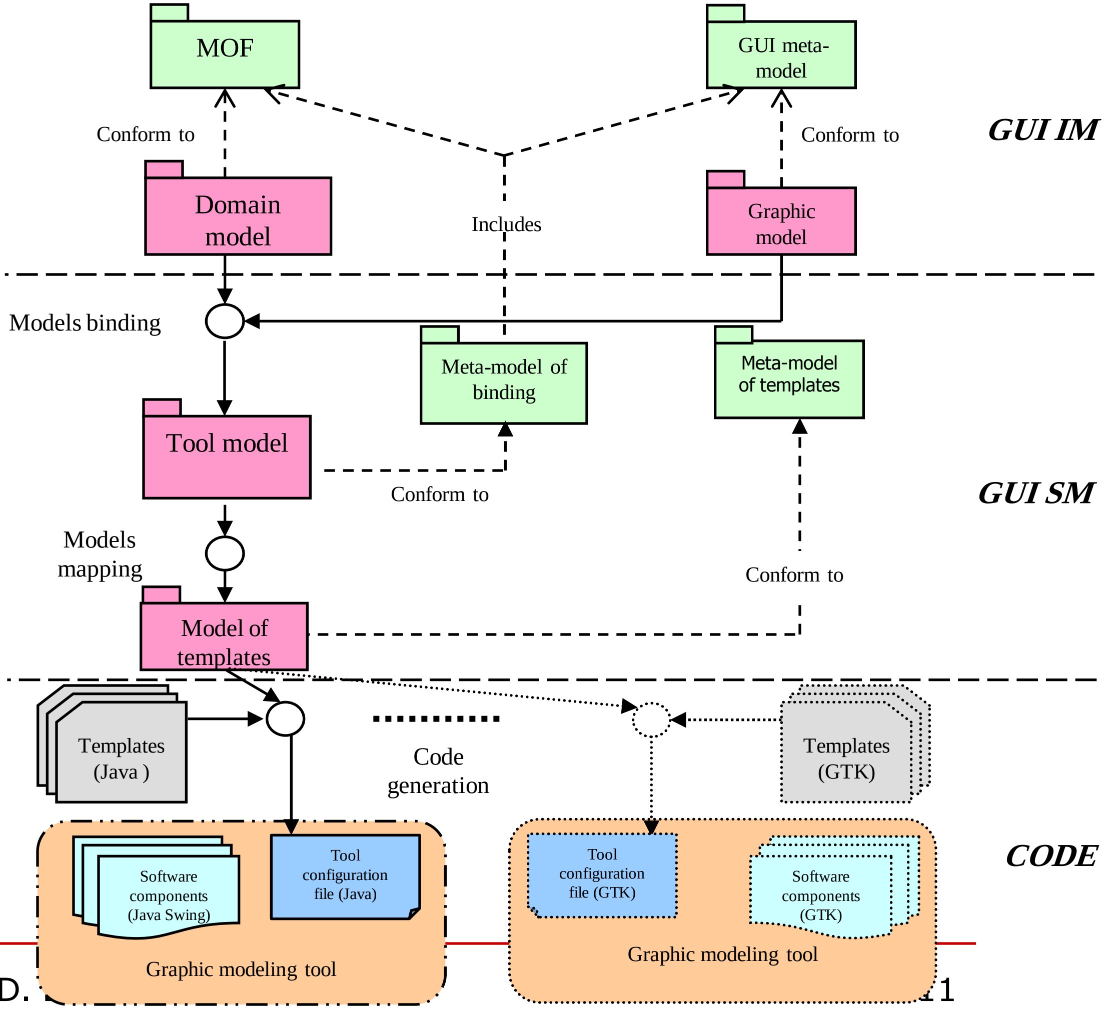
Approach

- Environment description independent of the execution technology
  - Description of domain
  - Description of view
  - Integration in the tool description
- Mapping into a particular technology


Description of domain
Integration of the domain concepts and that of graphical view
Description of the environment view
Description of the environment
Code generation
Environment code

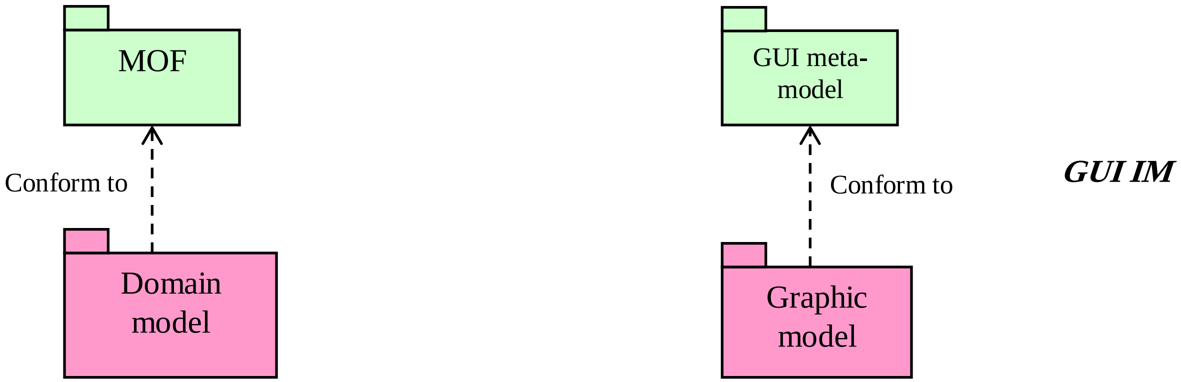
Framework

- Means to describe the needs
- Separation of needs
- Integration of specifications
- Model transformations
- Abstract Model of the tool
- Components reusable and templates
- Assembly of components on the basis of models



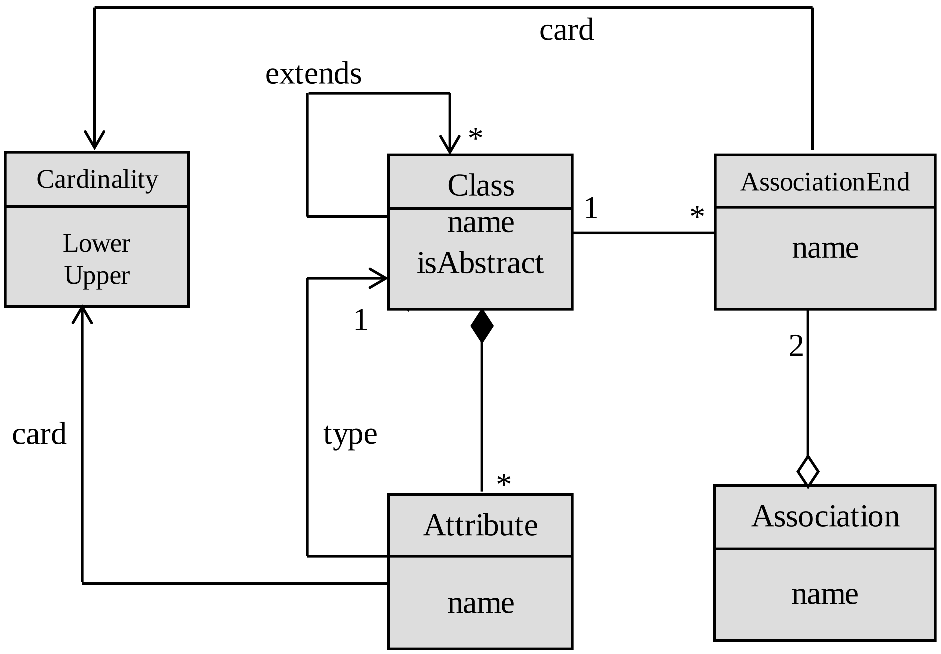
GUI IM

- Modeling of needs
  - Definition of the domain of applications
  - Definition of the graphical view



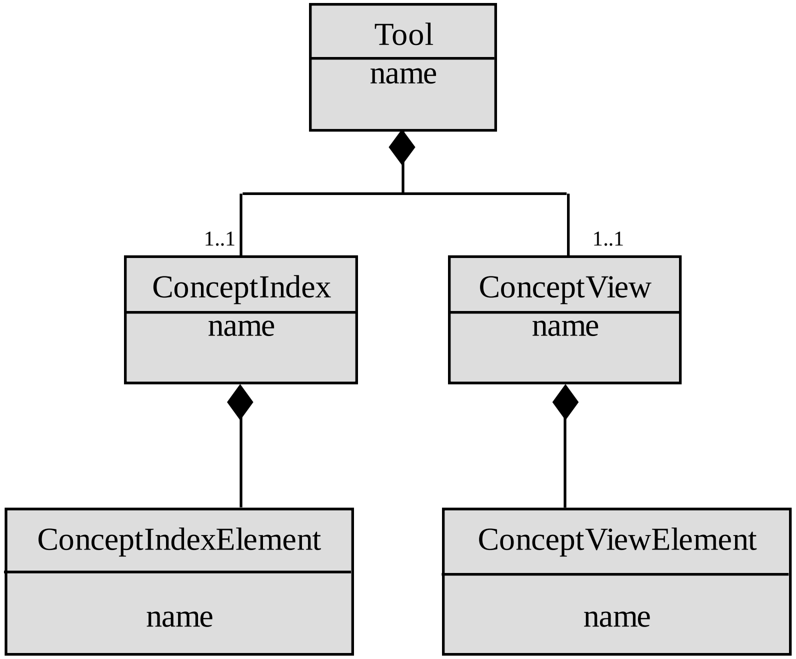
Definition of domain models

- M2 MOF define the concepts adapted to the needs of a domain of applications
  - Produced tools can use the repositories generated for M2 MOF
- lightMOF = un subset of MOF supported by our framework



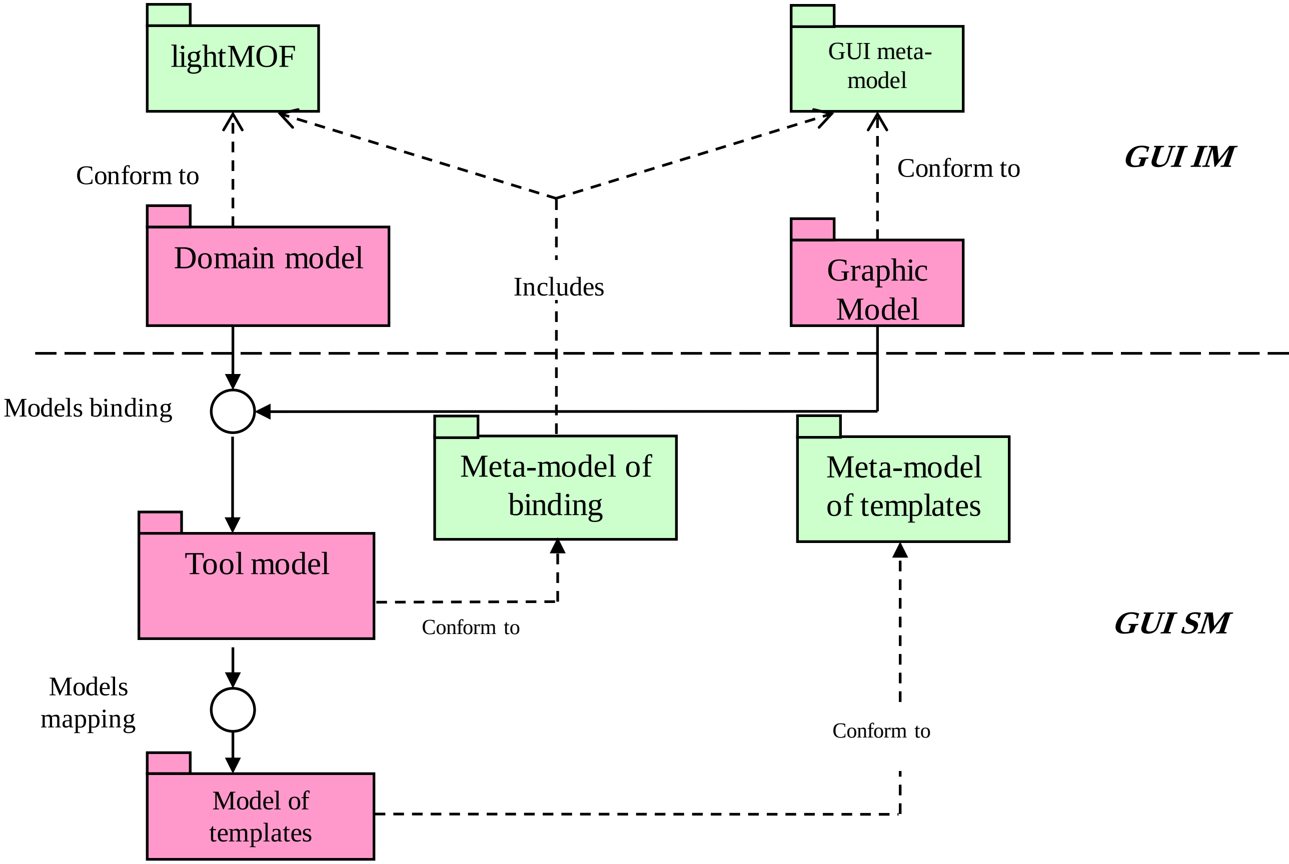
Definition of graphical views

- GUI M2 to capture the graphic interface of the modeling tool
- The concepts of GUI M2 can be implemented in several ways
- Graphic Model = choice between the implementations of the GUI M2 concepts

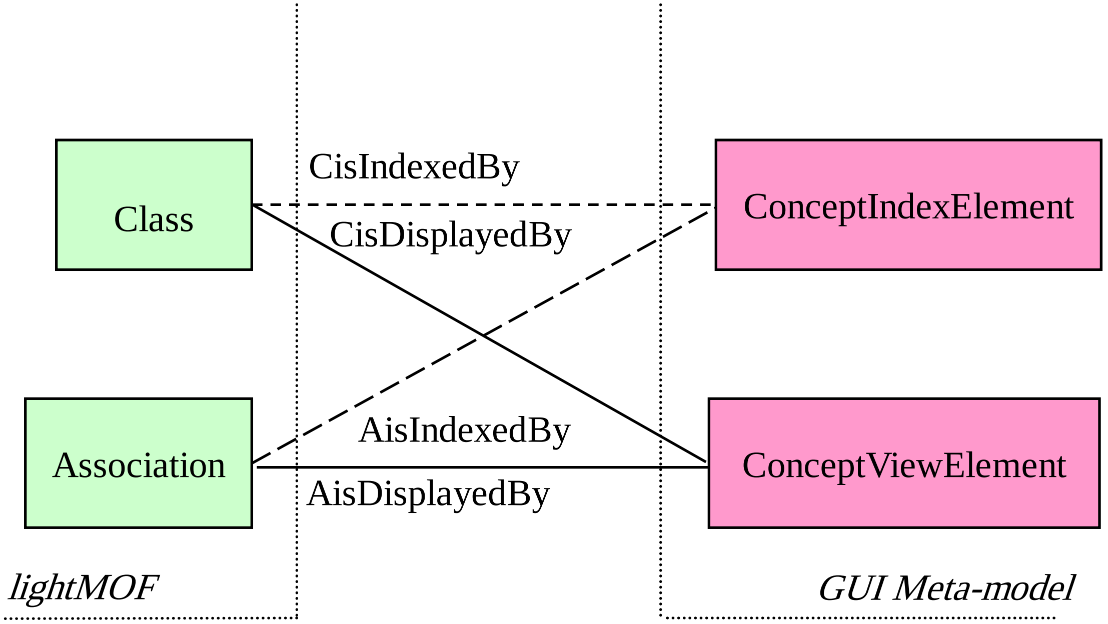


GUI SM (Integration of specifications)

- Binding the graphic models and the models of domains
- Mapping from the tool model to a model of templates



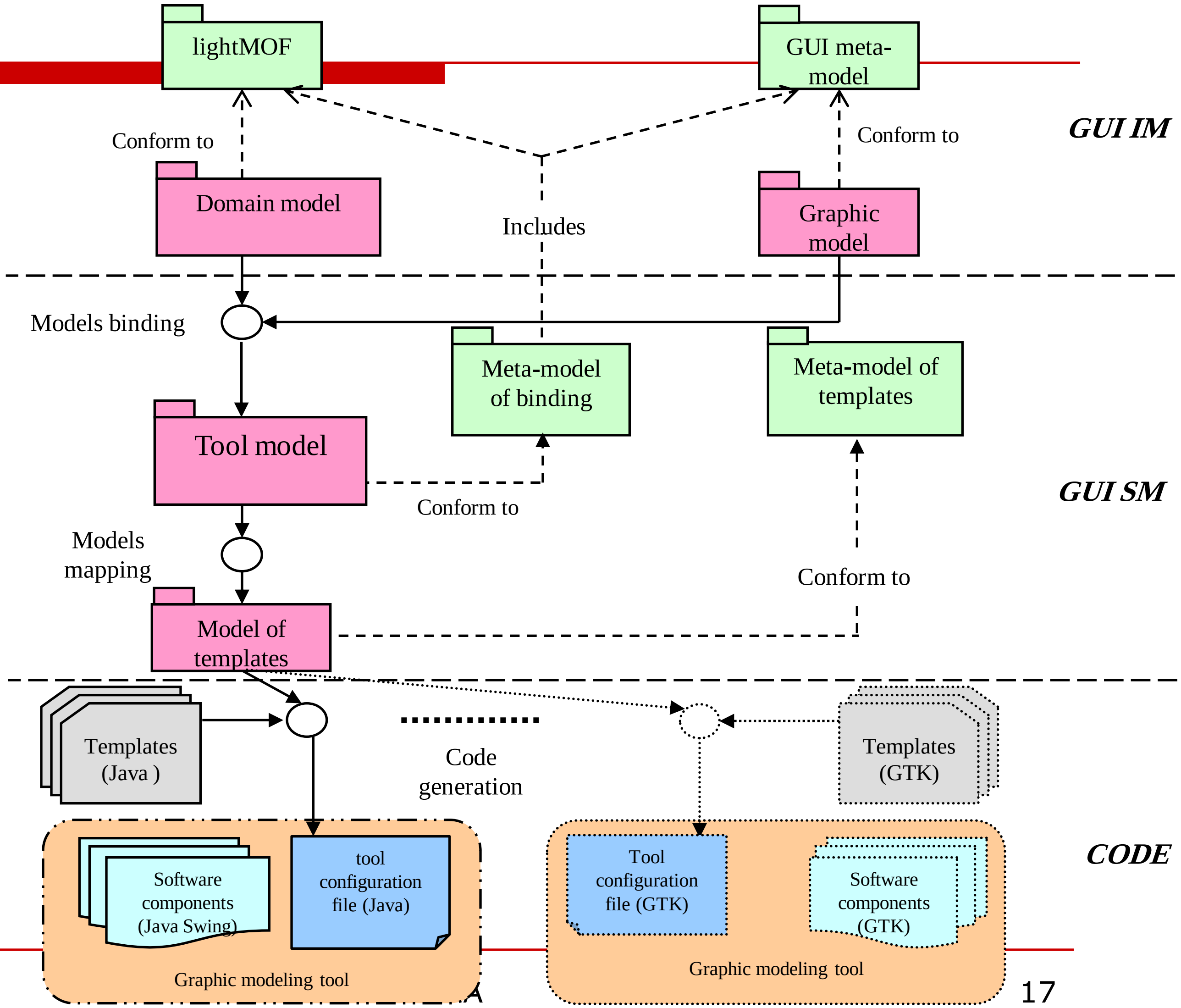
Models binding



- M2 of binding = lightMOF + GUI M2 + concepts of relations

CODE

- The different components of the tool
  - MOF repository
  - Graphic Interface



Plan

- Problem
- Objective
- Proposal
- Experiments
- Comparisons
- Conclusion and perspectives

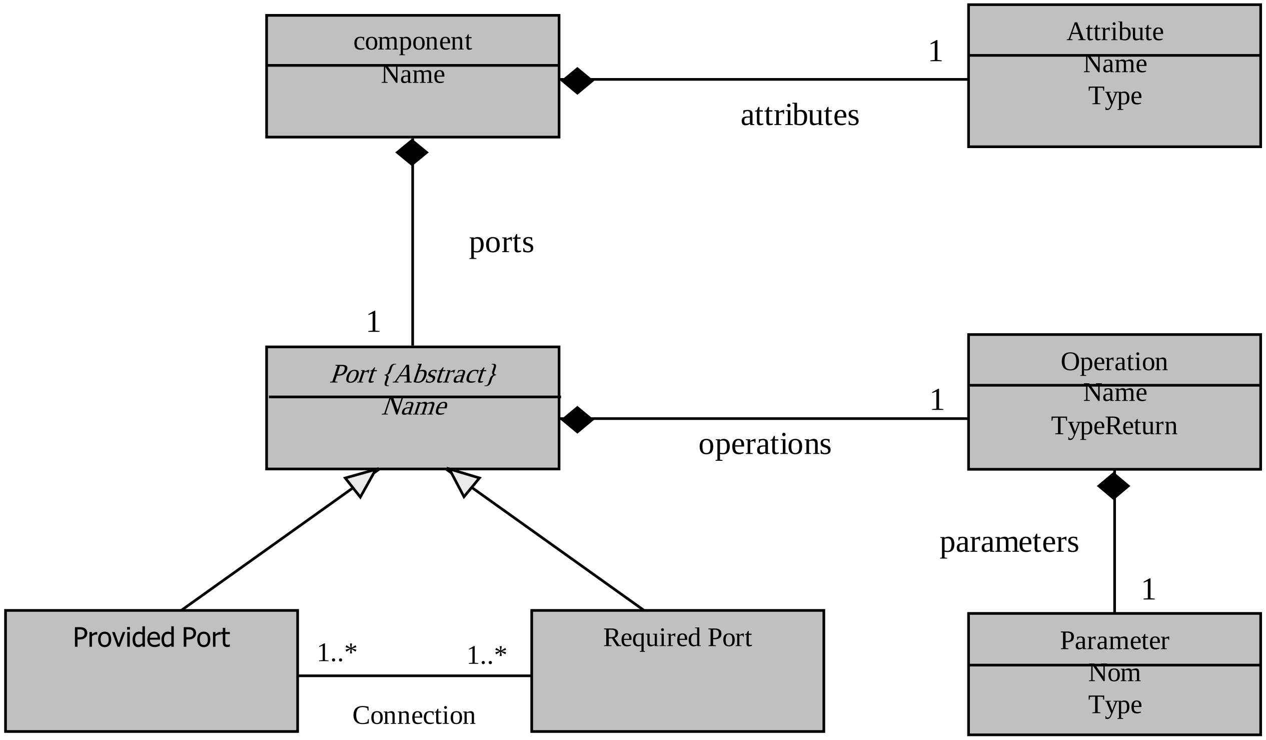
Experiments

- Production of graphic tools for modeling component-based applications
- Production of a graphical tool for our framework itself
  - Modeling of application domains

Experiment 1

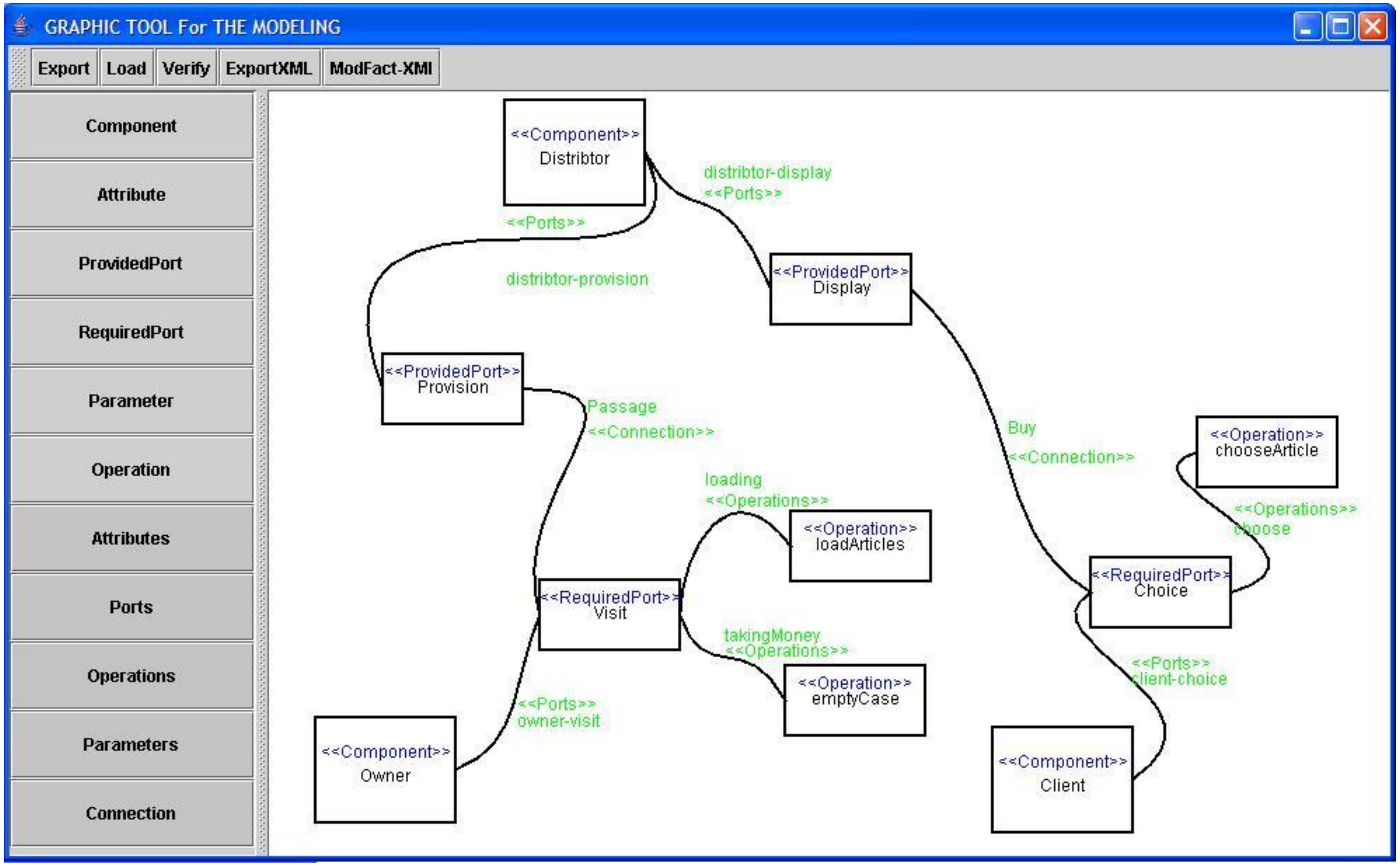
- Generation of several graphic tools for the same domain (component-based applications)
  - Define/Reuse the domain model of component-based applications
  - Define/Reuse a graphic model
  - Use the framework generators to produce the tools

The domain model of component-based applications



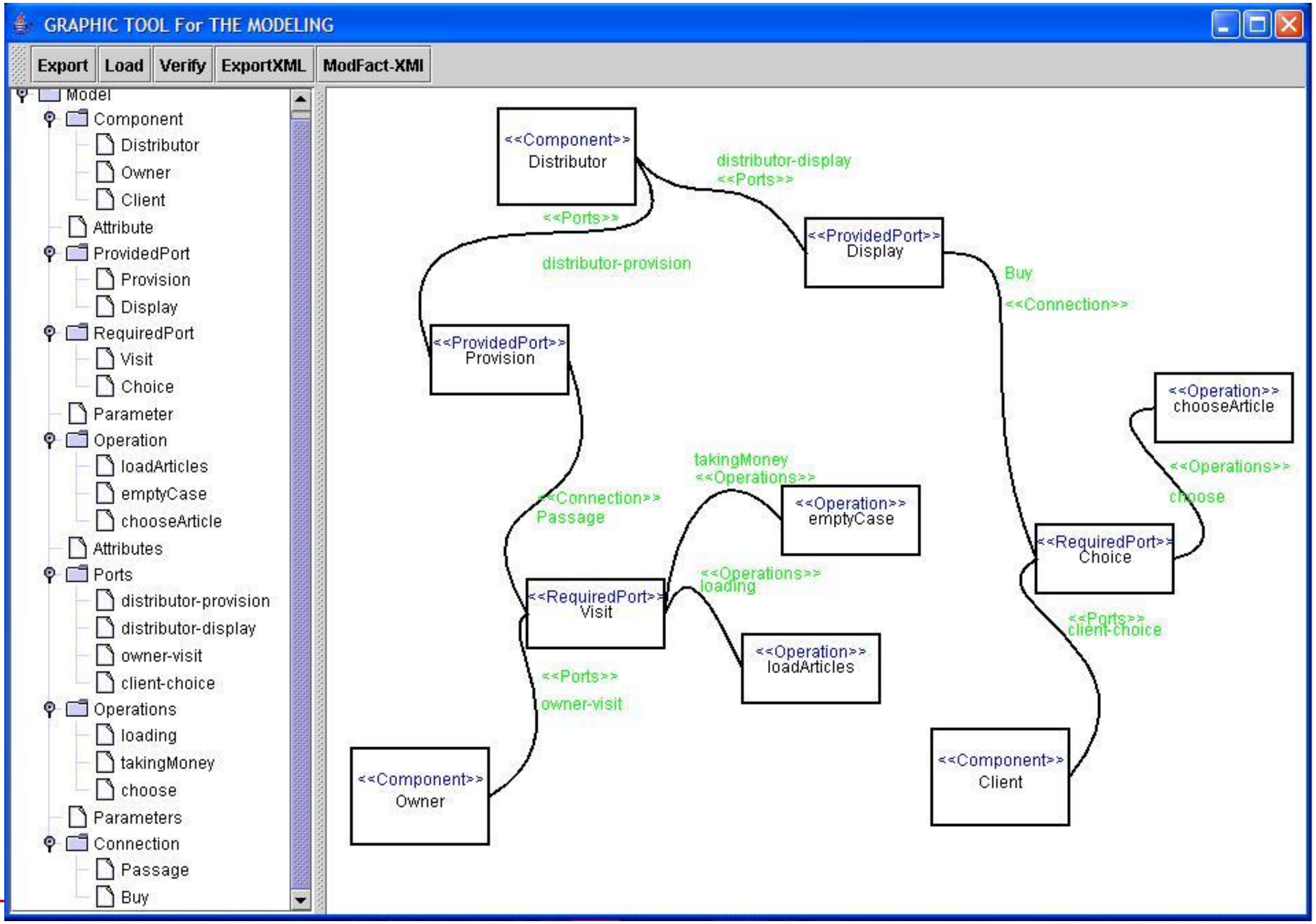
Graphic tool « 1 »

- Domain concepts → Buttons set
- Handling of model elements → Drawing board



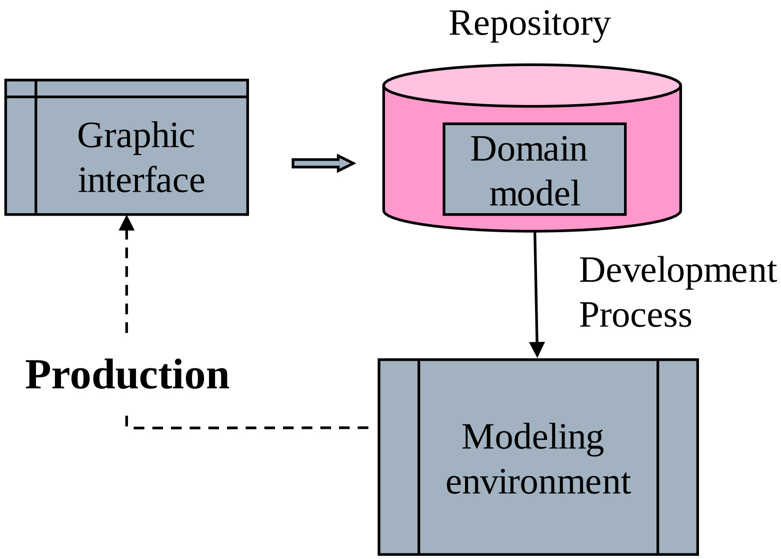
Graphic tool « 2 »

- Domain concepts → Tree
- Handling of model elements → Drawing Board



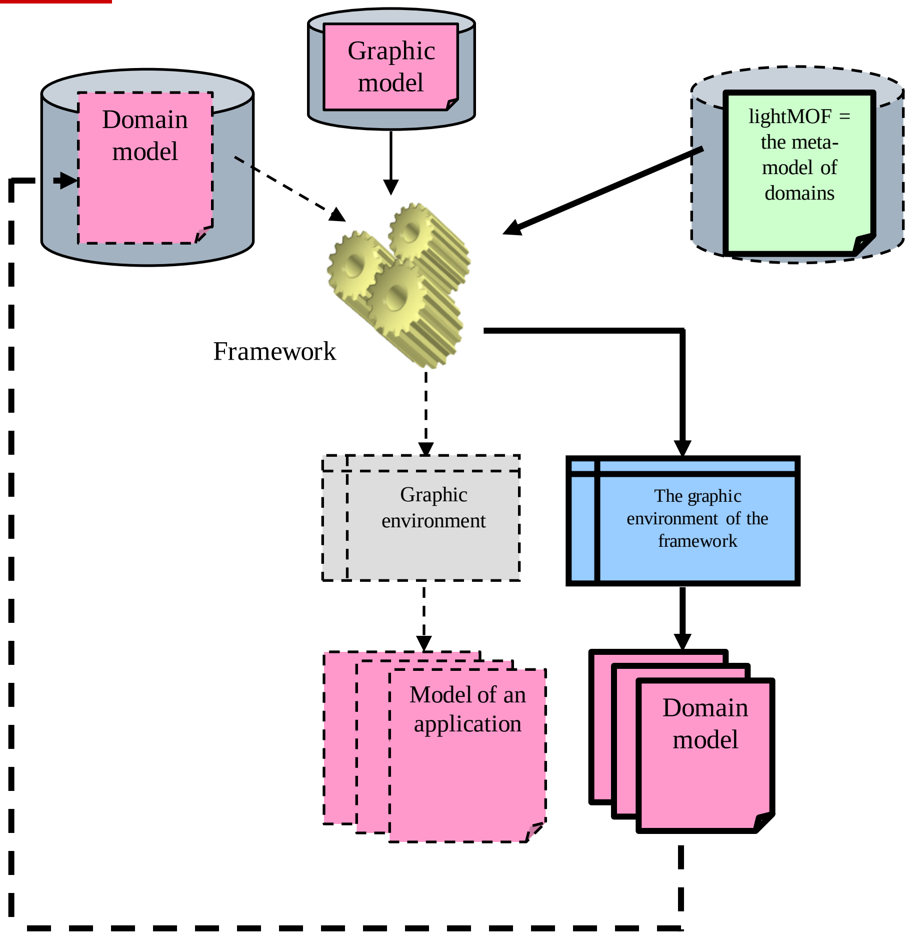
Experiment 2

- Generation of a graphic environment for modeling the domains of applications



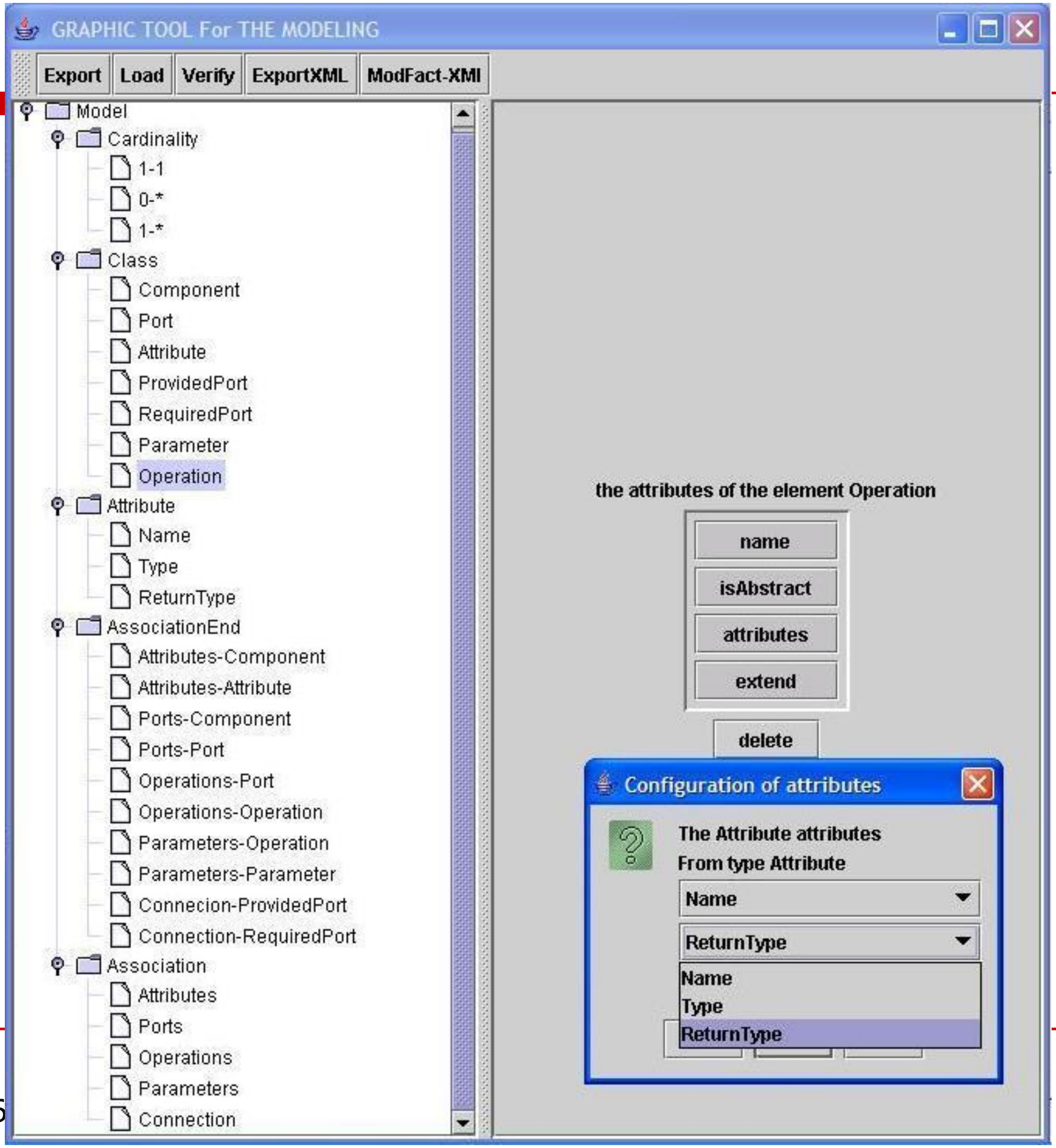
Production of the framework graphic tool

- LightMOF = the Meta-model of domains
  - We have directly used the repository produced for lightMOF
    - Write the Meta-model lightMOF as a model conform to lightMOF
  - Use the framework generateurs



Graphic tool for modeling of application domains

- lightMOF concepts → Tree
- Handling of model elements → Form set



the attributes of the element Operation

Configuration of attributes

The Attribute attributes
From type Attribute

Name

ReturnType

Plan

- Problem
- Objective
- Proposal
- Experiments
- Comparisons
- Conclusion and perspectives

Conclusion

- Domain-Specific Modeling / Many domains of applications
- Production of graphic environments for the domain-specific modeling / MDE approach
  - Modeling of environments
    - Abstract Models ➔ Capitalization for other technologies
  - Separation of concerns
    - Reuse of definitions
  - Framework + Tooling ➔ Automatic Production

Perspectives

- Extension of GUI meta-model
  - Capitalization more important
- Extension of the conformity with OCL
  - Transfer the checks of the development process to modeling
- Support to the simulation (new concern) → validation of the behavior
  - Model of actions / model of selection
  - binding the domain concepts / the actions → Production of interaction interface with the repository
- Production of Co-design tools → Definition of models / Collaboration
- Models binding (dynamic)

Thanks