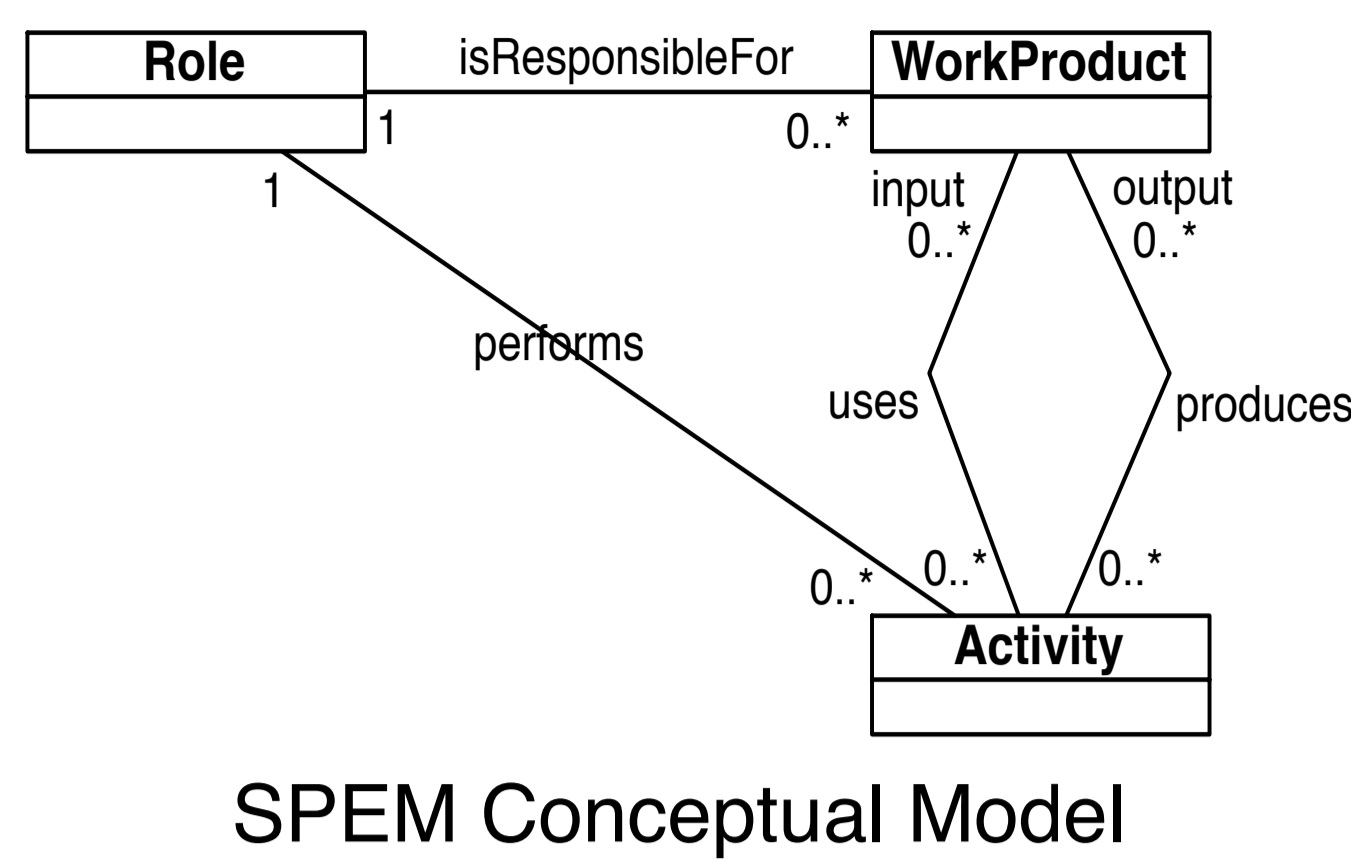
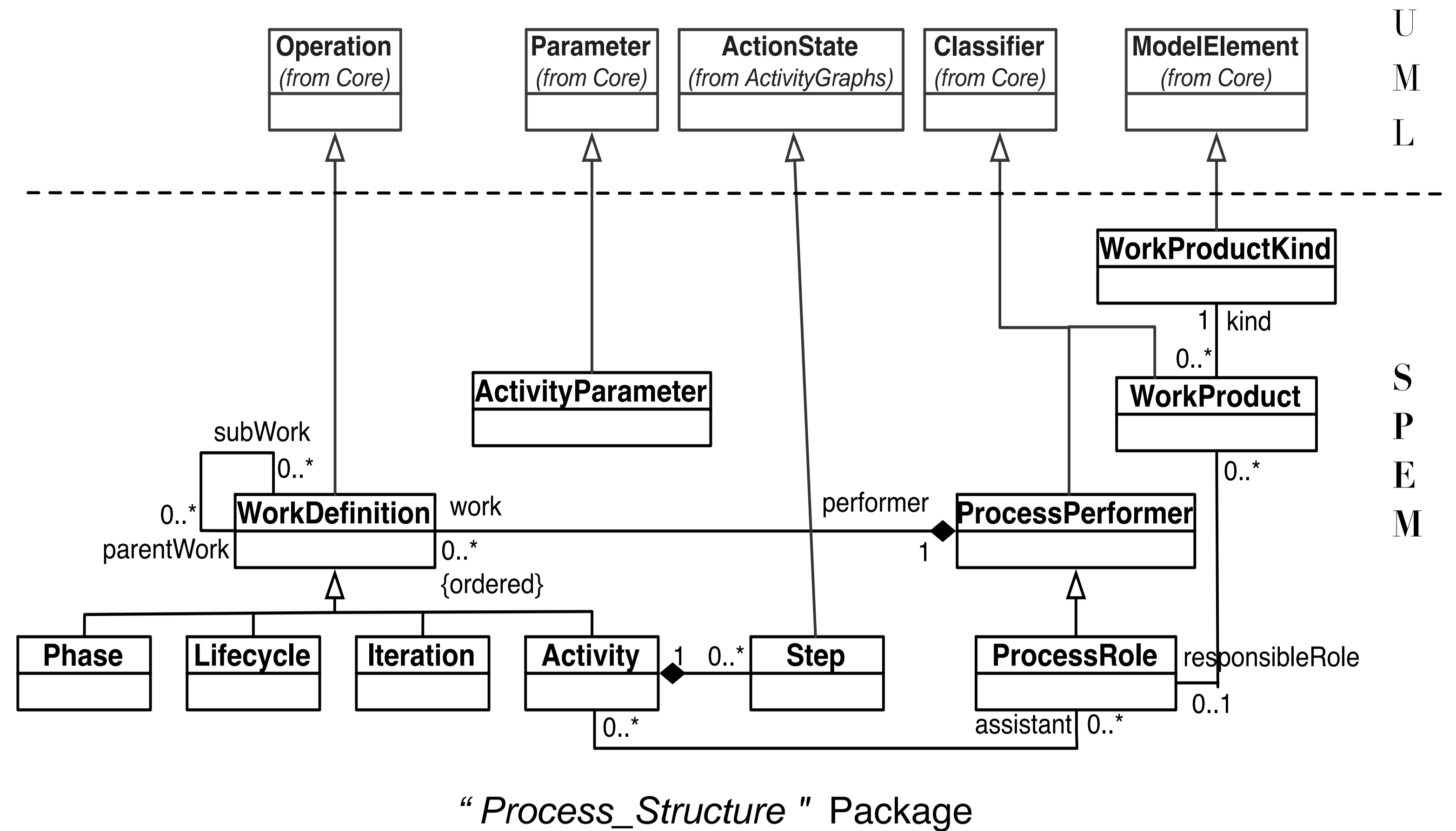


The SPEM metamodel :



- ✓ **OMG modeling language** used for the specification of concrete software development process,
- ✓ Offers an **object-oriented approach using the UML notation**,
 - ⇒ Graphical syntax
 - ⇒ Semi-formal semantics (natural language and OCL constraints).
- ✓ Integrated in the pyramidal architecture of MDA organization as a **MOF metamodel** and as an **UML profile**.



Some ambiguities in SPEM due to:

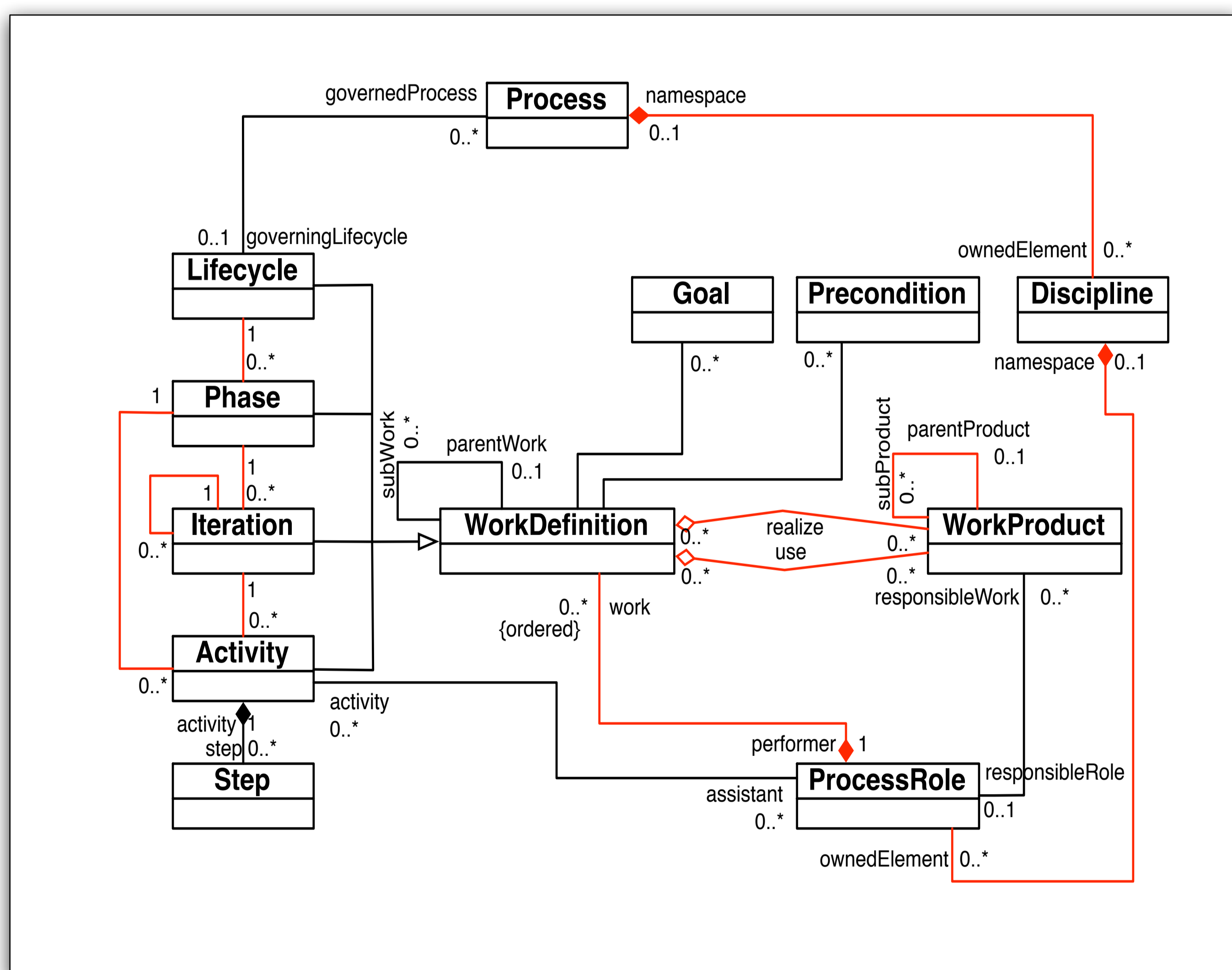
- ✓ OMG proposal is very generalist and provides no directives on how to use it,
- ✓ SPEM inherits some expressiveness from UML which is not adapted to software development process modeling (e.g. link between *WorkDefinition* and *WorkProduct* through UML Parameter metaclass allows to put *ProcessPerformer* as parameter of *WorkDefinition* !),
- ✓ Semantics essentially expressed in natural language that leads to the construction of inconsistent process models because of the lack of a formal definition of concepts (e.g. differences between *Phase*, *Lifecycle*, *Iteration* and *Activity* are only explained in natural language !).

⇒ **Specialization of the SPEM metamodel to define clearly concepts and formally express their semantics with OCL.**

Improving modeling with SPEM :

Restriction of SPEM metamodel

Semantics precisions with OCL constraints



Allows defining an axiomatic semantics
 ⇒ Limit the possible instantiations thus the valid process models.

Examples:

"A role must be responsible for all the products carried out by activities of which he is in charge, and reciprocally".

```
context ProcessRole inv :
let productsActivities : Set{WorkProduct} =
- Definition of the WorkProduct set made by
- the activities of which 'self' is responsible
self.work -> select(a : WorkDefinition |
a.oclIsTypeOf(Activity)).
oclAsType(Activity).output -> asSet()
in
self.workProduct = productsActivities
```

"To suppress the ProcessPerformer concept we have added an OCL constraint that forbids its instantiation".

```
context ProcessPerformer inv :
self.allInstances() -> size() = 0
```

Conclusion:

Being more directives, our proposal brings more assistance in the industrial construction of a process and more facility in the use of SPEM. Thanks to the restrictions put on SPEM, it is possible to ensure the coherence of its models. Our proposal being a restriction of SPEM, our model conforms to SPEM.

Perspectives:

OCL can only capture structural constraints. Our future work is to define an operational semantics for SPEM in order to enact a process model described in SPEM. So we are investigating several approaches including the ones that describe operational semantics for metamodel such as Kermet and xOCL, and endogen transformation language such as ATL or AGG.

Benoit Combemale*, Xavier Crégut*, Alain Caplain and Bernard Coulette****

* **IRIT-LYRE Laboratory**

Toulouse - France

<http://www.irit.fr/LYRE>

{benoit.combemale, xavier.cregut}@enseeiht.fr

** **GRIMM-ISYCOM Laboratory**

Toulouse - France

<http://www.univ-tlse2.fr/grimm/isycom/>

{alain.caplain, bernard.coulette}@univ-tlse2.fr