



26th annual **INCOSE**
international symposium

Edinburgh, UK
July 18 - 21, 2016



Towards V&V suitable Domain Specific Modeling Languages for MBSE: a tooled approach

Blazo Nastov¹, Vincent Chapurlat¹, Christophe Dony² and François Pfister¹



¹*LGI2P, Ecole des mines d'Alès, Nîmes, France*

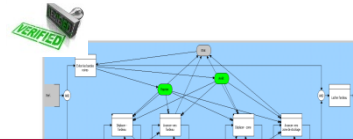
{blazo.nastov, vincent.chapurlat, [francois.pfister](mailto:francois.pfister@mines-ales.fr)}@mines-ales.fr



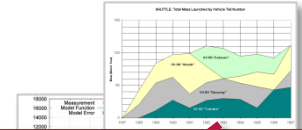
²*LIRMM, University of Montpellier, Montpellier, France*

dony@lirmm.fr

MBSE context: research problematic



Functional /
Behavioural



1) To create **Domain Specific Modeling Language (DSML)** suitable for modelling a system (multi disciplinary, multi views / aspects, ...) i.e. building models

Each Model is...

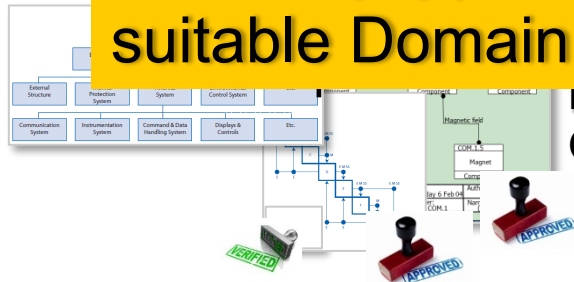
- A partial and incomplete, may be false but crucial representation of the Sol
- An input for reasoning during

2) Model V&V: as
(Verification), just

and conformity
s (Validation)

Contribution:

A tool-equipped approach for the design and use of V&V
suitable Domain Specific Modeling Languages for MBSE



Physical /
Organic
models

Systems Engineering

(Safety, Costs,
Mechanical...)



Agenda

Illustrated background

Contributions

Case study

Conclusion and Perspectives



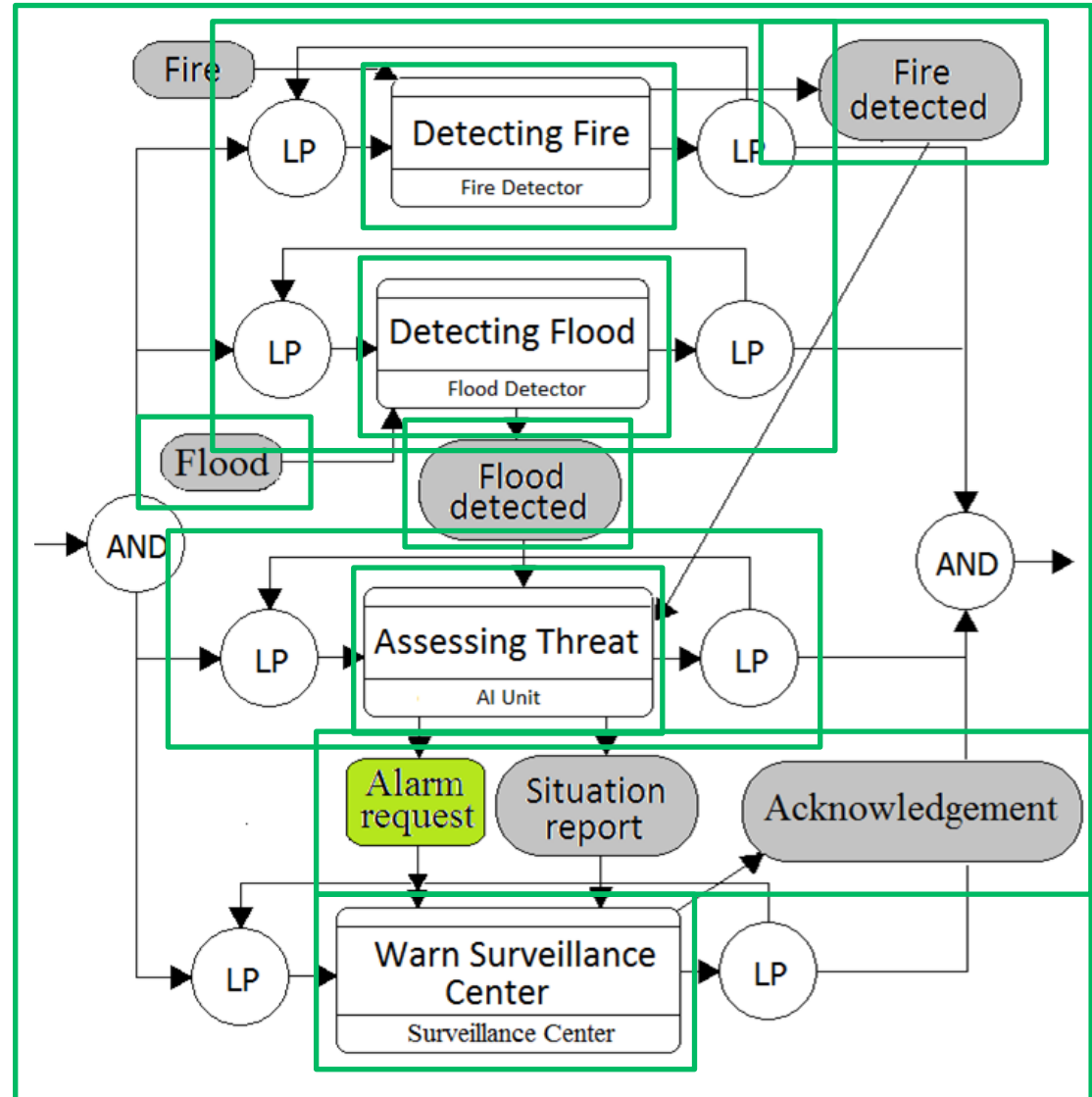
Illustrated background

Illustration: enhanced Functional Flows Block Diagram (eFFBD)

Function

Flow
(I/O, Trigger)

Construct
(parallelism, choice, loop, ...)



Functional / Behavioural model (dynamics of a system)

1) To create DSML: things to consider

1) To create **Domain Specific Modeling Language (DSML)** suitable for modelling a system (multi disciplinary, multi views / aspects, ...) i.e. building models

Semantics

Static

Structural p

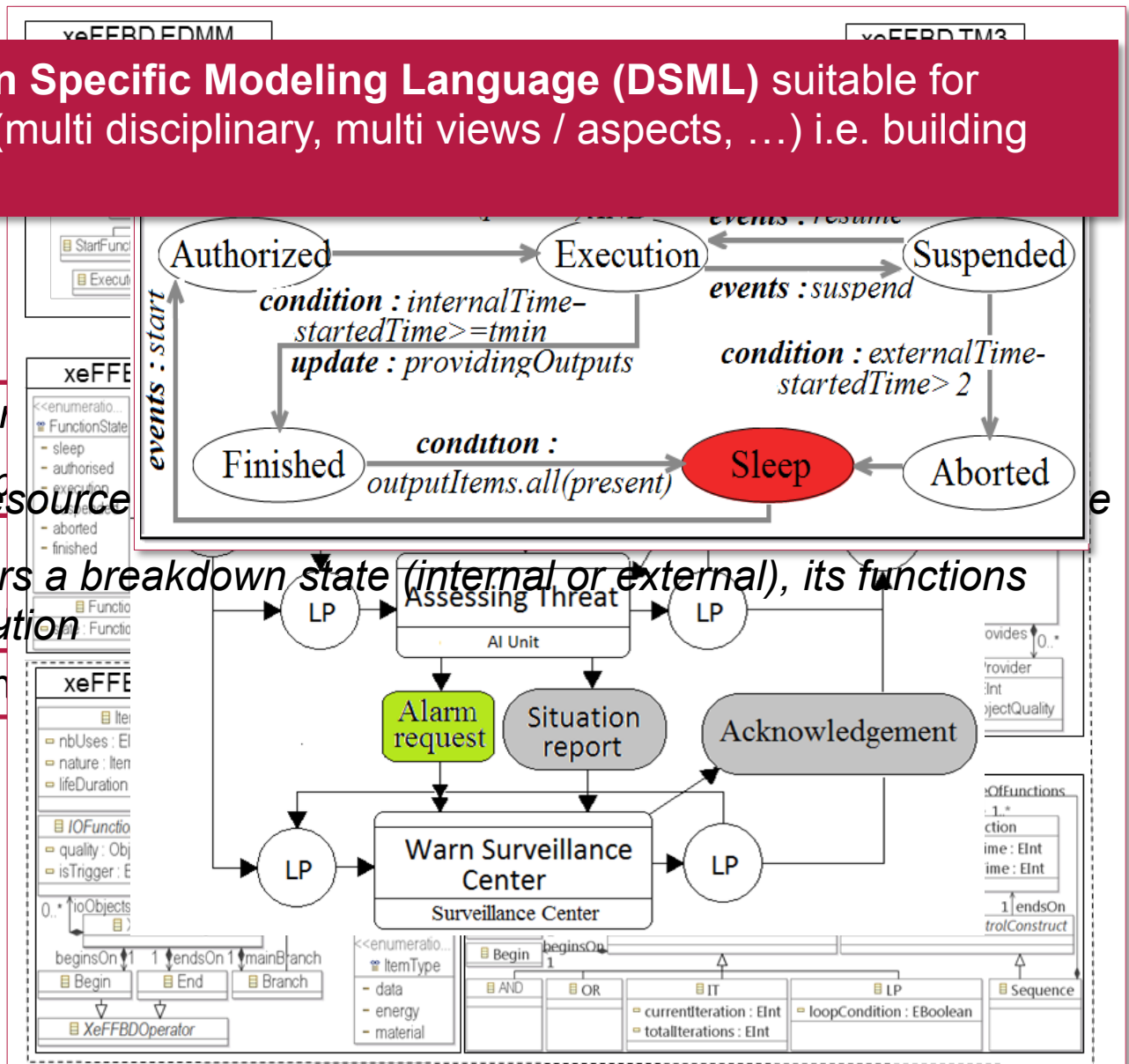
Behavioral p

The quantity of a Resource

or zero

If a component enters a breakdown state (internal or external), its functions must suspend execution

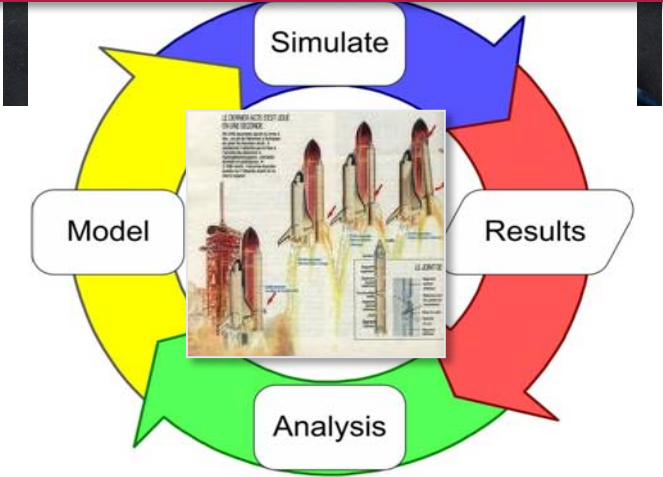
Behavioral n



Excellent

2) Model V&V: assure model's well-formedness, coherence and conformity (**Verification**), justifiability and relevance for expert's objectives (**Validation**)

Formal properties proof



Based on 3rd party formal approaches (M2M transformation)

- (+) *Reuse of existing approaches*
- (-) *Information loss, relevance between the source and the target*

Based on 3rd party formal approaches (M2M transformation)

- (+) *Reuse of existing approaches*
- (-) *Information loss, relevance between the source and the target*

Based on 3rd party formal approaches (M2M transformation)

- (+) *Reuse of existing approaches*
- (-) *Information loss, relevance between the source and the target*

Detailed contribution

A ***tool-equipped approach*** for the design of DSMLs allowing *Verification and Validation of models by using simultaneously simulation and formal properties proof*

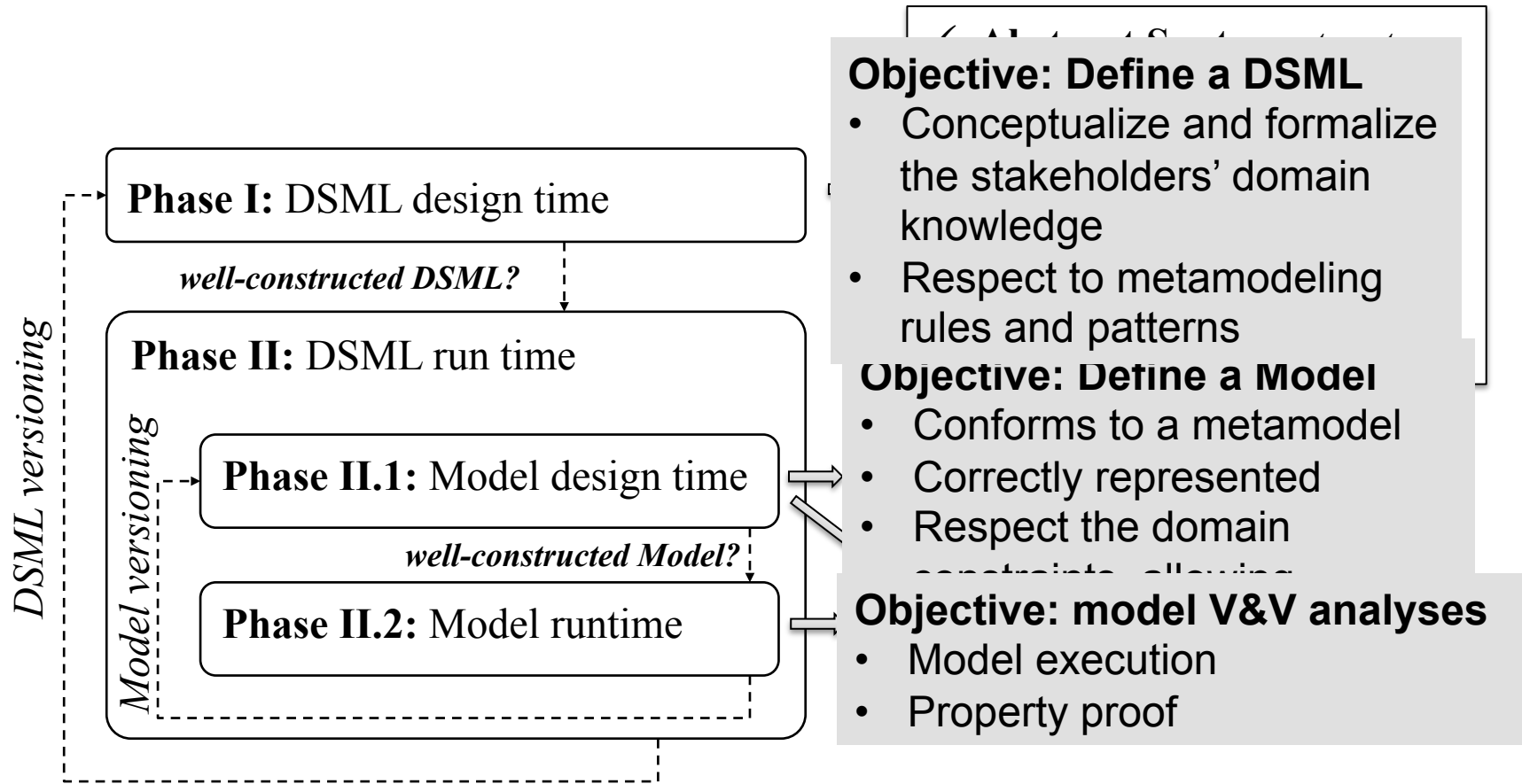
**xviCore: eXecutable, Verifiable
and Interoperable Core**



Contributions

Conceptual contributions: life cycle

Contribution goal: to provide stakeholders with a life cycle including the design of V&V suitable DSMLs and models



Conceptual contributions: meta languages

xviCore follows previously proposed lifecycle

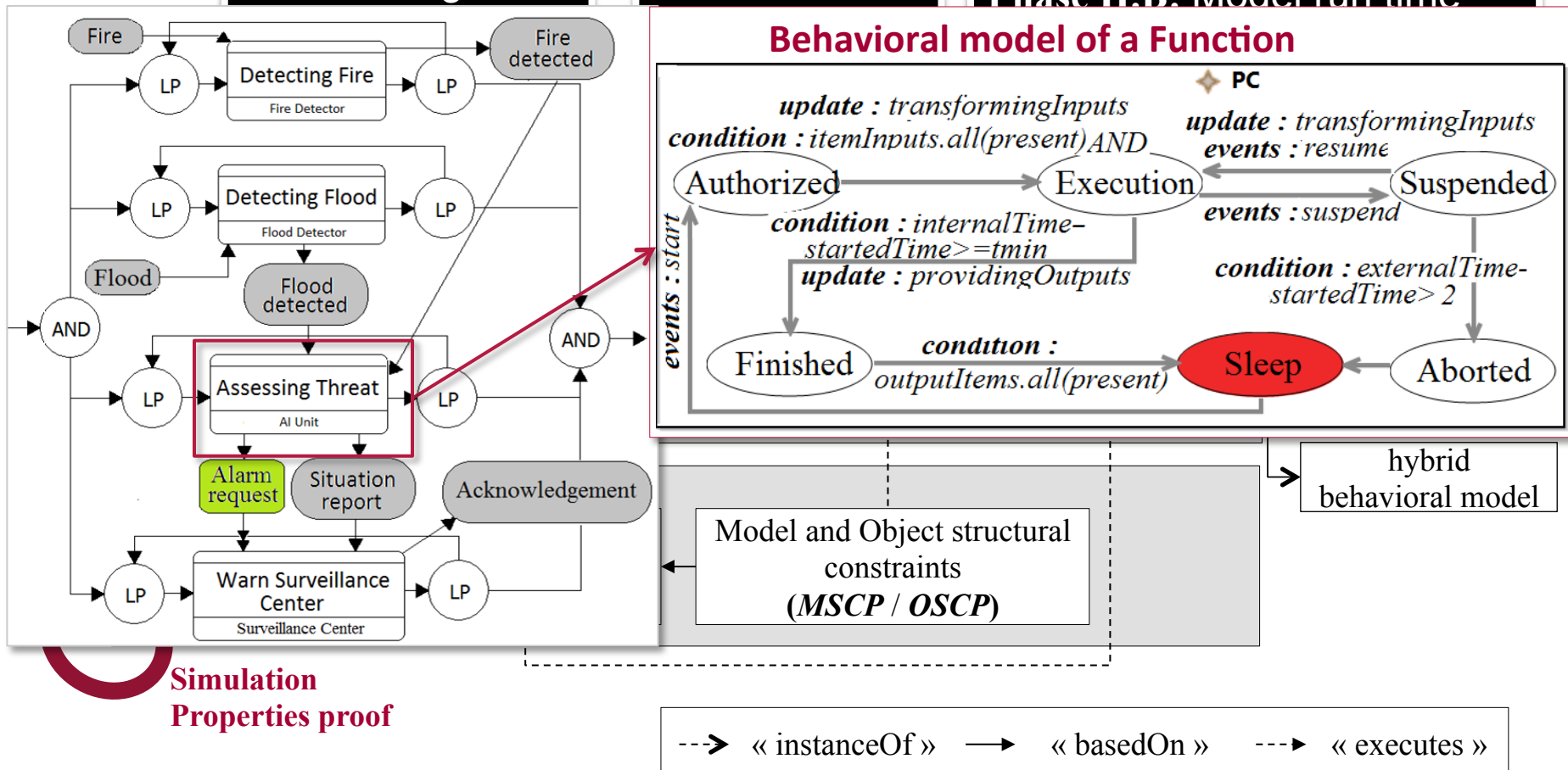
xviCore promotes **3 meta-languages**

Phase I:
DSML design time

Phase II:
DSML run time

Phase II.A: Model design time

Phase II.B: Model run time



Conceptual contributions: simulation mechanisms

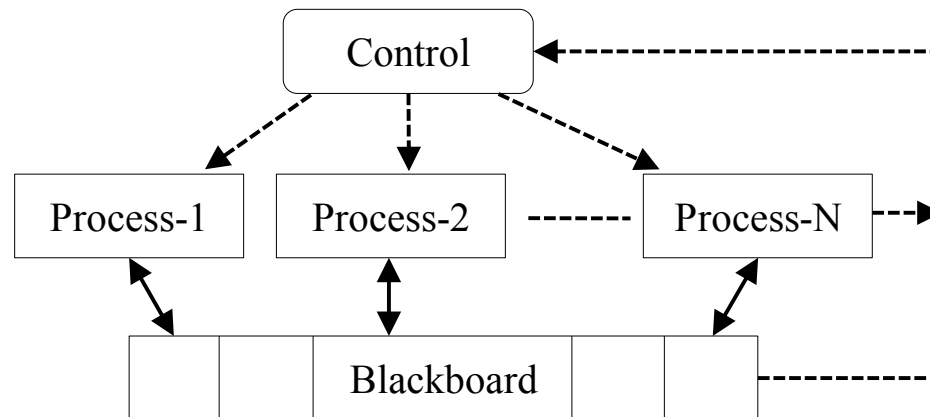
Observation

The operational semantics is defined as a set of **behavioral models** that are associated to chosen domain concepts of a DSML

Problem

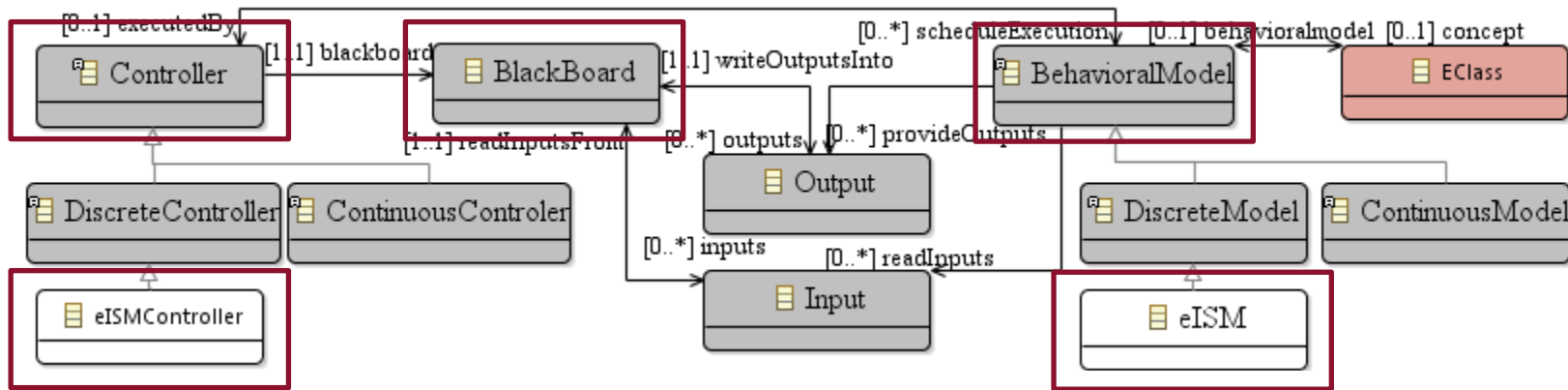
How to coordinate all behavioral models from one or several DSMLs during simulation?

State of the art: Blackboard Design Pattern (Engelmore and Morgan, 1988)



Methodological contribution: simulation mechanisms

Objective: applying the Blackboard Design Pattern in the MBSE context



- Allows data exchange between behavioral models

Proof of ○ Based on a simple execution algorithm (SIM)

(Vandermeulen)

Operational

Separation

- Read inputs from blackboard
- Calculate future state
- Write outputs into blackboard
- and other original rules

Underlying structure based on the LTL (Linear Temporal Logic)

Methodological contribution: verification mechanisms

Formal properties proof mechanisms of xviCore approach

Formal specification

- syntaxes i.e. structure of the model
- dynamic semantics i.e. behavioral models)

Formal properties (structural and behavioral properties)

- using property modelling language: UPSL-SL(Chapurlat, 2013)

A model-checker tool (bridging specification and properties)

Reuse of existing tools :

OCCL interpreter for structural properties

LTL model checker for behavioral properties

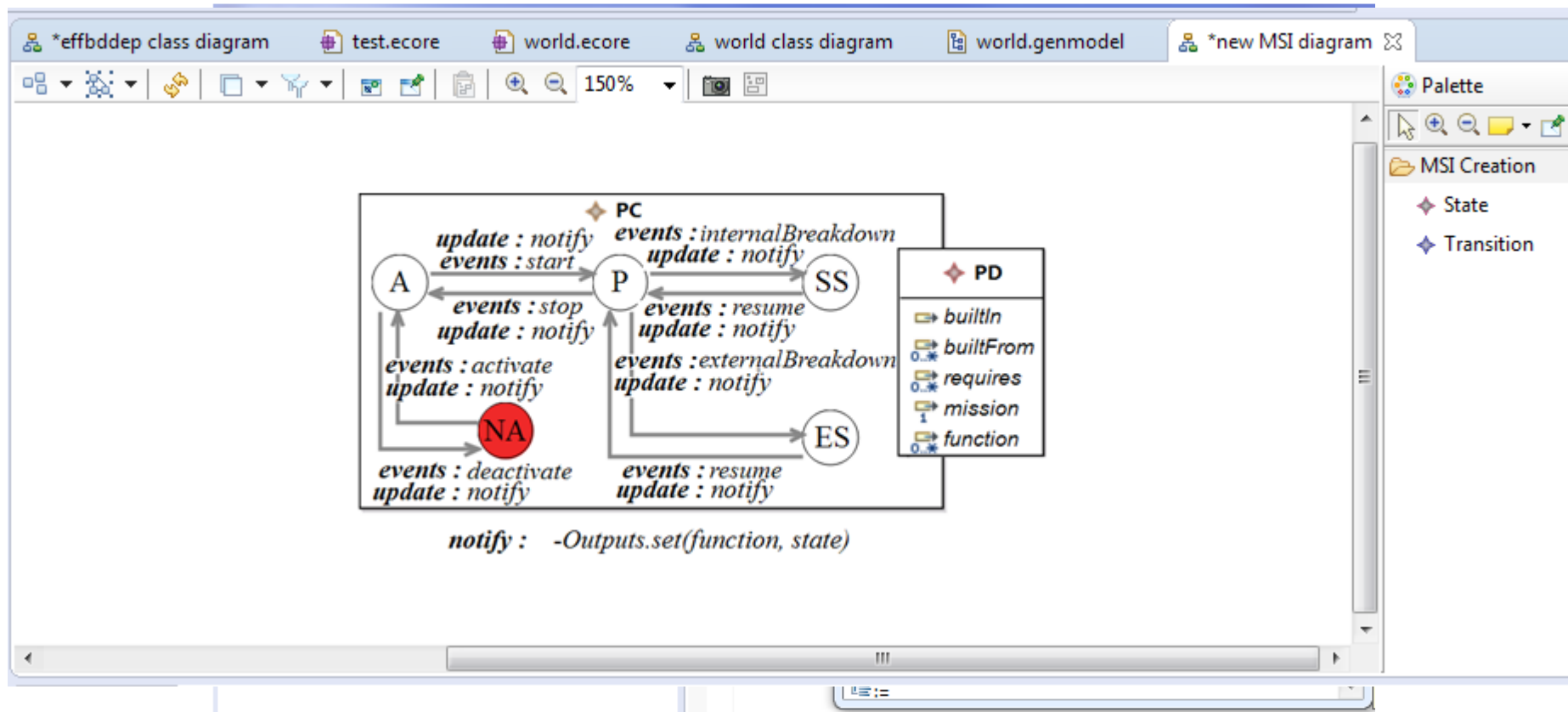
Technical contribution: current editors

DSML design

Editor for the DSML structure

Editor for the DSML behavior

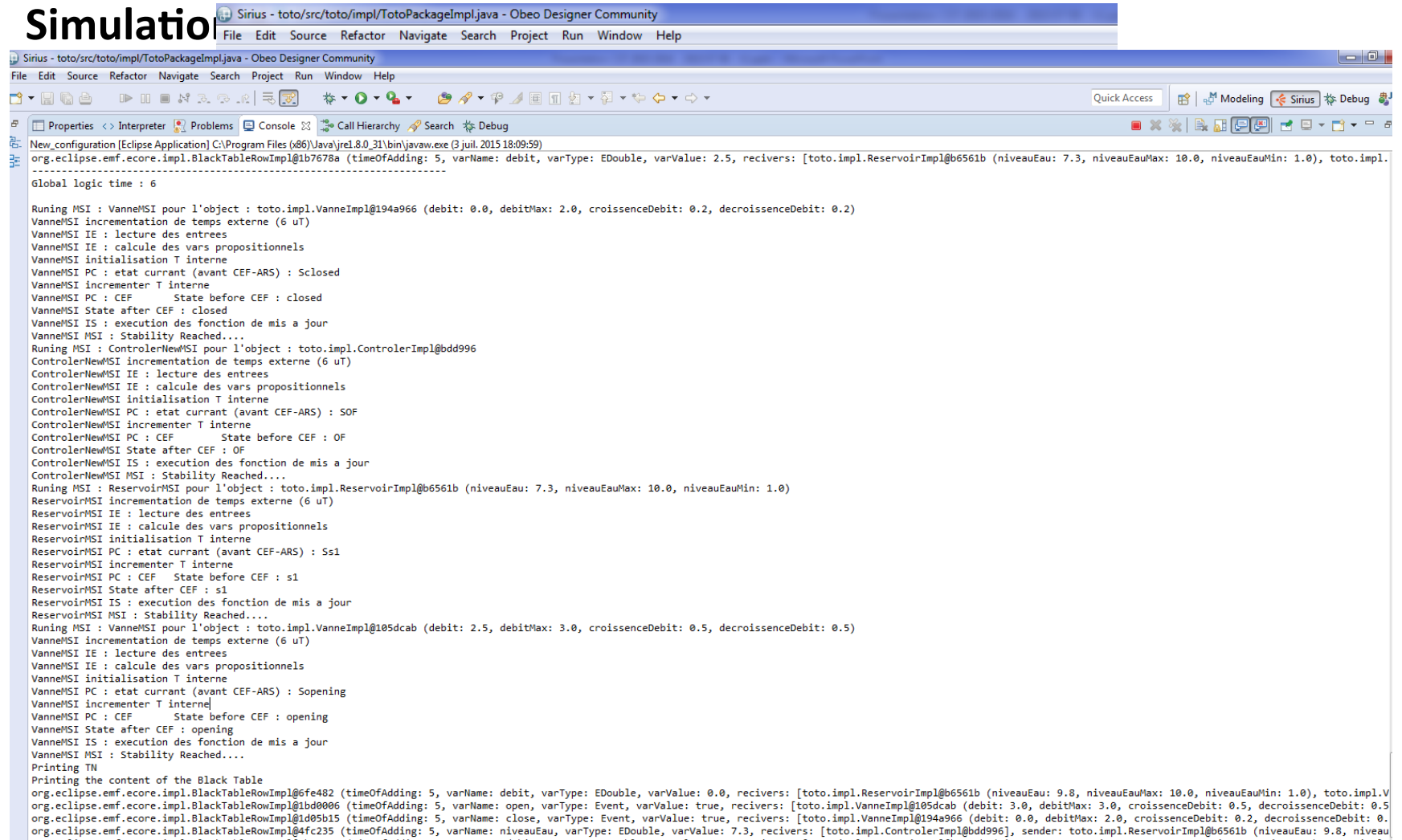
Editor for the DSML constraint



Technical contribution: extending EMF

We have extended the code generation library of EMF
Generation of a simulation library

Simulation



The screenshot shows the Sirius IDE interface with the console window displaying a simulation log. The log contains several entries for different components, including VanneMSI, ControlerNewMSI, and ReservoirMSI, detailing their initialization, state transitions, and stability checks. The log also includes a 'Printing the content of the Black Table' section at the bottom.

```
Sirius - toto/src/toto/impl/TotoPackageImpl.java - Obeo Designer Community
File Edit Source Refactor Navigate Search Project Run Window Help

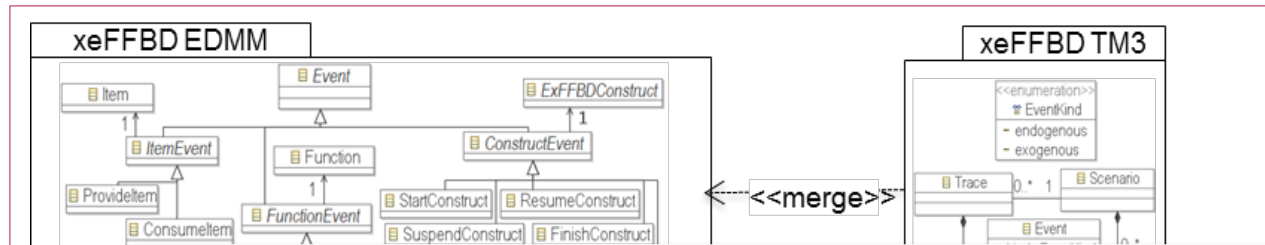
Sirius - toto/src/toto/impl/TotoPackageImpl.java - Obeo Designer Community
File Edit Source Refactor Navigate Search Project Run Window Help

New_configuration [Eclipse Application] C:\Program Files (x86)\Java\jre1.8.0_31\bin\javaw.exe (3 juil. 2015 18:09:59)
org.eclipse.emf.ecore.impl.BlackTableRowImpl@1b7678a (timeOfAdding: 5, varName: debit, varType: EDouble, varValue: 2.5, recivers: [toto.impl.ReservoirImpl@b6561b (niveauEau: 7.3, niveauEauMax: 10.0, niveauEauMin: 1.0), toto.impl.
Global logic time : 6

Running MSI : VanneMSI pour l'objet : toto.impl.VanneImpl@194a966 (debit: 0.0, debitMax: 2.0, croissanceDebit: 0.2, decroissanceDebit: 0.2)
VanneMSI incrementation de temps externe (6 ut)
VanneMSI IE : lecture des entrees
VanneMSI IE : calcul des vars propositionnels
VanneMSI initialisation T interne
VanneMSI PC : etat courant (avant CEF-ARS) : Sclosed
VanneMSI incremter T interne
VanneMSI PC : CEF      State before CEF : closed
VanneMSI State after CEF : closed
VanneMSI IS : execution des fonction de mis a jour
VanneMSI MSI : Stability Reached....
Running MSI : ControlerNewMSI pour l'objet : toto.impl.ControlerImpl@bdd996
ControlerNewMSI incrementation de temps externe (6 ut)
ControlerNewMSI IE : lecture des entrees
ControlerNewMSI IE : calcul des vars propositionnels
ControlerNewMSI initialisation T interne
ControlerNewMSI PC : etat courant (avant CEF-ARS) : SOF
ControlerNewMSI incremter T interne
ControlerNewMSI PC : CEF      State before CEF : OF
ControlerNewMSI State after CEF : OF
ControlerNewMSI IS : execution des fonction de mis a jour
ControlerNewMSI MSI : Stability Reached....
Running MSI : ReservoirMSI pour l'objet : toto.impl.ReservoirImpl@b6561b (niveauEau: 7.3, niveauEauMax: 10.0, niveauEauMin: 1.0)
ReservoirMSI incrementation de temps externe (6 ut)
ReservoirMSI IE : lecture des entrees
ReservoirMSI IE : calcul des vars propositionnels
ReservoirMSI initialisation T interne
ReservoirMSI PC : etat courant (avant CEF-ARS) : Ss1
ReservoirMSI incremter T interne
ReservoirMSI PC : CEF      State before CEF : s1
ReservoirMSI State after CEF : s1
ReservoirMSI IS : execution des fonction de mis a jour
ReservoirMSI MSI : Stability Reached....
Running MSI : VanneMSI pour l'objet : toto.impl.VanneImpl@105dcab (debit: 2.5, debitMax: 3.0, croissanceDebit: 0.5, decroissanceDebit: 0.5)
VanneMSI incrementation de temps externe (6 ut)
VanneMSI IE : lecture des entrees
VanneMSI IE : calcul des vars propositionnels
VanneMSI initialisation T interne
VanneMSI PC : etat courant (avant CEF-ARS) : Sopening
VanneMSI incremter T interne
VanneMSI PC : CEF      State before CEF : opening
VanneMSI State after CEF : opening
VanneMSI IS : execution des fonction de mis a jour
VanneMSI MSI : Stability Reached....
Printing TN
Printing the content of the Black Table
org.eclipse.emf.ecore.impl.BlackTableRowImpl@6fe482 (timeOfAdding: 5, varName: debit, varType: EDouble, varValue: 0.0, recivers: [toto.impl.ReservoirImpl@b6561b (niveauEau: 9.8, niveauEauMax: 10.0, niveauEauMin: 1.0), toto.impl.V
org.eclipse.emf.ecore.impl.BlackTableRowImpl@1bd0006 (timeOfAdding: 5, varName: open, varType: Event, varValue: true, recivers: [toto.impl.VanneImpl@105dcab (debit: 3.0, debitMax: 3.0, croissanceDebit: 0.5, decroissanceDebit: 0.5
org.eclipse.emf.ecore.impl.BlackTableRowImpl@1d05b15 (timeOfAdding: 5, varName: close, varType: Event, varValue: true, recivers: [toto.impl.VanneImpl@194a966 (debit: 0.0, debitMax: 2.0, croissanceDebit: 0.2, decroissanceDebit: 0.2
org.eclipse.emf.ecore.impl.BlackTableRowImpl@4fc235 (timeOfAdding: 5, varName: niveauEau, varType: EDouble, varValue: 7.3, recivers: [toto.impl.ControlerImpl@bdd996], sender: toto.impl.ReservoirImpl@b6561b (niveauEau: 9.8, niveau
```


Application case: EFFBD and PBD

Reminder: enhanced Functional Flows Block Diagram (eFFBD)



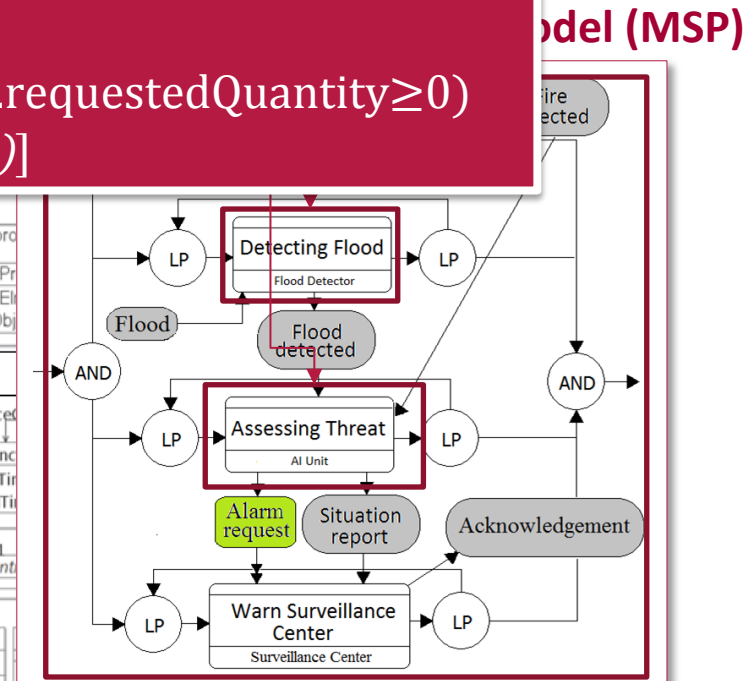
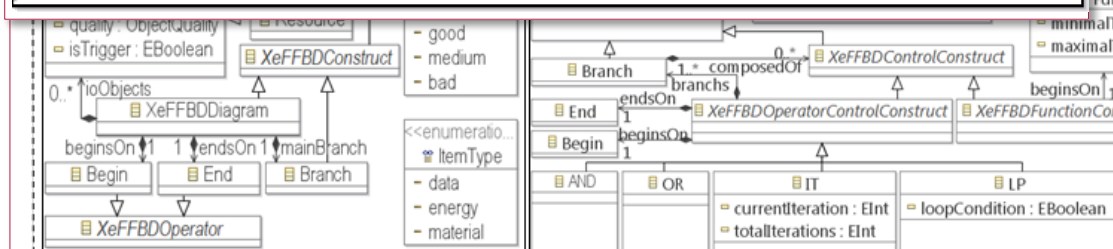
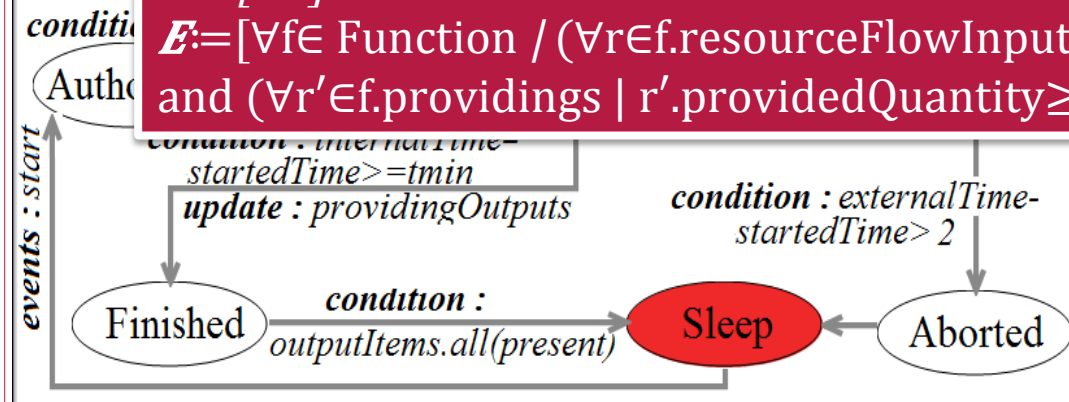
Structural Property: “The quantity of a Resource that a function provides or requires must be positive or zero”

Formal specification using UPSL-SE:

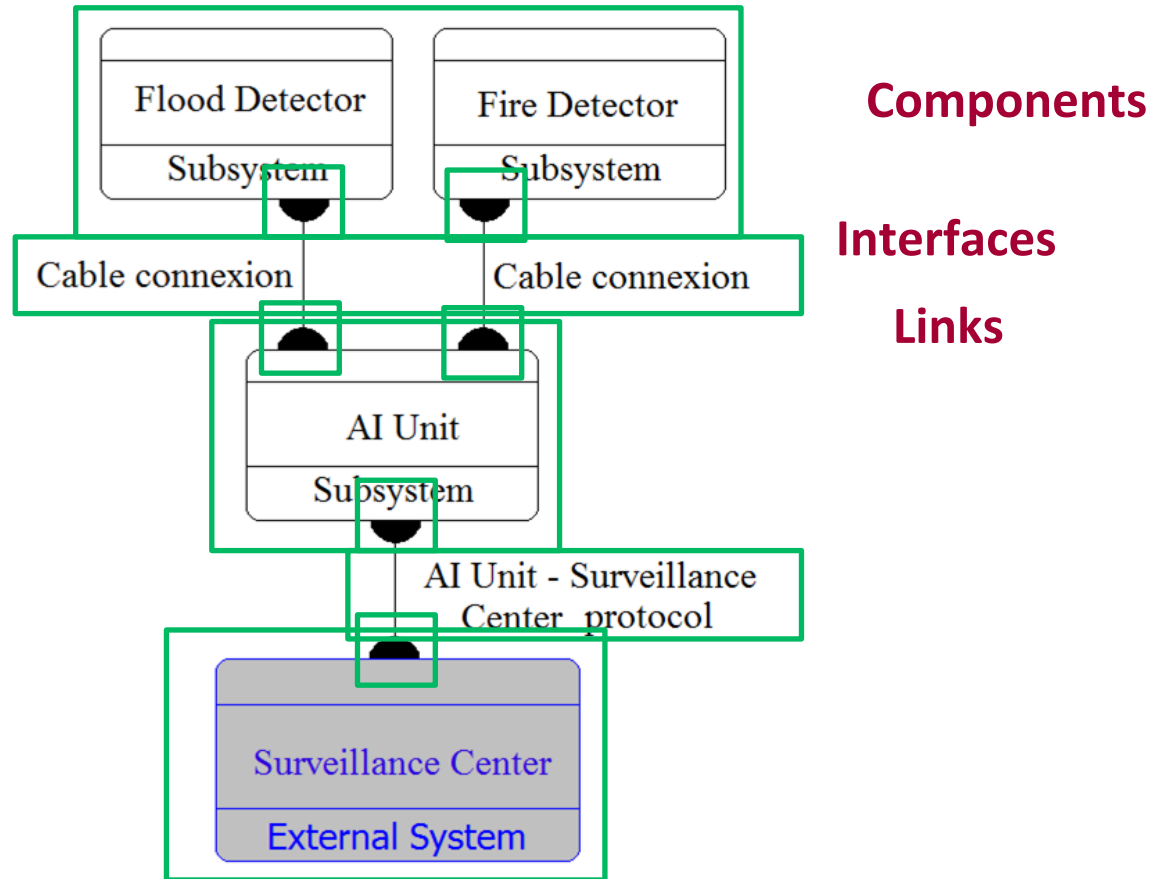
$C := [T]$

$R := [\Rightarrow]$

$E := [\forall f \in \text{Function} / (\forall r \in f.\text{resourceFlowInput} / r.\text{requestedQuantity} \geq 0) \text{ and } (\forall r' \in f.\text{providings} \mid r'.\text{providedQuantity} \geq 0)]$



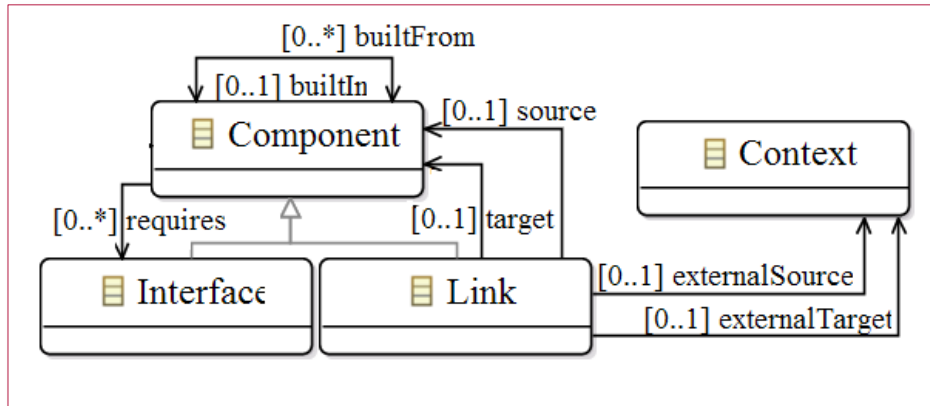
Sum up and application to Physical Block Diagram (PBD)



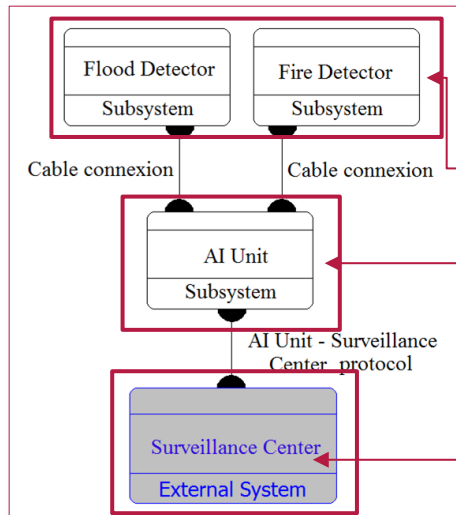
Model of architectural composition (the physical components of a system)

Example: Physical Block Diagram (PBD)

PBD abstract syntax (SP)

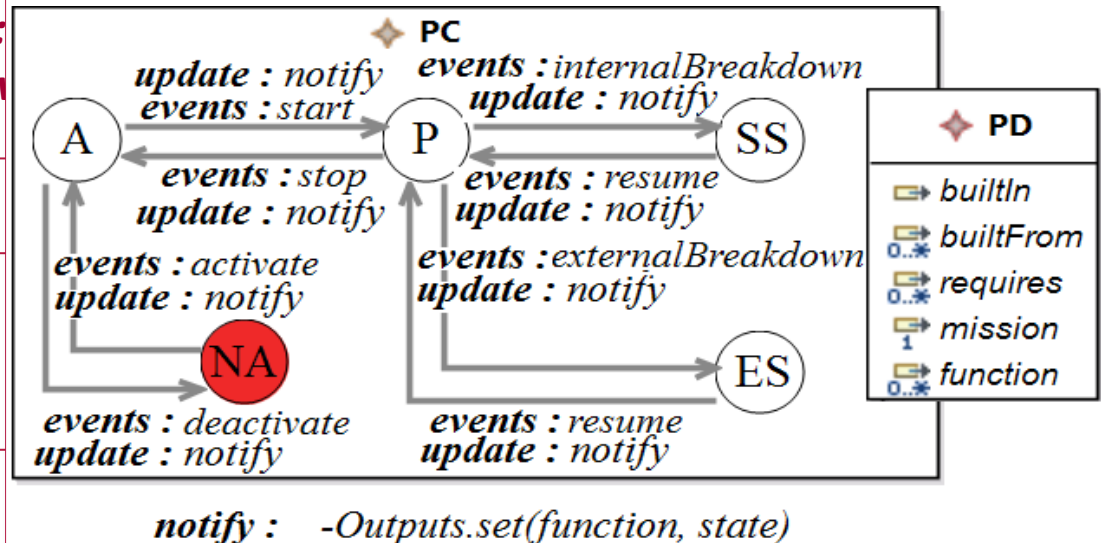


PBD model (MSP)



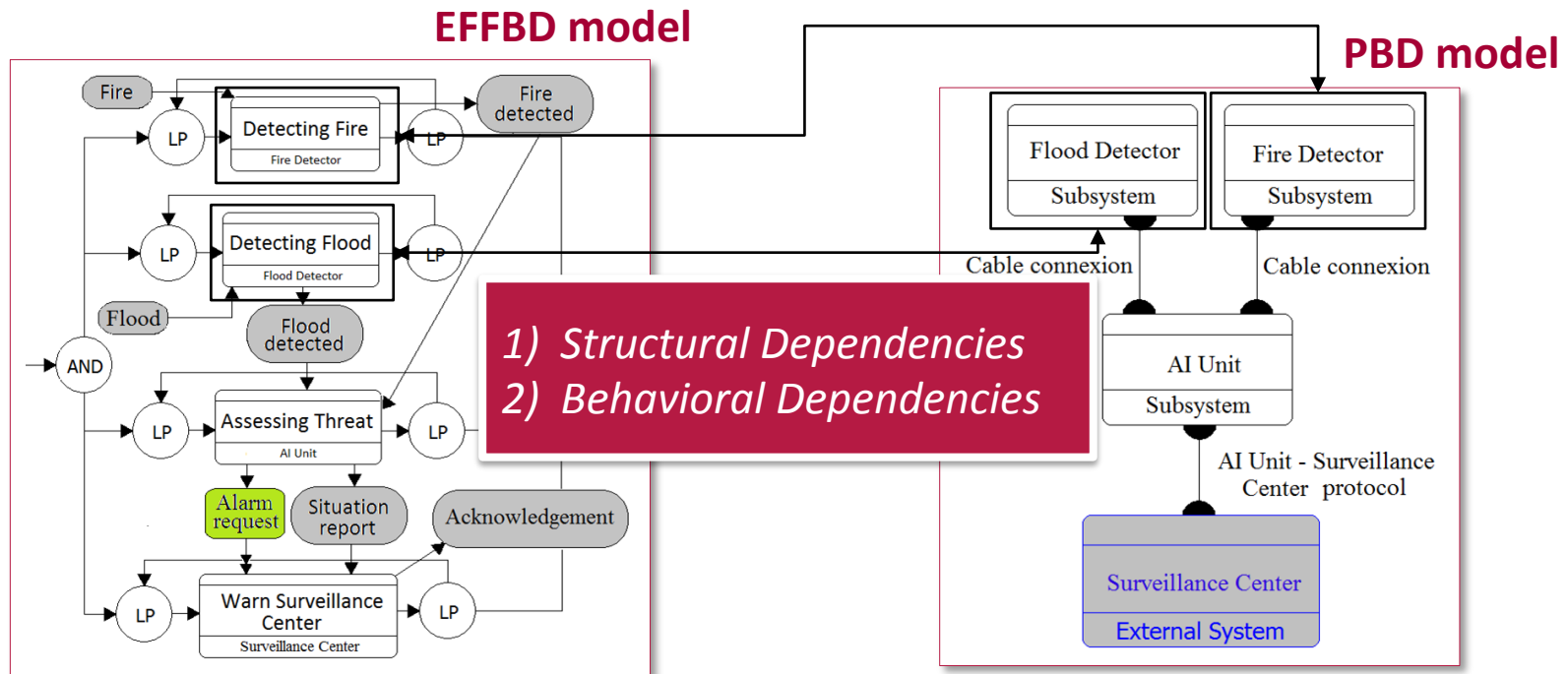
**Request:
behav**

PBD BP (for Component)



Dependencies between eFFBD and PBD

*“Physical components of a system **perform** one or more functions and functions are **allocated** to a component”*



Dependencies between eFFBD and PBD

Constraint:

"If a component enters a breakdown state (internal or external), its functions must suspend execution"

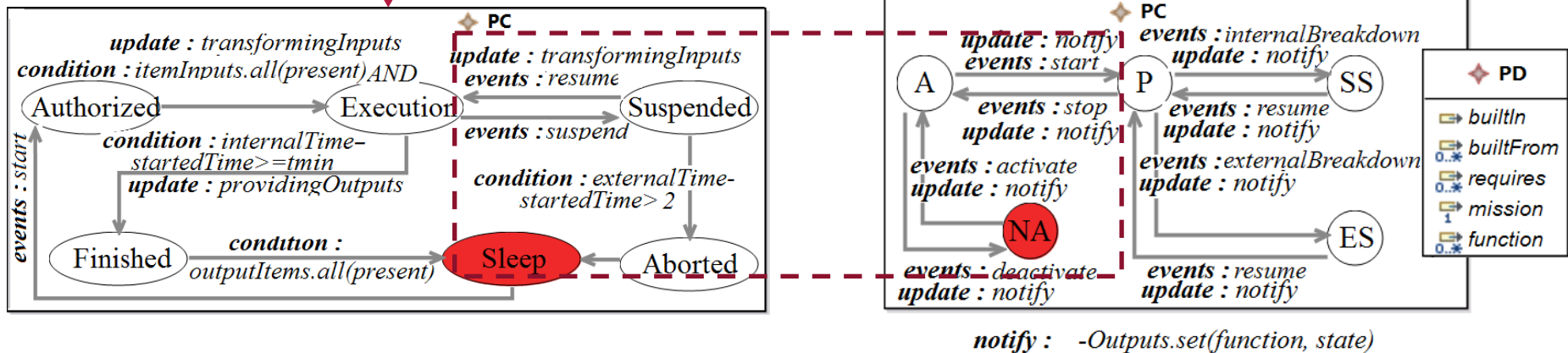
Formal specification using UPSL-SE:

$C := [\forall c \in \text{Component} \mid c.\text{State} = \text{SS} \text{ OR } c.\text{State} = \text{ES}]$

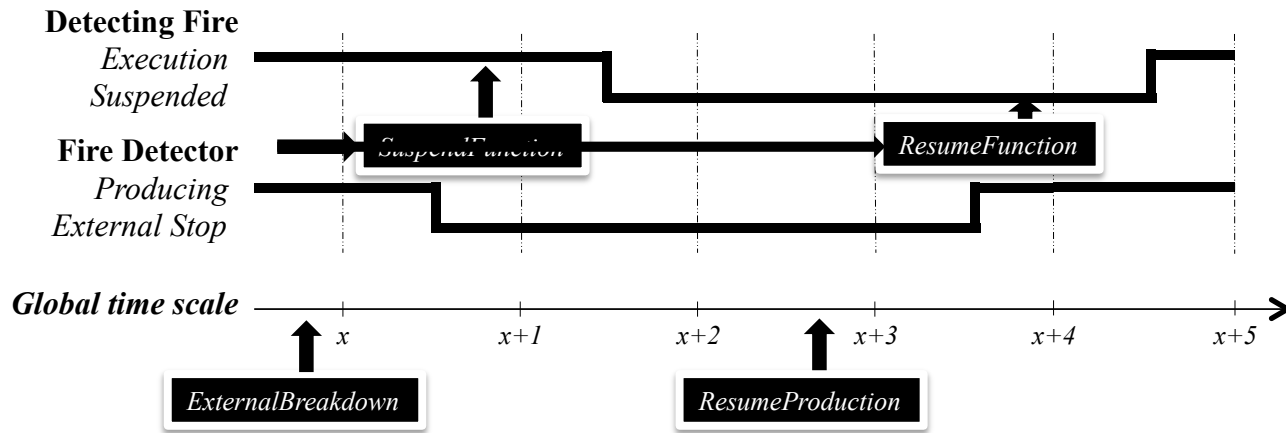
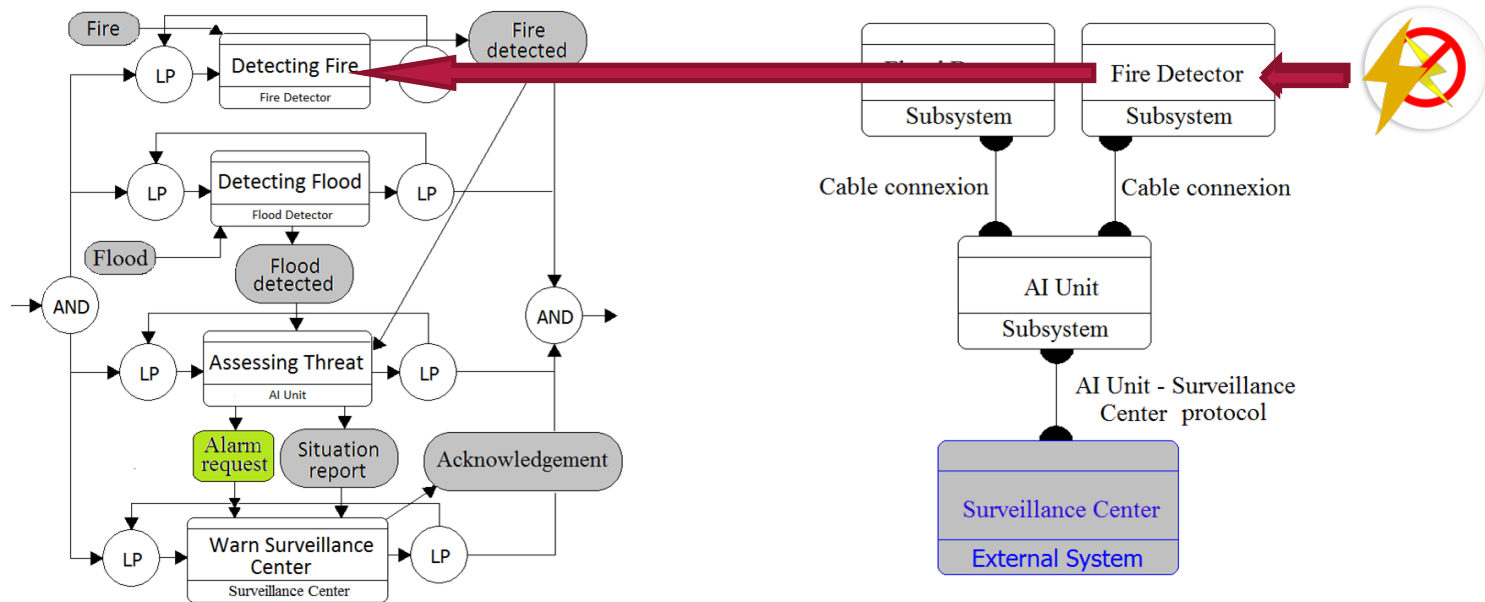
$R := [\Rightarrow]$

$E := [o(\forall f \in c.\text{performs} \mid f.\text{State} \neq \text{Execution})]$

Behavioral dependencies



Simulation of eFFBD and PBD



Conclusion and Perspectives

Research focus

Modelling Complex Systems

Contributions

A lifecycle process for modeling, verification and validation

An approach for the design of DSMLs and models

Simulation and Formal verification of properties

Tool support

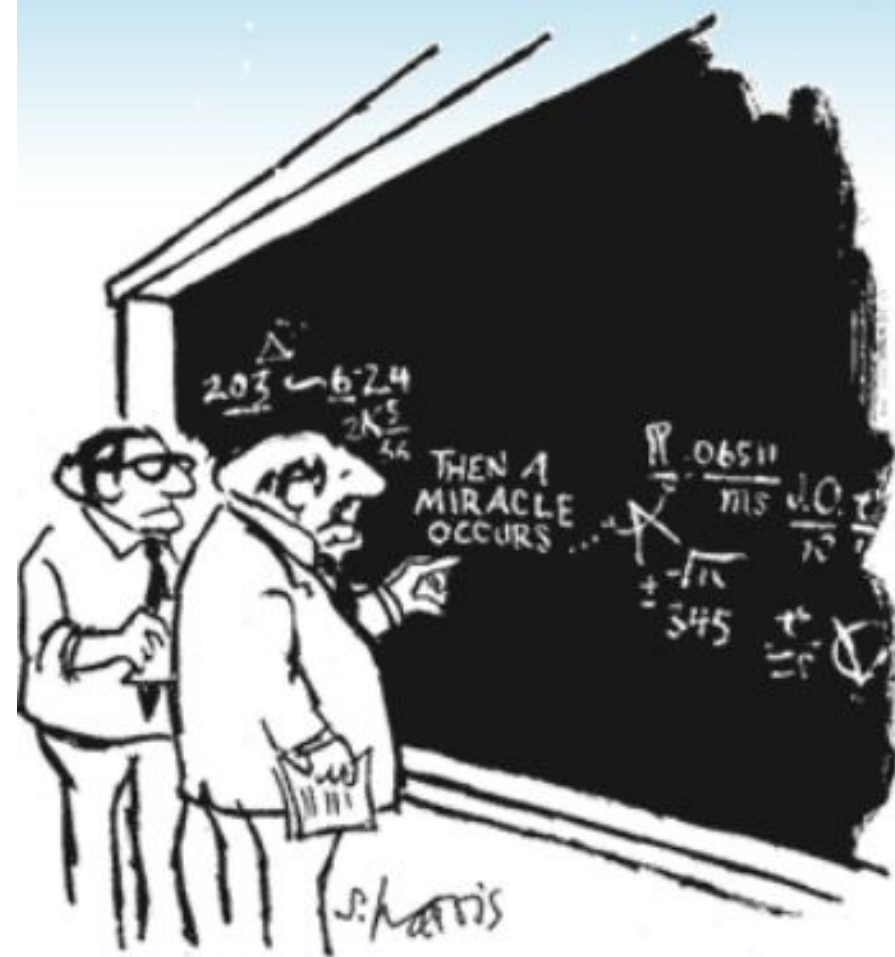
Application case: eFFBD and PBD

Future works

Continuous and Hybrid behavioral models

**Thank you
for your attention**

Questions?



"I THINK YOU SHOULD BE MORE
EXPLICIT HERE IN STEP TWO."