Project Presentation

Synchronization between display
objects and representation templates
in graphical language construction

François Helg
Fabien Rohrer

Assistant :
Frédéric Fondement

Overview

- François
  - Aim of overall project (**SVG-Based Modeling Tools**)
  - Some definitions concerning languages construction
  - Problematic
- Fabien
  - Resolution
  - To do

Aim of the overall project

- This project concentrates on defining a graphical concrete syntax for a language, if the abstract syntax is given.

  - Part of TopModL initiative
    - TopModL is an international open-source initiative launched to provide an extensible framework for model-driven experimentation.
    - The goal of the TopModL initiative is to contribute to the advance of model-driven technologies by promoting an extensible open-source framework fully compatible with the latest OMG standards and recommendations.

ProBXS

- Project of Fabrice Hong (Semester Project in Winter 2004)

- Concrete syntax graphical edition tool

- « SVG image becomes an editing tool »
  - Display representation based on SVG templates
  - DopiDom extension (dynamical behavior)

What do we need then?

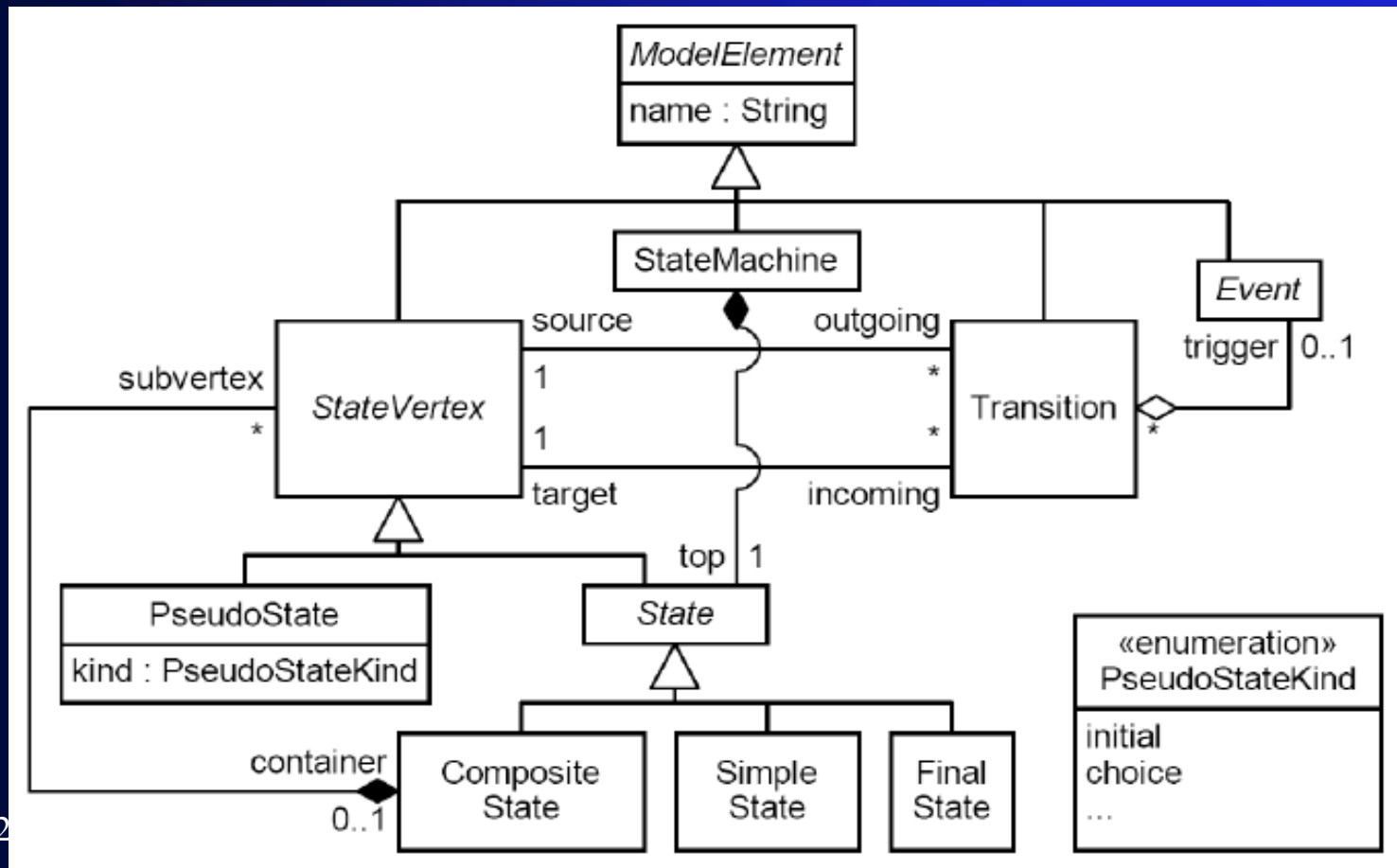- ProBXS provides an editing tool which works well, but…

SYNCHRONIZATION!

With model and metaModel

How to define a language?

- Definition of a language
  - Abstract syntax
    - metaModel
    - Capture concepts of the language
  - Concrete syntax
    - SVG templates
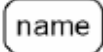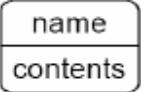   - Capture display representation of the language

Example (statechart language)

- Abstract Syntax

Example (statechart language)

- Concrete Syntax (Templates)

Transition	SimpleState	Composite State	FinalState	PseudoState (initial)	PseudoState (choice)
–event⟫	name	name / contents	◉	●	○

- Actually, often made in an informal way. If formal, often too complicated or too restrictive.

Example (statechart language)

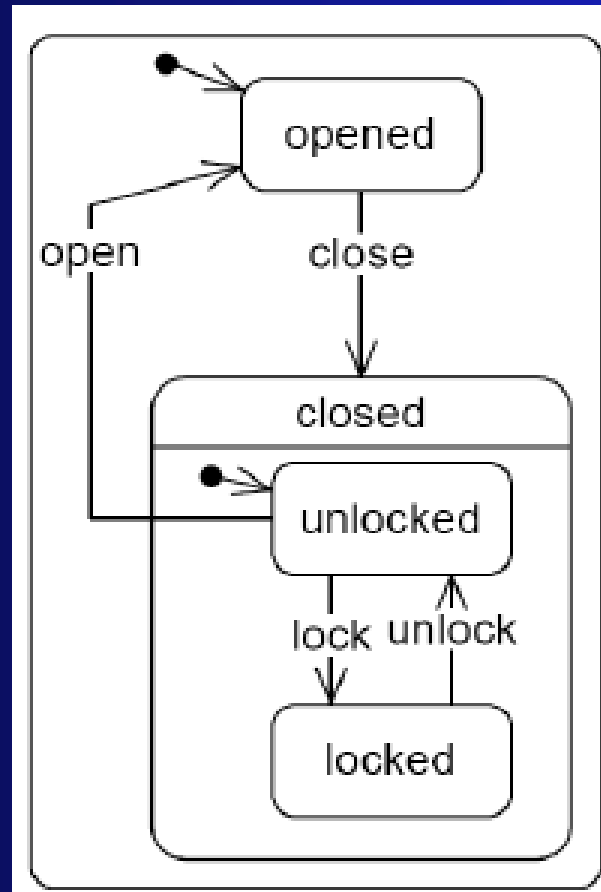- raw instance of the metamodel (model)

(a) Instantiation of metamodel
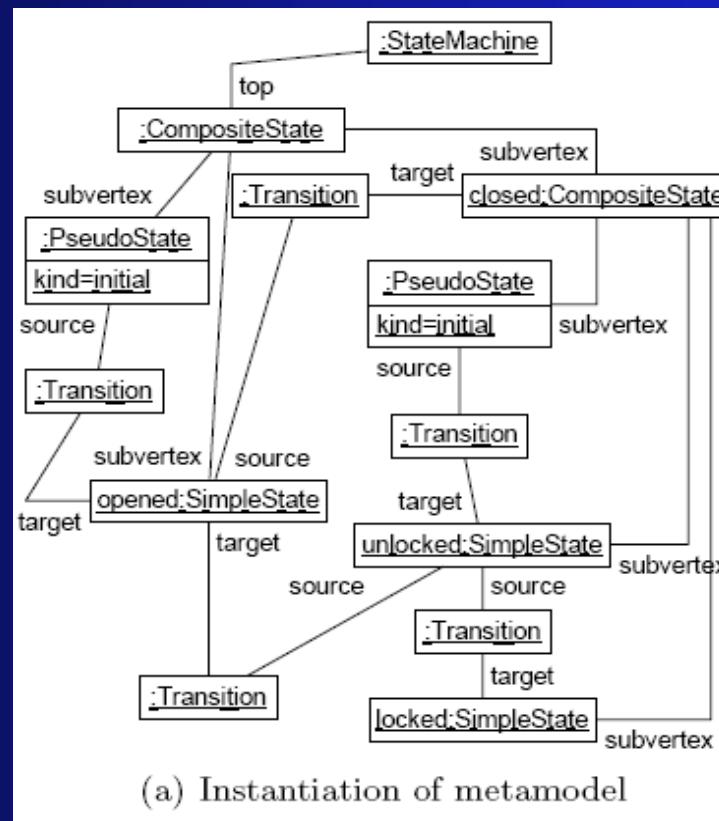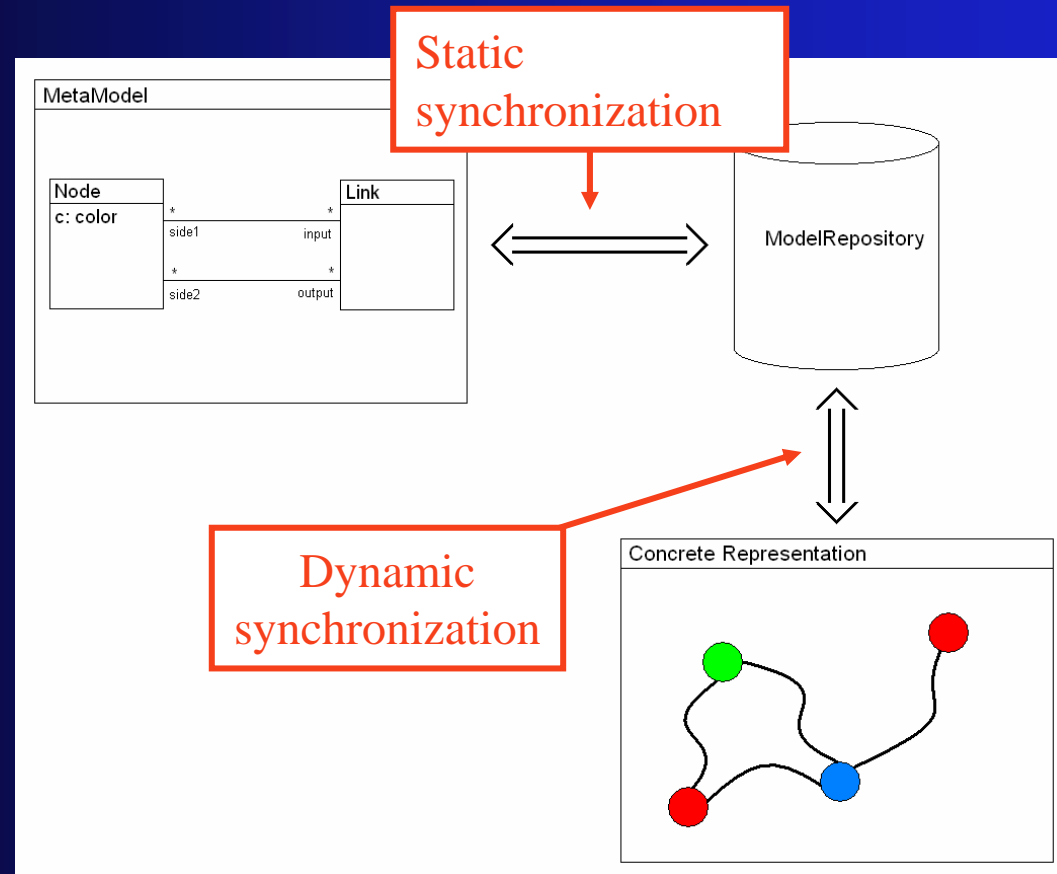
René Magrit

- This is not a pipe

Example (statechart language)
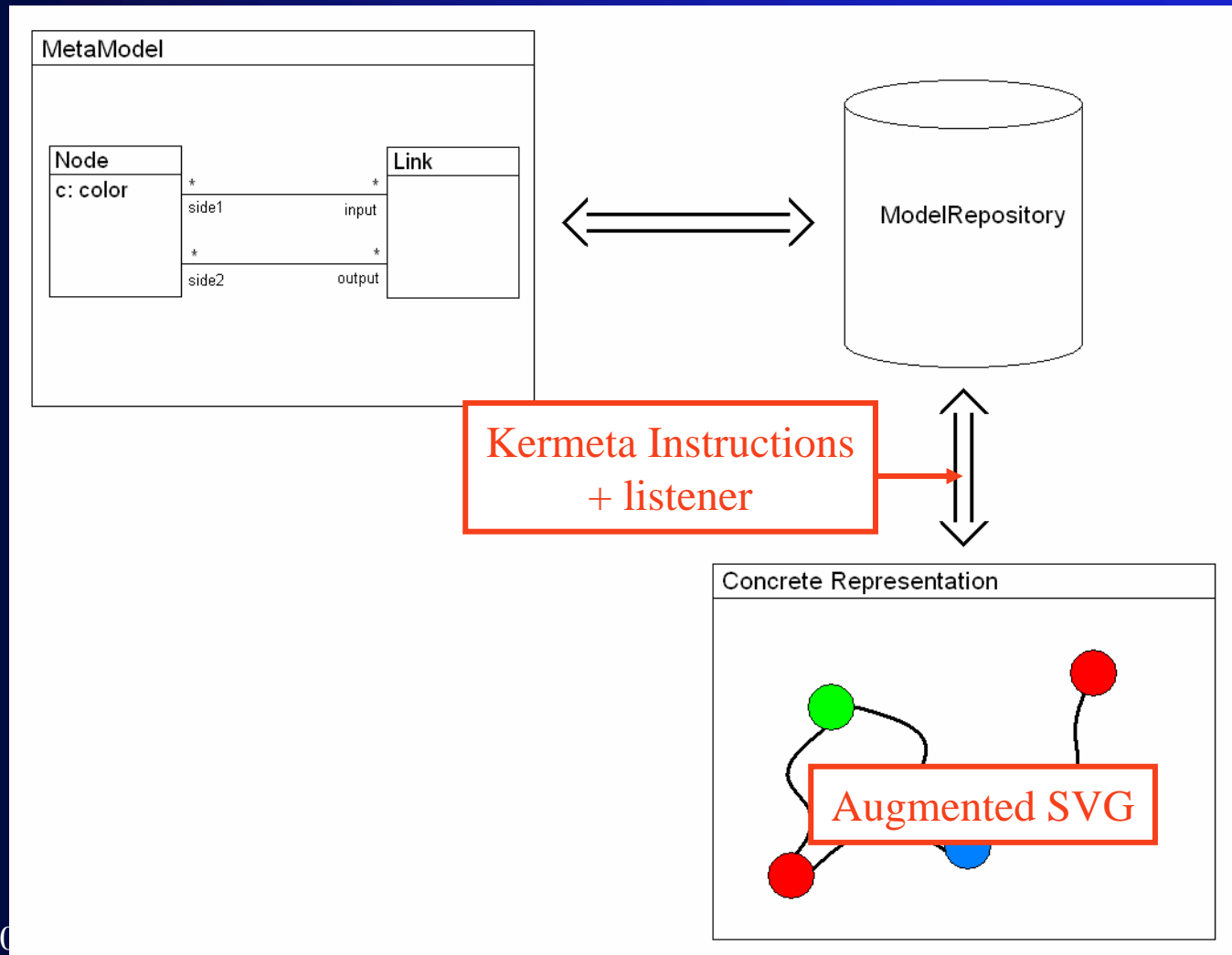
- This is not a model!!!

Problematic

- Need to keep synchronized display objects and representation templates
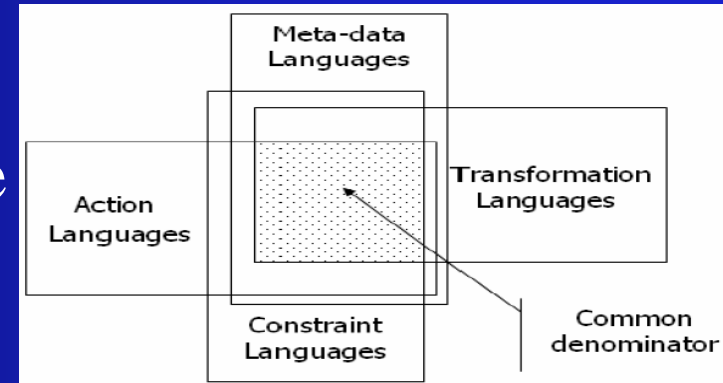
Implementation (our solution)

Augmented SVG

- SVG +…
  - Dynamical features (Project of Fabrice)
  - Some OCL's queries for descendent synch.
    - + listener on the repository
  - KerMeta's method calls for rising synch.
- Exemple :
  - **<c:content query=''{OCL | self.name.text}"
    update(s:String)= ''{KerMeta |
    self.name.text := s}''/>**

KerMeta

- ## What is KerMeta?
  - ### Metamodelling language
  - ### Defines
    - #### Structure of metamodel
    - #### Behavior of metamodel
- ## Why Kermeta?
  - ### Language open and adapted
  - ### Hide the aspects of the ModelRepository
  - ### Powerful

Implementation (our solution)
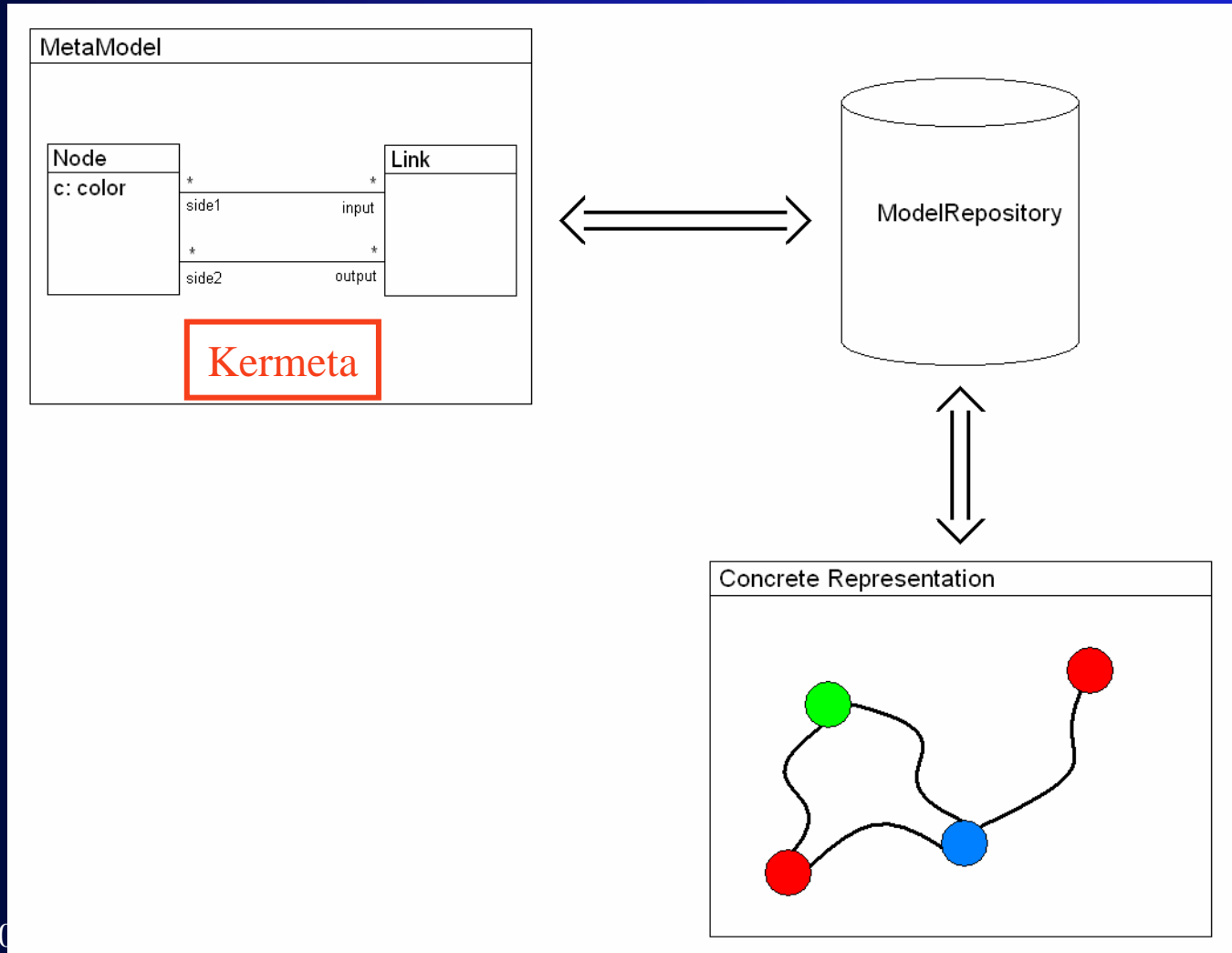
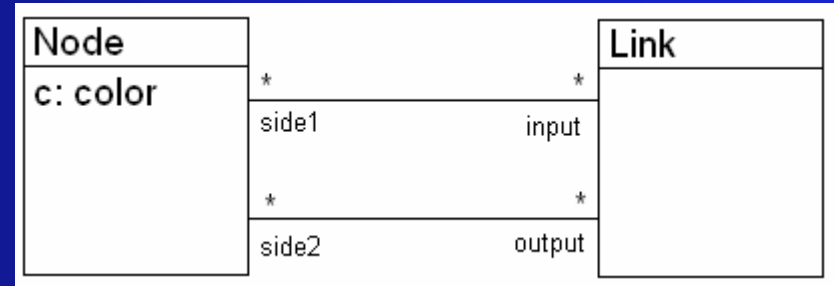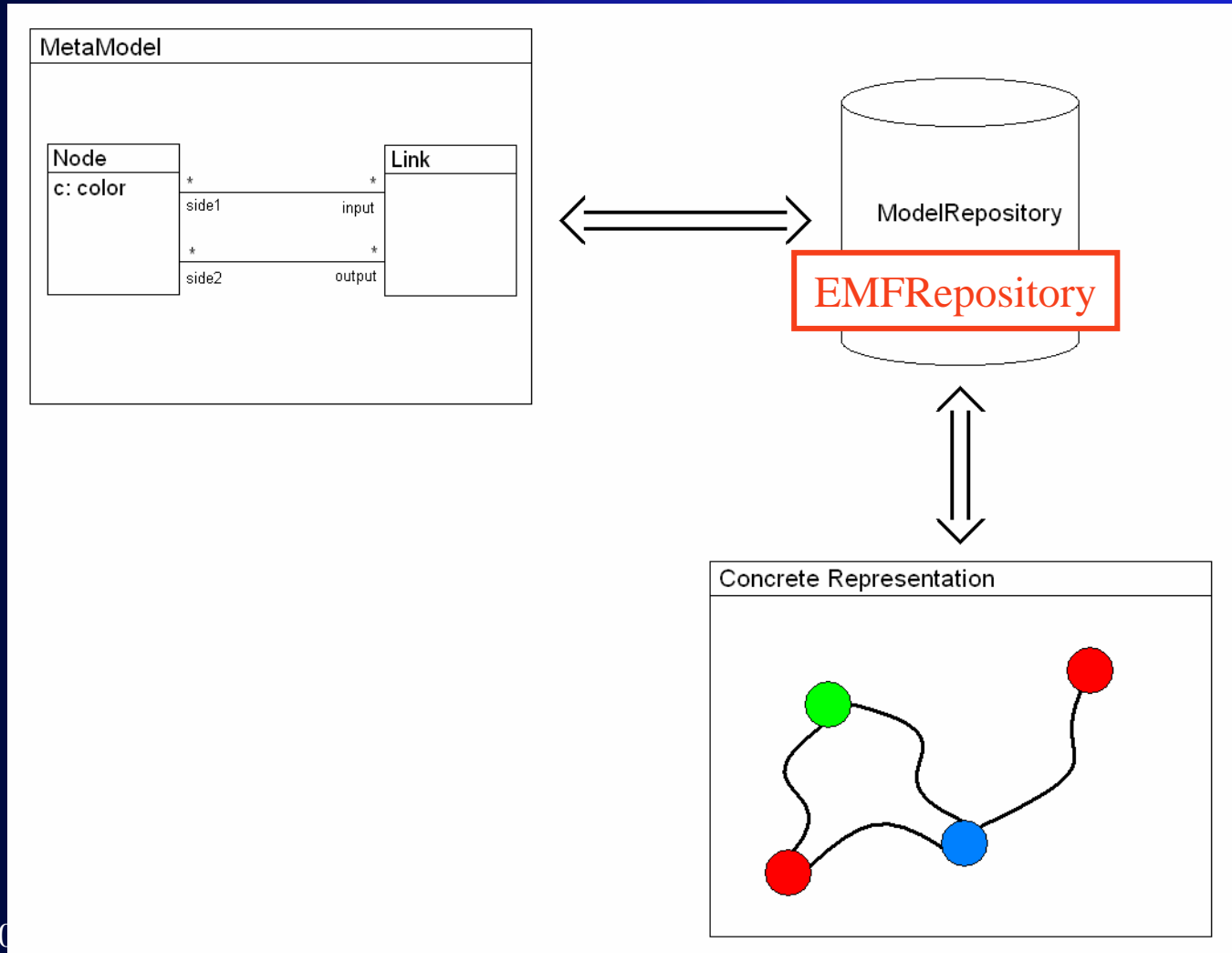# Exemple of a metaModel in Kermeta

require kermeta
using kermeta::standard

class Node
{

    atribute c : color
    reference input : set Link[0..*]#side1
    reference output : set Link[0..*]#side2
}

class Link
{

    reference side1 : set Node[0..*]#input
    reference side2 : set Node[0..*]#output
}

Implementation (our solution)

EMFRepository

- Inspired from the resource manager of EMF models
- Repository who can contain a set of resources (EMFRessources)
- Each ressources can contain an instance

- In our project:
  - Contains a kind of « instance » of the metaModel (root ~ main class)
  - Store Kermta's objects
  - Don't care because it's the reponsability of the Kermeta's interpreter

Pros & cons

- Choosen languages are specific for these tasks
- Don't care about the modelRepository

- Depend to the platforms and the technologie
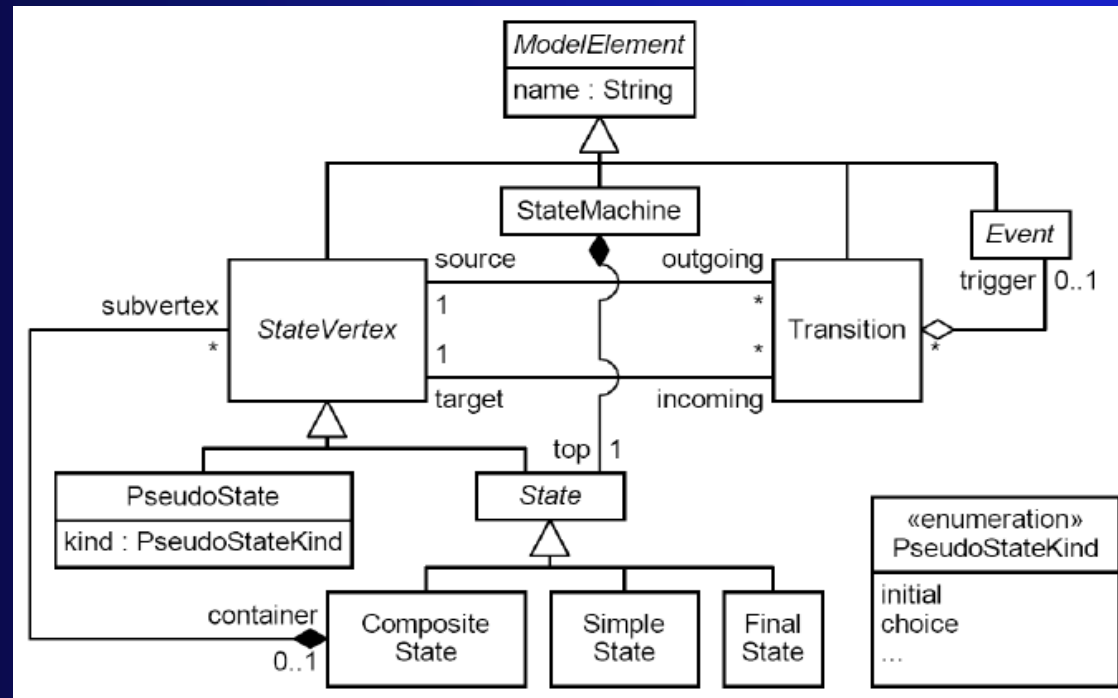- Multi-language & Multi-Repository

Alternative

- Use other languages
- Use an other architecture

- We still have to do more research on that!

TO DO

- Put KerMeta instructions into SVG templates

- Implement bidirectional links between SVG & ModelRepository

- Implement a way to do Multi-Language and Multi-Repositary

Demo of ProBXS

Time for questions!

?