

## Definition of an eXecutable SPEM2.0

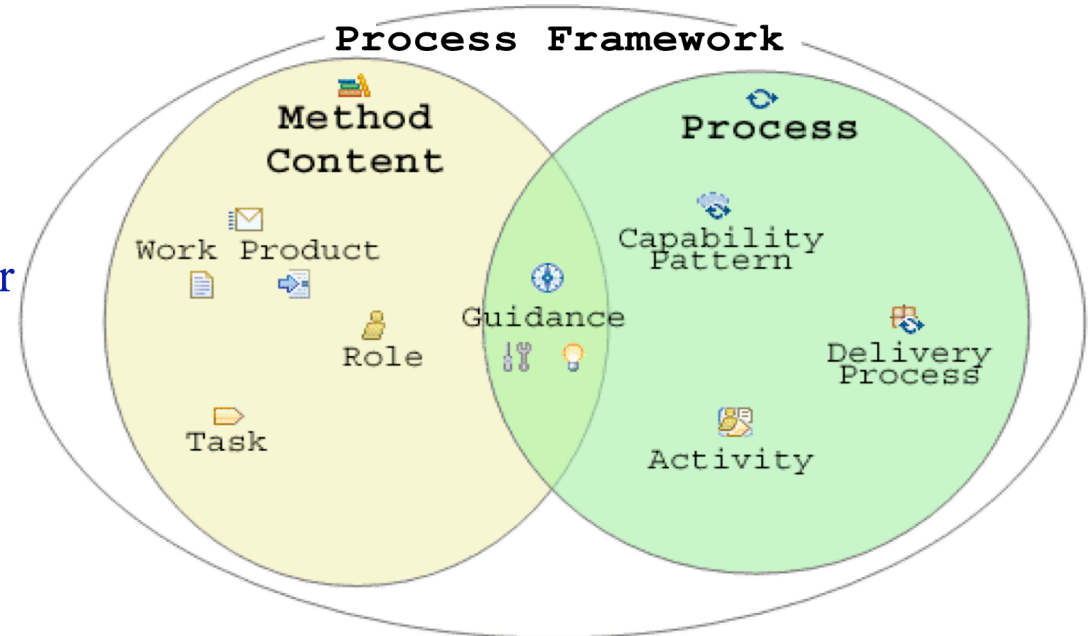
Reda Bendraou<sup>1</sup>, Benoît Combemale<sup>2</sup> Xavier Crégut<sup>2</sup> & Marie-Pierre Gervais<sup>1</sup>

<sup>1</sup> Laboratoire d'Informatique de Paris 6 (LIP6), France  
[first\\_name.last\\_name@lip6.fr](mailto:first_name.last_name@lip6.fr)

<sup>2</sup> Institut de Recherche en Informatique de Toulouse (CNRS UMR 5505), Toulouse, France  
[first\\_name.last\\_name@enseeiht.fr](mailto:first_name.last_name@enseeiht.fr)

# SPEM2.0 (Software Process Engineering Metamodel): Vision

- **OMG's Standard for Software Process Modeling**
  - MOF Metamodel + UML2.0 Profile
  - A revision to SPEM1.1 (2005)
- A clear separation between method contents (work products, roles, guidance, etc.) and their instantiations within a specific development process.
  - **Method Vs Process**
- Extension, reusability and process customization as main preoccupations
  - Plugins, Variability Types,...



## SPEM2.0 RFP Motivations (issued by end of 2004)

- To resolve lacks and semantics ambiguities of SPEM1.1
  - Feedbacks from different users and implementers of SPEM1.1 (industrials et academics, tool implementers, etc.)
- Alignment with UML2.0
  - More expressive (e.g. *UML2.0 Activity*), more modular, Diagram Interchange, etc.
- To ensure software process model executions (enactment)
- Switching from DTD to XML schema
  - XML schema more expressive than DTD
  - XMI/MOF2.0-based XML schema for the MOF SPEM2.0 metamodel

## SPEM2.0 Propositions for Process Model Enactments (1/2)

### 1) An export towards process management tools (e.g. IBM Rational Portfolio Manager, MS Project)

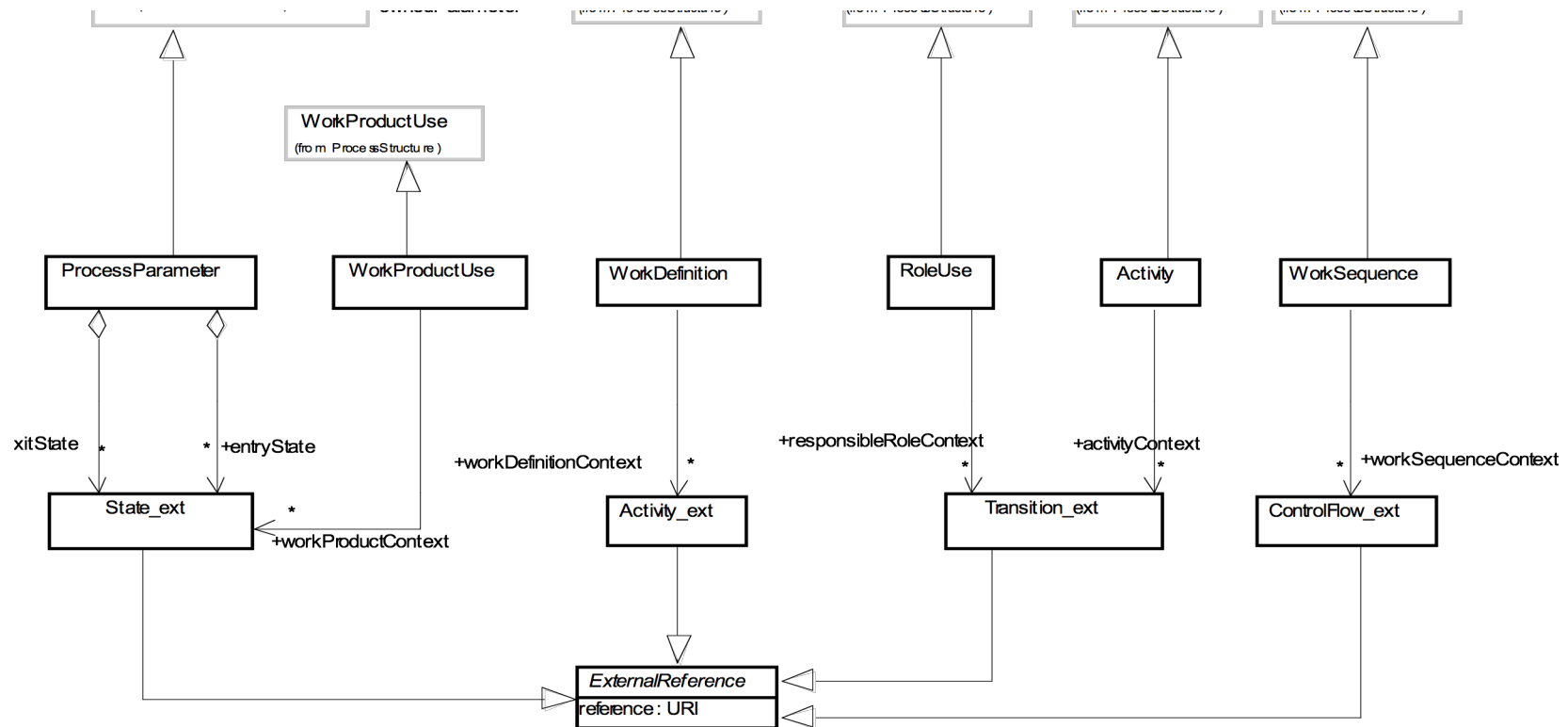
- Still need to affect duration to tasks, persons to roles, etc. in order to get an estimation of the process's duration and required resources
- Project Plans used by project manager for estimation purposes only
- **No concrete process execution**: no automatic rooting of artifacts or control flow regulation, no guards verifications, etc.
- Dependability to project planning tools



# SPEM2.0 Propositions for Process Model Enactments (2/2)

## 2) Linking SPEM2.0 process elements with external behavioral formalisms (Process Behavior package)

- Allows process behavior descriptions by means of references to external behavior models (Activity diagrams, workflows, etc).



## SPEM2.0 Propositions for Process Model Enactments (2/2)

### 2) Linking SPEM2.0 process elements with external behavioral formalisms (Process Behavior package)

- Allows process behavior descriptions by means of references to external behavior models (Activity diagrams, workflows, etc).
- Combining different behavioral models for process descriptions (e.g. reuse of UML2.0 Activity diagrams and state machines)
  - How to derive a process execution from these models?
- **Mapping to behavioral models is up to tool implementers**
  - No standard mapping is proposed
  - You need to develop your own process engines or reuse already existing ones (e.g. BPEL)
  - Feasibility of the approach?

# Outline

- Process models enactment in SPEM2.0

- **xSPEM**: SPEM2.0 extension for software process validation & execution

- **xSPEM** Software Process Model Validation

- xSPEM to Petri Nets

- **xSPEM** Software Process Execution

- xSPEM to WS-BPEL2.0

- xSPEM Process Models Interpretation

- Conclusion

# SPEM2.0 Metamodel: Structure

- Seven Packages with an increasing complexity

- Implementation: Six levels of conformity

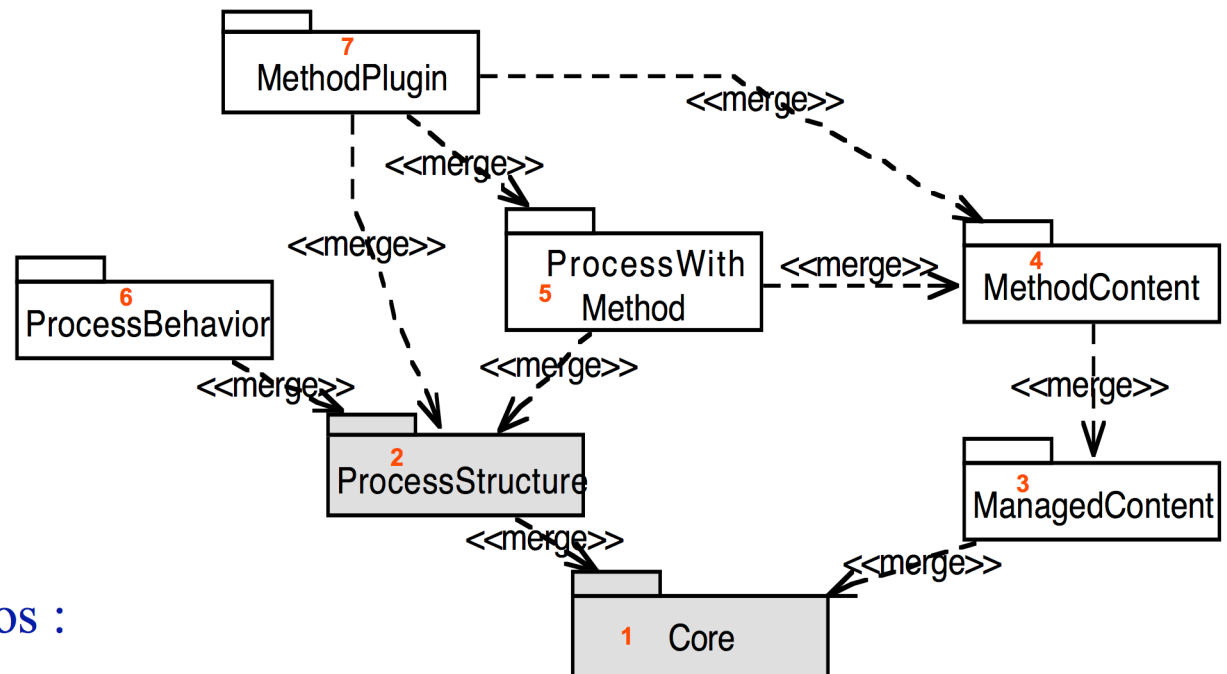
Level 1= Implementation of "Process Structure"

.....

Niveau 6=Niveau 5+  
"Method Plugin"

- Other implementation scenarios :

- Process Structure with Managed Content
- Process Structure with Method Plugin
- Method Content with Managed Content




## **xSPEM Core: Identification**

# xSPEM Core: Identification

- Core elements
  - WorkDefinition
  - WorkDefinitionParameter
  - Kind, etc.

# xSPEM Core: Identification

- Process elements
    - Activity
    - WorkSequence
    - RoleUse
    - WorkProductUse, etc.
  - Core elements
    - WorkDefinition
    - WorkDefinitionParameter
    - Kind, etc.
- 

# xSPEM Core: Identification

- Process elements
    - Activity
    - WorkSequence
    - RoleUse
    - WorkProductUse, etc.
  - Core elements
    - WorkDefinition
    - WorkDefinitionParameter
    - Kind, etc.
- 
- ```
graph TD; PE[Process elements] --> CE[Core elements]; CE --> BM[Behavior models];
```
- Behavior models
    - External Reference (eg. D. Activités, SM, etc)



# xSPeM Core: Identification

- Documenting process and method elements

- Content Description
- Section
- Guidance, etc

- Process elements

- Activity
- WorkSequence
- RoleUse
- WorkProductUse, etc.

- Core elements

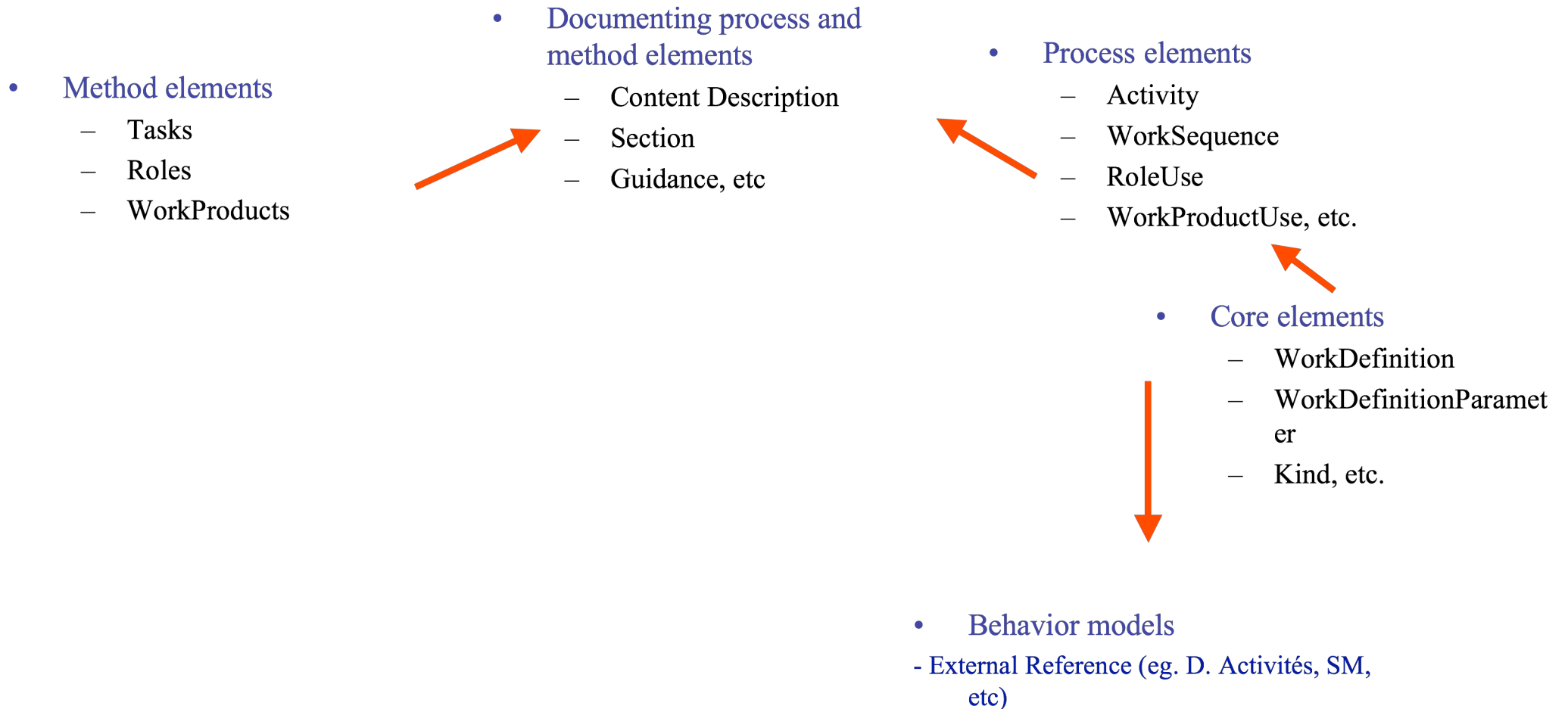
- WorkDefinition
- WorkDefinitionParameter
- Kind, etc.

- Behavior models

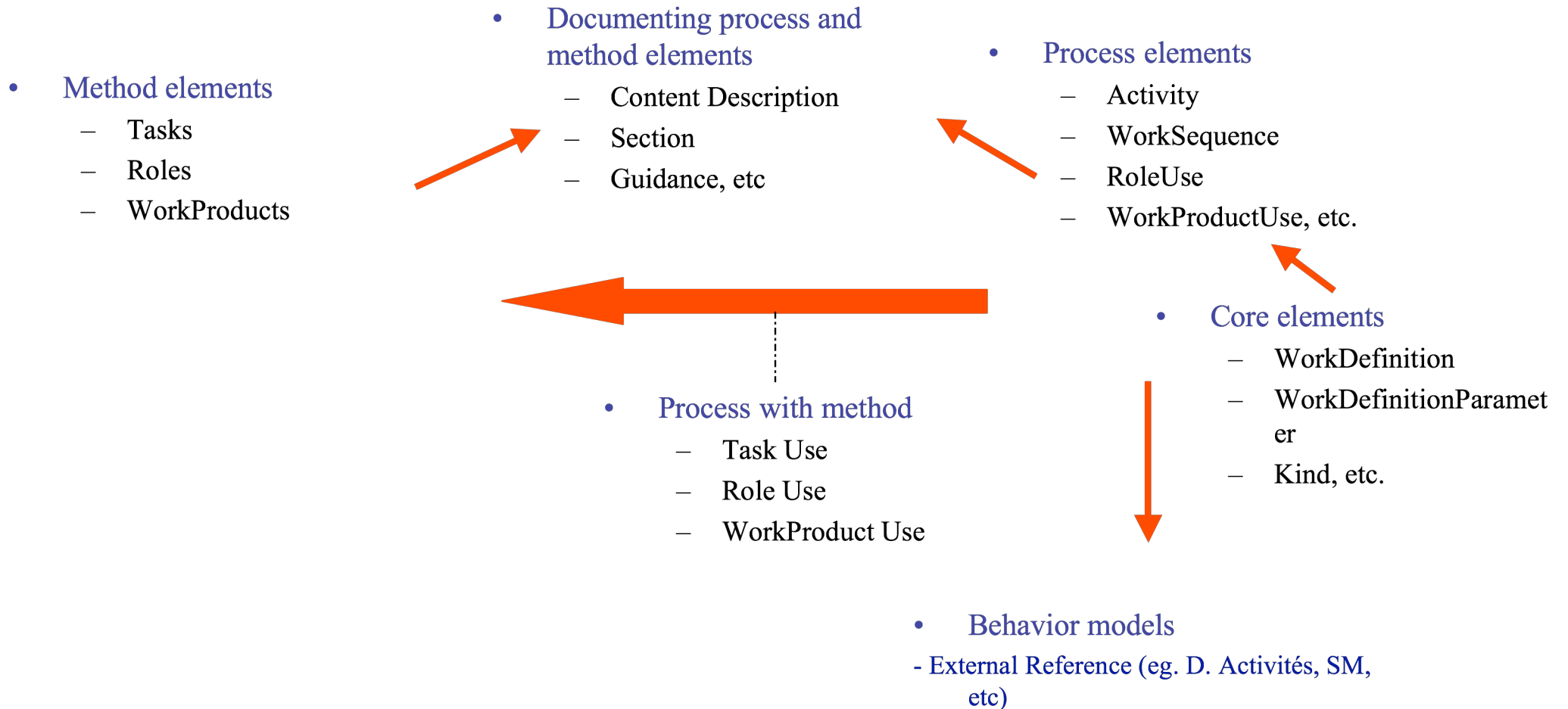
- External Reference (eg. D. Activités, SM, etc)



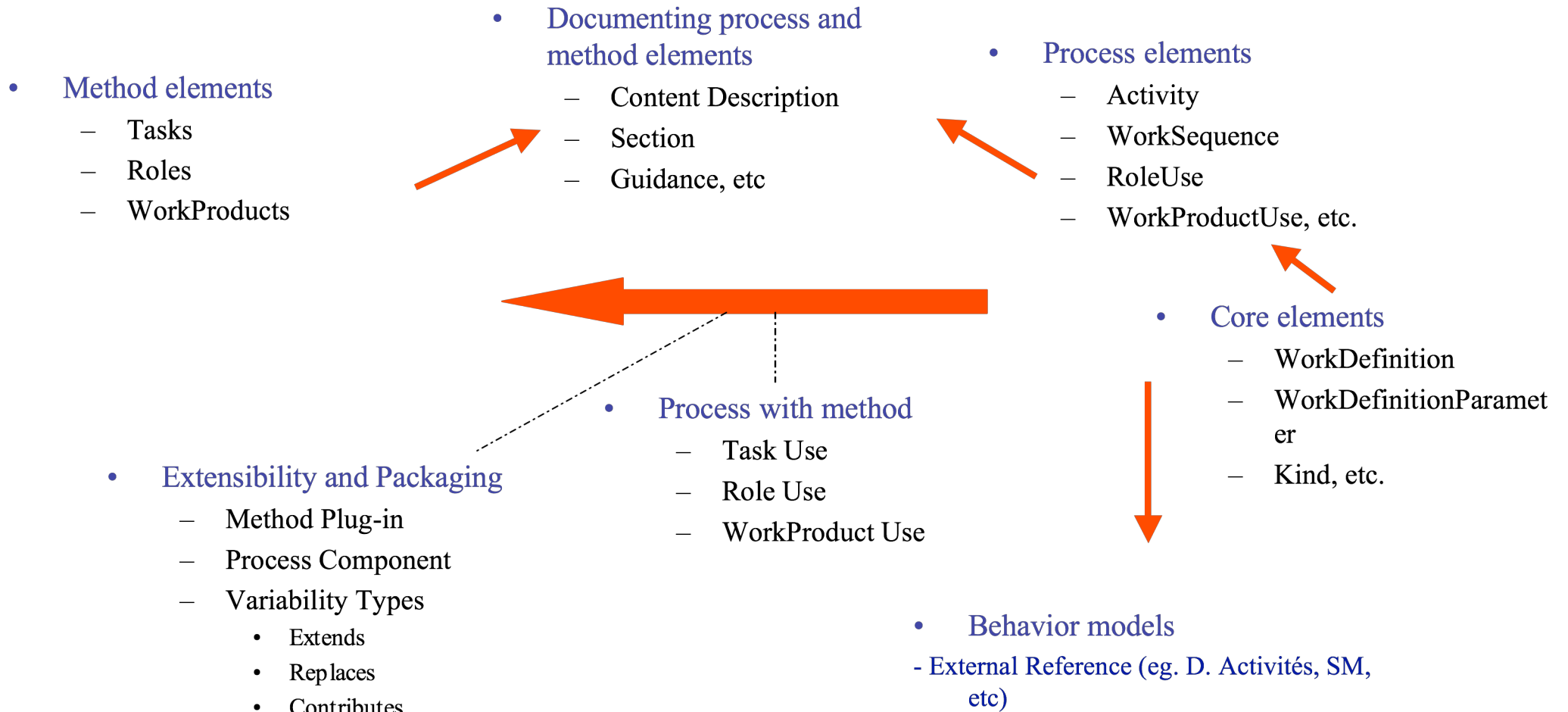
# xSPeM Core: Identification



# xSPeM Core: Identification




# xSPeM Core: Identification

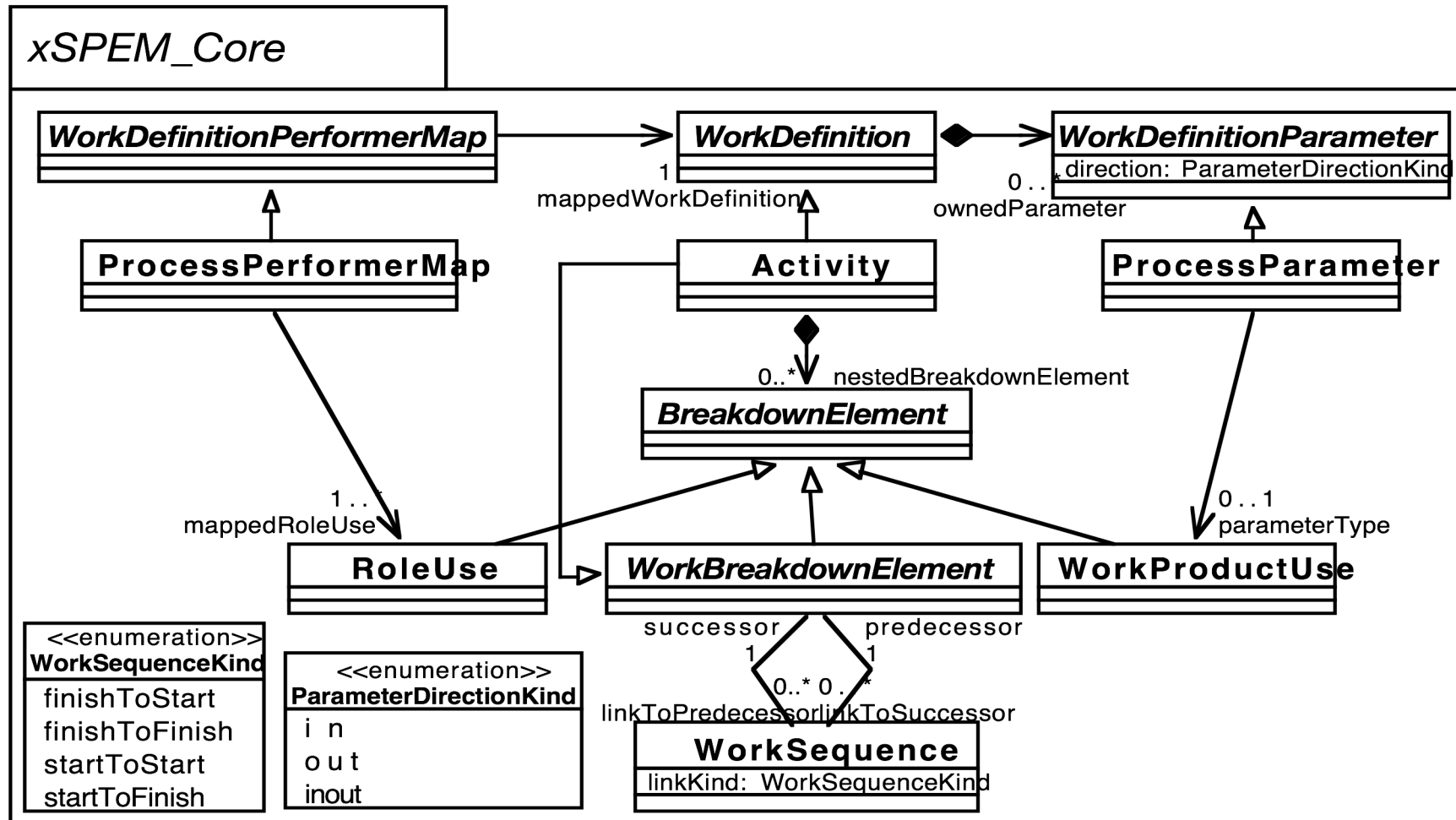


# xSPeM Core: Identification

## xSPeM\_Core

- Process elements
    - Activity
    - WorkSequence
    - RoleUse
    - WorkProductUse, etc.
  - Core elements
    - WorkDefinition
    - WorkDefinitionParameter
    - Kind, etc.
- 

# xSPeM Core: Identification

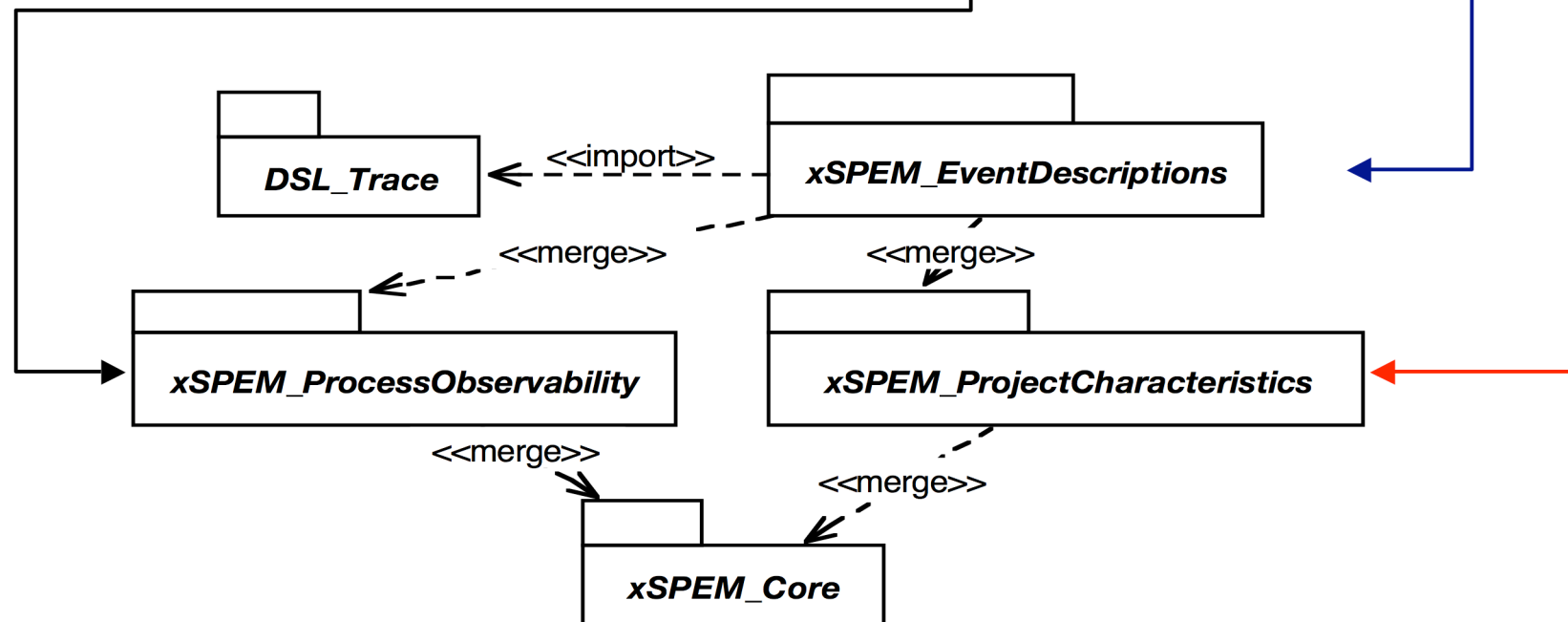


# xSPEM: extension for Process Enactment

## - General Presentation -

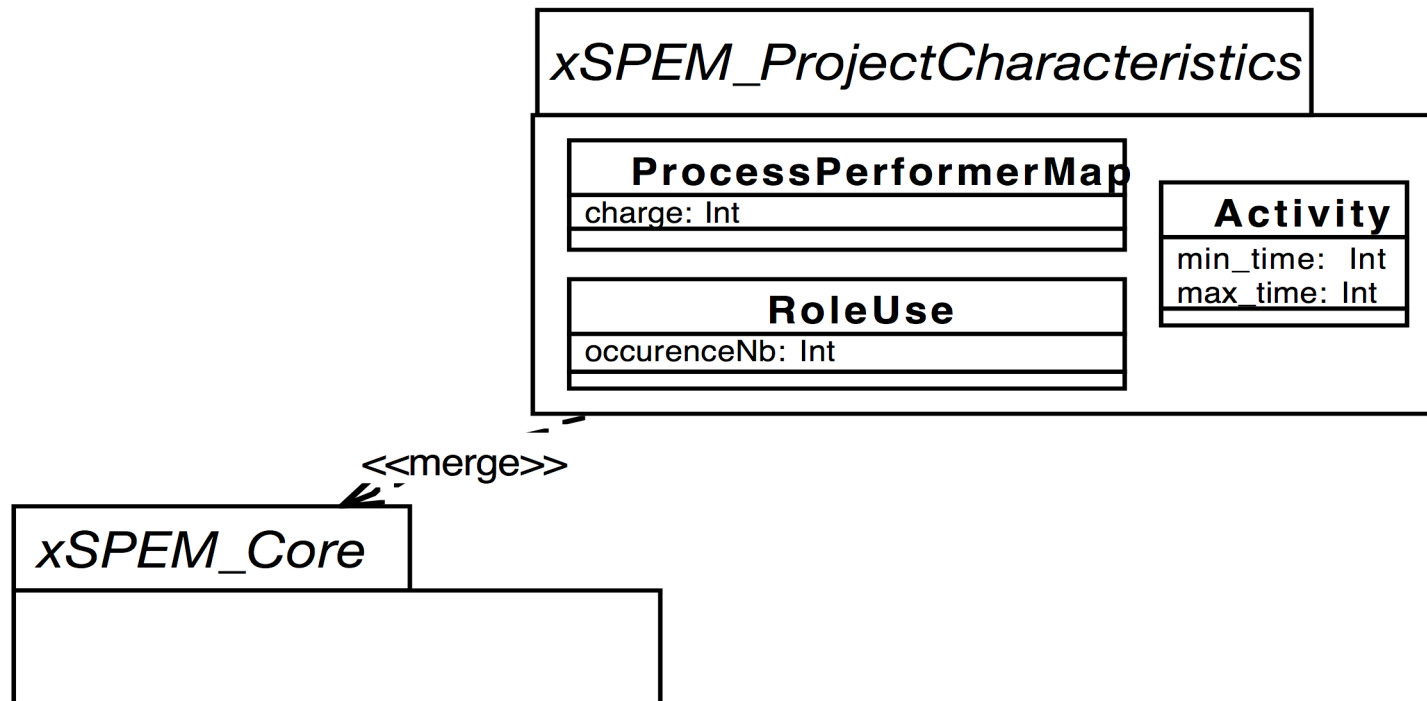
- SPEM 2.0 extension for process enactments includes:

- Project instantiation
- Process state definition
- Events definition for process evolution



# xSPEM: extension for Process Enactment

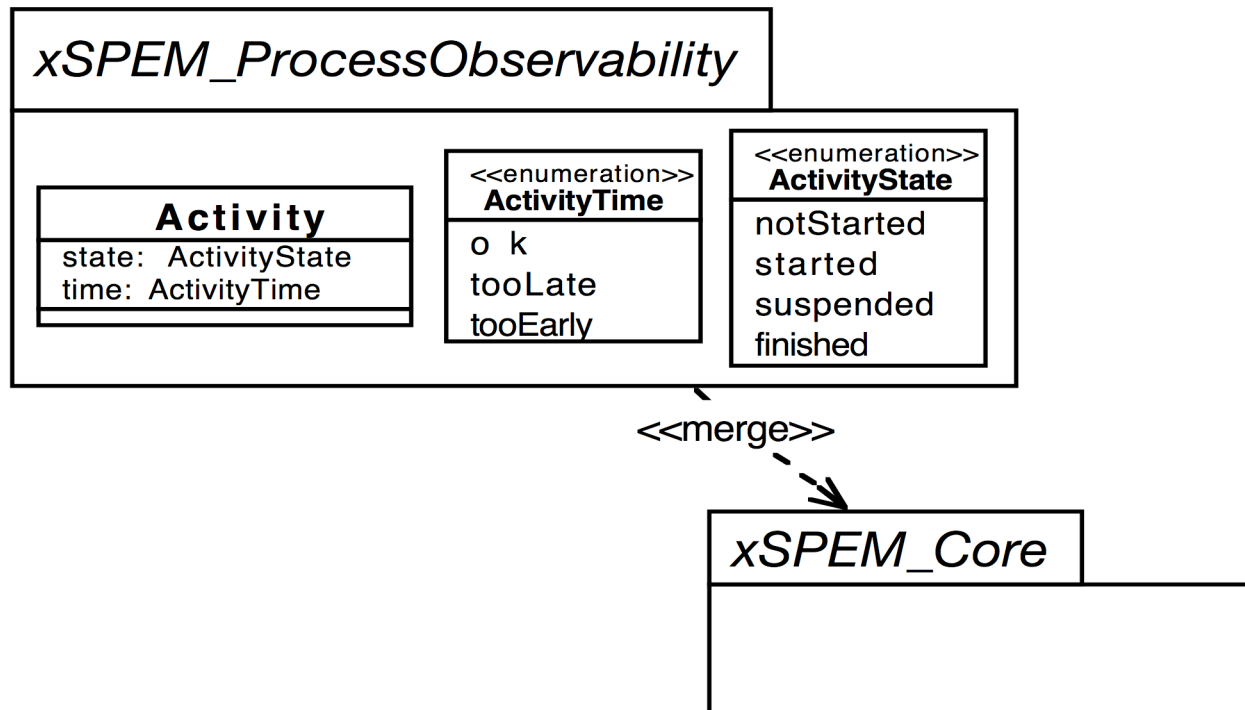
## - Project instantiation -





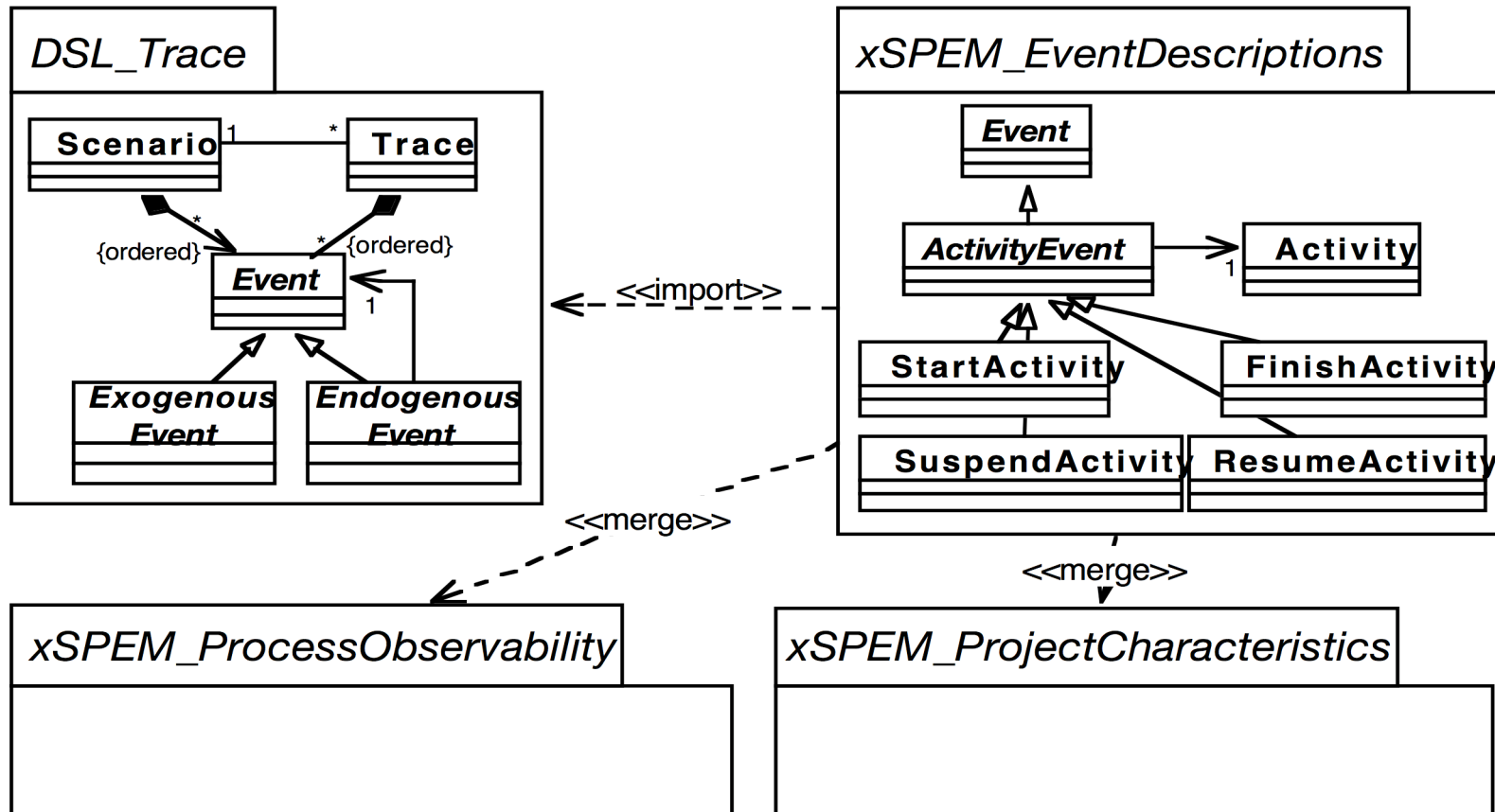
# xSPEM: extension for Process Enactment

- State definition -



# xSPEM: extension for Process Enactment

## - Event Definitions -



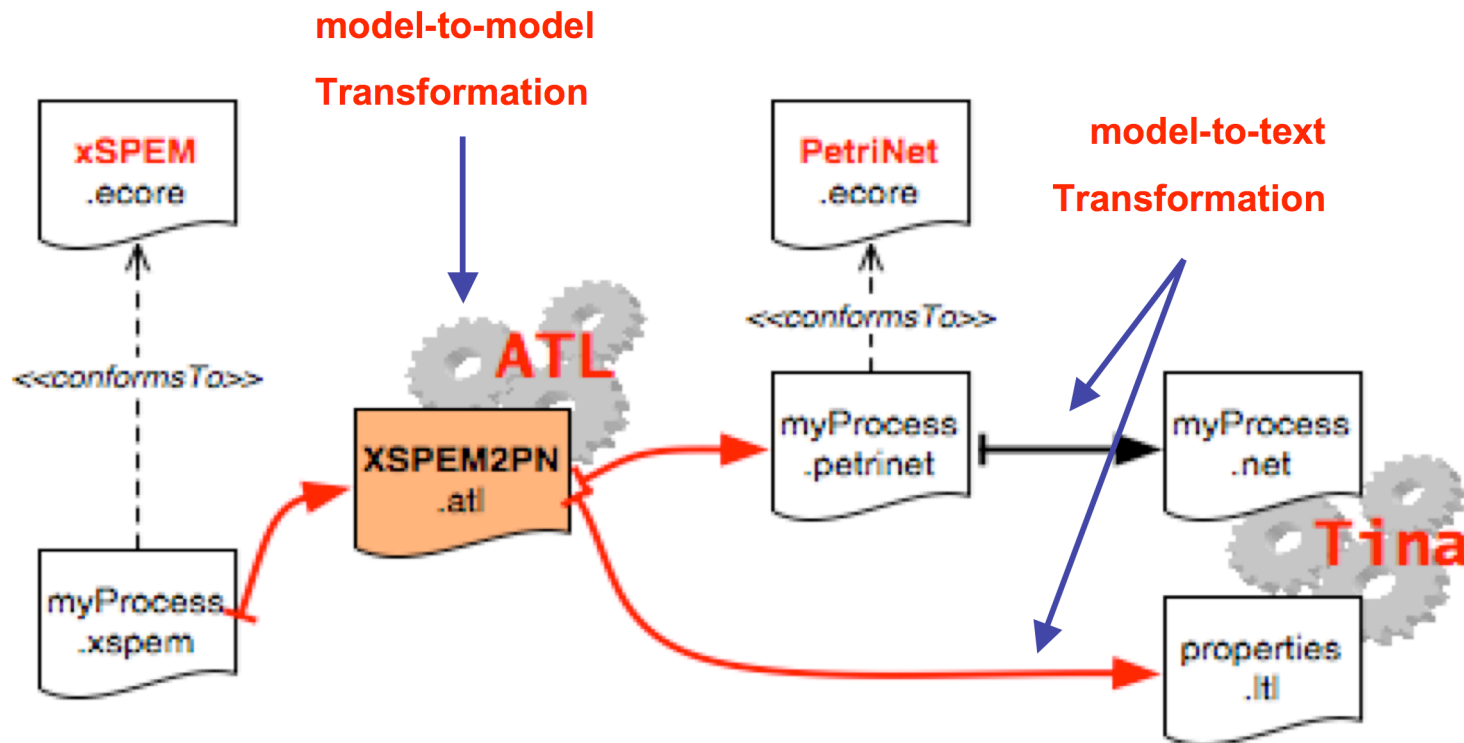
# Outline

- **Process models enactment in SPEM2.0**
- **xSPEM: SPEM2.0 extension for software process validation & execution**
- **xSPEM Software Process Model Validation**
  - xSPEM to Petri Nets
- **xSPEM Software Process Execution**
  - xSPEM to WS-BPEL2.0
  - xSPEM Process Models Interpretation
- **Conclusion**

# Validation of xSPEM Process Models

- Formalization of xSPEM Semantics by a translation toward a formal model (time Petri net)
  - ⇒ *definition of the xSPEM translational semantics*
- Goal: to reuse simulators and model-checkers in order to validate and to simulate xSPEM process models
- Implementation using MDE (EMF/Ecore) and Models Transformation (ATL)

## xSPEM-to-Petri Net: the process



⇒ Sources: Included in the Topcased project, <http://www.topcased.org/>

⇒ Principles of the translation given as ATL Use Cases of the Eclipse website for an xSPEM subset (SimplePDL) :

<http://www.eclipse.org/m2m/atl/usecases/SimplePDL2Tina/>

# Outline

- **Process models enactment in SPEM2.0**
- **xSPEM: SPEM2.0 extension for software process validation & execution**
- **xSPEM Software Process Model Validation**
  - xSPEM to Petri Nets
- **xSPEM Software Process Execution**
  - xSPEM to WS-BPEL2.0
  - xSPEM Process Models Interpretation
- **Conclusion**

## **xSPEM: Enactment**

## **xSPEM: Enactment**

### **Two Approaches were explored**

- Mapping of **xSPEM** process models toward a Process Execution Language
- Implementation of an Interpreter for **xSPEM** models : The **xSPEM** Execution Model



## **xSPEM: Enactment**

### **Two Approaches were explored**

- Mapping of **xSPEM** process models toward a Process Execution Language
- Implementation of an Interpreter for xSPEM models : The xSPEM Execution Model

# The xSPEM-2-WS-BPEL Approach

- Mapping of xSPEM models toward WS-BPEL
  - BPEL (Business Process Execution Language)
  - A well established standard with a tooling support

| xSPEM                                                      | WS-BPEL                                                                                                                                                                                                                                                                                  |
|------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Activity (Outermost)                                       | BPEL Process (next, a BPEL Sequence Activity is required to incorporate nested activities)                                                                                                                                                                                               |
| Activity (Nested)                                          | BPEL Invoke Activity with a Receive Activity                                                                                                                                                                                                                                             |
| Activity's properties (min_time and max_time, state, time) | BPEL Variable                                                                                                                                                                                                                                                                            |
| WorkProductUse                                             | BPEL Variable                                                                                                                                                                                                                                                                            |
| RoleUse                                                    | BPEL Variable                                                                                                                                                                                                                                                                            |
| WorkSequence :finishToStart                                | BPEL Sequence Activity                                                                                                                                                                                                                                                                   |
| WorkSequence :StartToStart                                 | BPEL Flow Activity combined with the Link element for Synchronization                                                                                                                                                                                                                    |
| WorkSequence :StartToFinish                                | BPEL Flow Activity combined with the Link element for Synchronization                                                                                                                                                                                                                    |
| WorkSequence :finishToFinish                               | BPEL Flow Activity combined with the Link element for Synchronization                                                                                                                                                                                                                    |
| ProcessParameter                                           | BPEL Variable with attribute MessageType equals to the WorkProductUse used as an Activity ProcessParameter. If the Activity has more than one ProcessParameter than one Part (name=processParameterName) and its type is to be defined for each ProcessParameter within the MessageType. |
| ProcessPerformerMap                                        | BPEL Variable                                                                                                                                                                                                                                                                            |

# The xSPEM-2-WS-BPEL Approach

- Mapping of xSPEM models toward WS-BPEL
  - BPEL (Business Process Execution Language)
  - A well established standard with a tooling support

| xSPEM                                                      | WS-BPEL                                                                                                                                                                                                                                                                                  |
|------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Activity (Outermost)                                       | BPEL Process (next, a BPEL Sequence Activity is required to incorporate nested activities)                                                                                                                                                                                               |
| Activity (Nested)                                          | BPEL Invoke Activity with a Receive Activity                                                                                                                                                                                                                                             |
| Activity's properties (min_time and max_time, state, time) | BPEL Variable                                                                                                                                                                                                                                                                            |
| WorkProductUse                                             | BPEL Variable                                                                                                                                                                                                                                                                            |
| RoleUse                                                    | BPEL Variable                                                                                                                                                                                                                                                                            |
| WorkSequence :finishToStart                                | BPEL Sequence Activity                                                                                                                                                                                                                                                                   |
| WorkSequence :StartToStart                                 | BPEL Flow Activity combined with the Link element for Synchronization                                                                                                                                                                                                                    |
| WorkSequence :StartToFinish                                | BPEL Flow Activity combined with the Link element for Synchronization                                                                                                                                                                                                                    |
| WorkSequence :finishToFinish                               | BPEL Flow Activity combined with the Link element for Synchronization                                                                                                                                                                                                                    |
| ProcessParameter                                           | BPEL Variable with attribute MessageType equals to the WorkProductUse used as an Activity ProcessParameter. If the Activity has more than one ProcessParameter than one Part (name=processParameterName) and its type is to be defined for each ProcessParameter within the MessageType. |
| ProcessPerformerMap                                        | BPEL Variable                                                                                                                                                                                                                                                                            |

A Sample of mapping rules →

## The **xSPEM-2-WS-BPEL** Approach: Results

## The xSPEM-2-WS-BPEL Approach: Results

### ✓ Positive aspects

- + Leveraging the existing tools and technologies in the Business Process domain
- + A clear separation between process description (high-level) and its execution (low-level)
  - Hide the complexity of the BPEL code through the transformation

## The xSPEM-2-WS-BPEL Approach: Results

### ✓ Positive aspects

- + Leveraging the existing tools and technologies in the Business Process domain
- + A clear separation between process description (high-level) and its execution (low-level)
  - Hide the complexity of the BPEL code through the transformation

### ▼ Negative aspects

- Human interaction aspects are not taken into account by BPEL
- Need for a configuration phase before execution
- Need to regenerate the BPEL code if xSPEM process models are modified
- Modifications in the BPEL code are not traced-up in xSPEM process models

## **xSPEM: Enactment**

### **Two Approaches were explored**

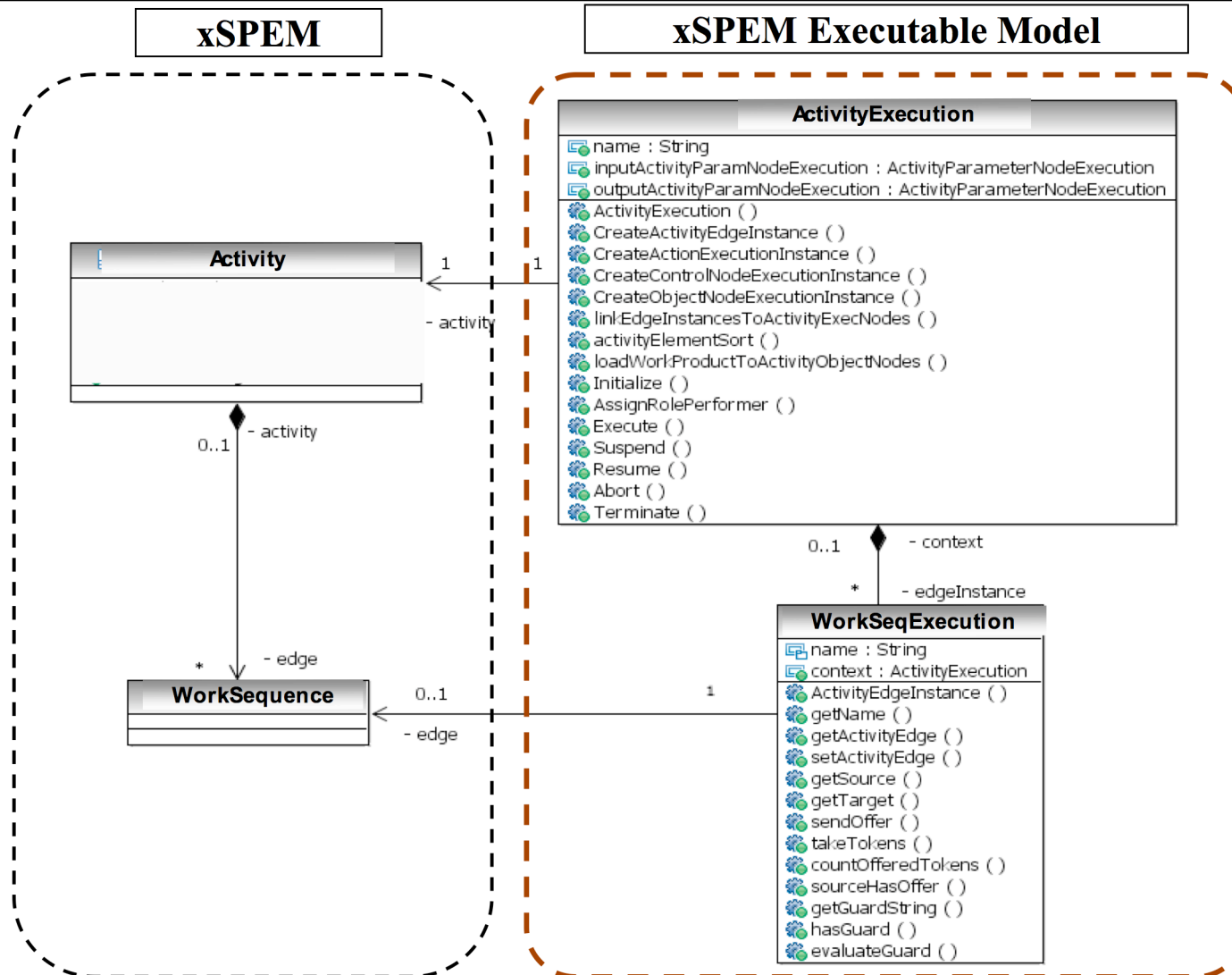
- Mapping of xSPEM process models toward a Process Execution Language
- Implementation of an Interpreter for xSPEM models : The xSPEM Execution Model

# The xSPEM Execution Model Approach

- Definition of an Execution Model for xSPEM
  - Specifies the precise execution behavior of xSPEM elements
  - Comes in form of *classes and operations*
    - Operations aim to reproduce the execution behavior of xSPEM elements
  - Goal: xSPEM Process models executed straightforwardly
- Inspired from the GOF visitor pattern
  - For each xSPEM element → a run time "Execution" visitor class is defined in the execution model
    - Eg. *ActivityExecution, WorkSequenceExecution, etc.*
  - An Association links the *Execution Class* with the *xSPEM Metaclass*
- Used as a Basis of xSPEM Process Engine



# The xSPEM Execution Model: Example



## The xSPEM Execution Model : Results

- xSPEM Execution Model to be used as a Process Engine
  - xSPEM Process Models executed through interpretation
    - **No additional steps or configuration are needed**
- Early feedback through simulation and testing
  - Automation of many iterative and no-interacting process's activities
- Strong Coupling of xSPEM Process Models and their Executions
  - Possibility to modify the process definition at runtime without restarting the process execution
    - **Conditions still need to be identified**
- Connection of the process execution with external applications

# Outline

- **Process models enactment in SPEM2.0**
- **xSPEM: SPEM2.0 extension for software process validation & execution**
- **xSPEM Software Process Model Validation**
  - xSPEM to Petri Nets
- **xSPEM Software Process Execution**
  - xSPEM to WS-BPEL2.0
  - xSPEM Process Models Interpretation
- **Conclusion**

# Conclusion

- Critical analysis of execution within SPEM2.0
- Definition of Core SPEM2.0 for process enactment: *xSPEM\_Core*
- Extension of *xSPEM\_Core* for process execution and validation
- Exploring validation and execution approaches
  - xSPEM to Petri Nets
  - xSPEM to WS-BPEL
  - xSPEM Execution Model
- Evaluation of xSPEM within Modelplex and Topcased projects
  - [www.modelplex-ist.org](http://www.modelplex-ist.org)
  - [www.topcased.org](http://www.topcased.org)



# Perspectives

- Short-terms perspectives
  - xSPEM graphical editor (*Topcased, WP1*)
  - User-friendly integration of xSPEM models validation in the Topcased Toolkit (*Topcased, WP3*)
- Middle-terms perspectives
  - xSPEM simulator (*Topcased, WP2*)
  - xSPEM Execution model implemented using a meta-programming language: Kermeta => <http://www.Kermeta.org>

