2. SYSTEMES INDUSTRIELS COMPLEXES

Interoperability requirements for collaborative process analysis

Centre de Recherche
LGI2P

Auteurs

Nicolas Daclin
Sihem Mallek
Vincent Chapurlat

Interoperable Systems and Organization Engineering

Partenaires

Université Bordeaux 1

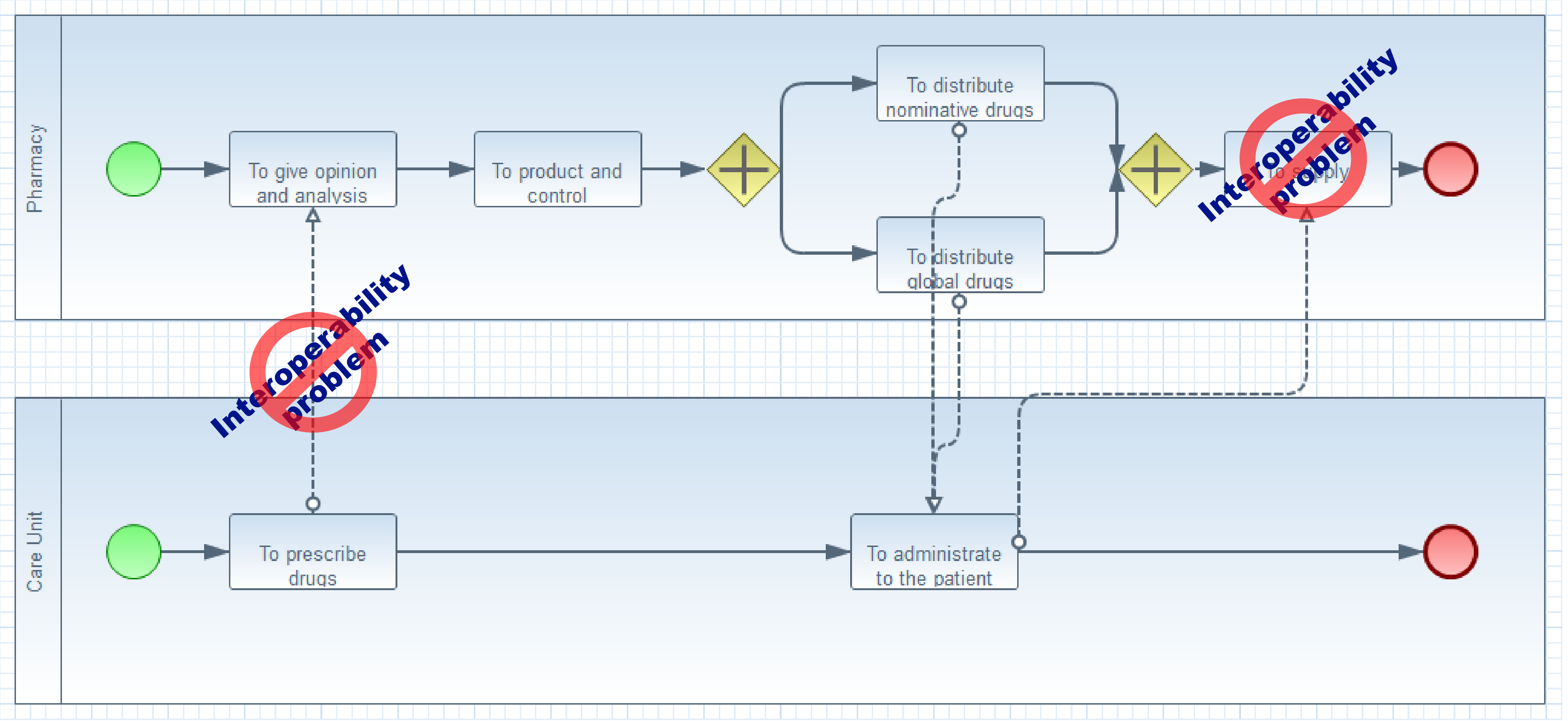
Overall context

- To **verify interoperabilit**y requirements within **collaborative processes** models using **formal verfication techniques** to detect and anticipate interoperability problems.

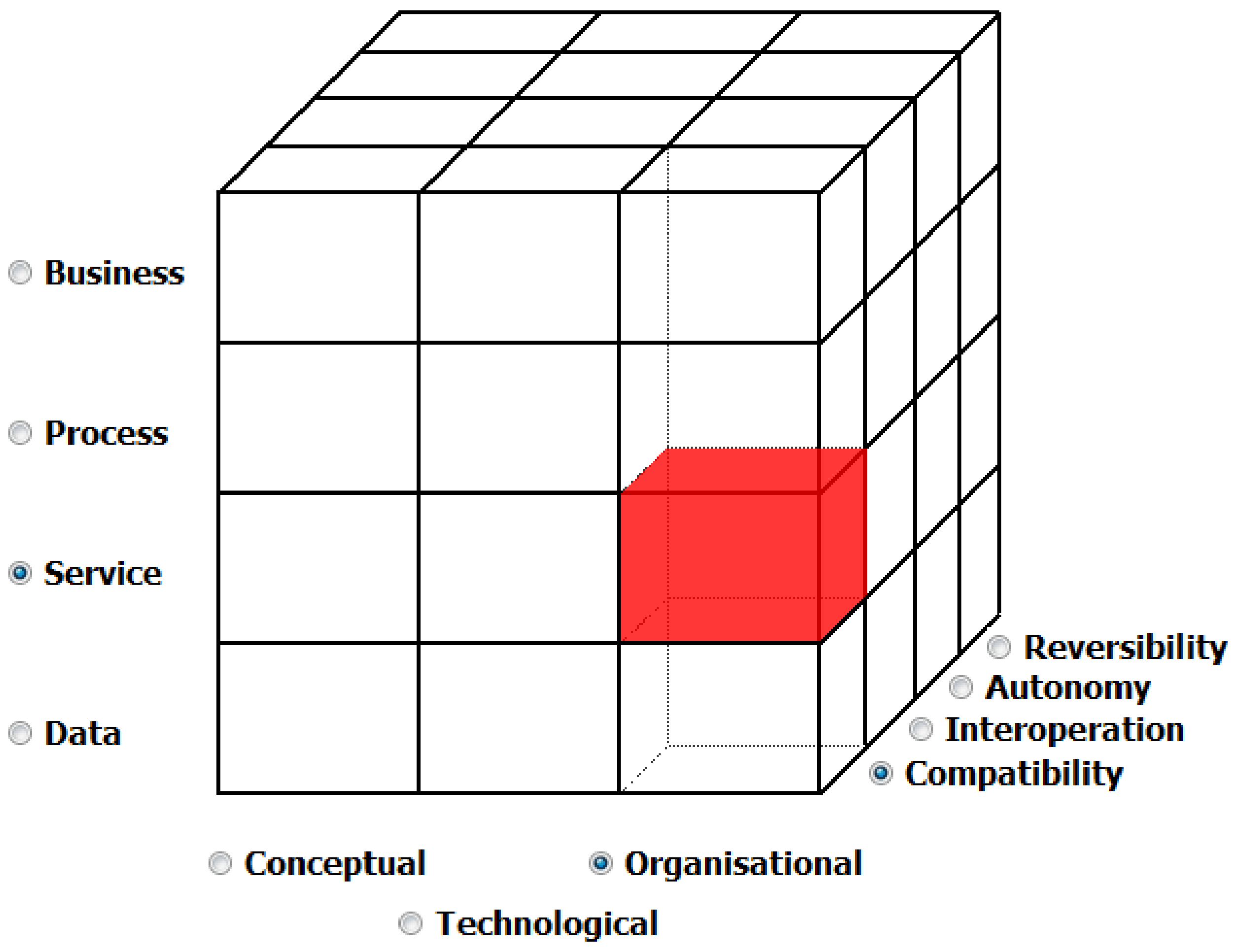
Problematic

- How to guide and facilitate end-users to **select** their own interoperability requirements?
- How to allow end-users to **write** their own interoperability requirements?
- How to ensure that interoperability requirements are **well written**?

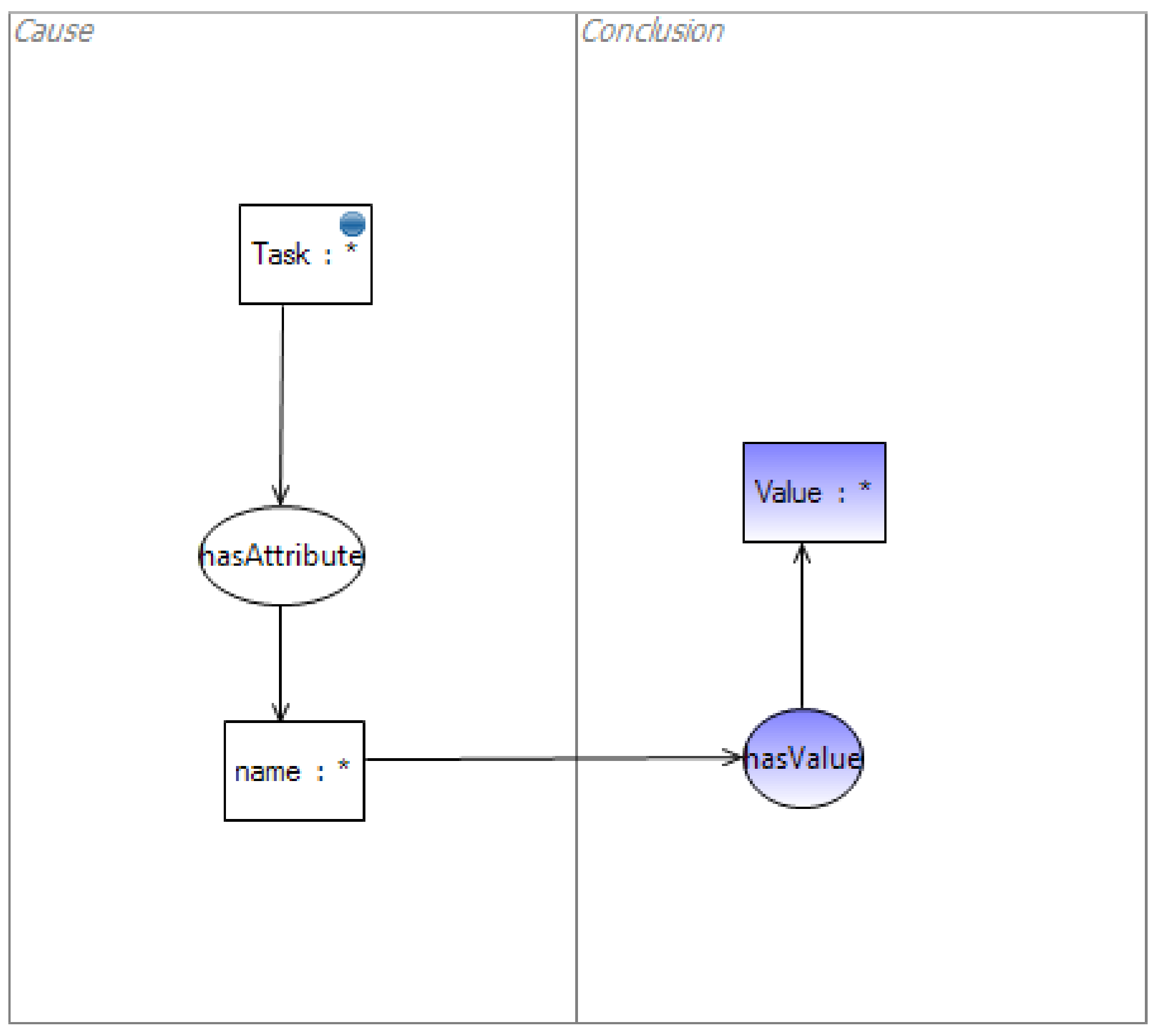
Collaborative process model (BPMN 2.0)



Interoperability requirements framework



Conceptual Graph



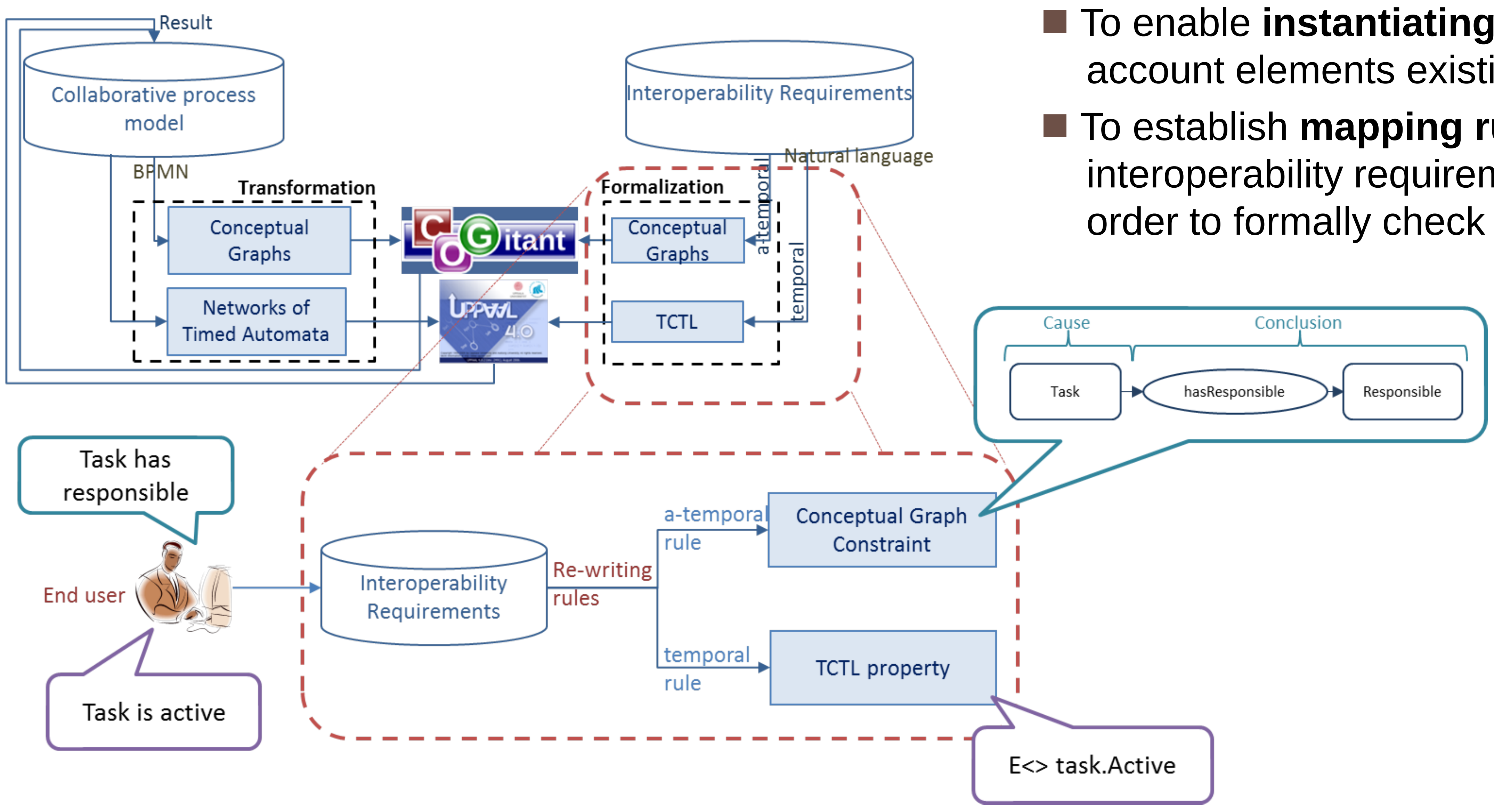
TCTL

E<> Task.Working and T>5 and T<10

A<> Resource.Active and Resource.T<10

Interoperability requirements writing

- To allow the selection of predefined interoperability requirements consistently positioned in a **human readable repository**.
- To enable **instantiating selected requirements** taking into account elements existing in the collaborative process model.
- To establish **mapping rules** to re-write correctly these interoperability requirements into TCTL and Conceptual Graphs in order to formally check them.



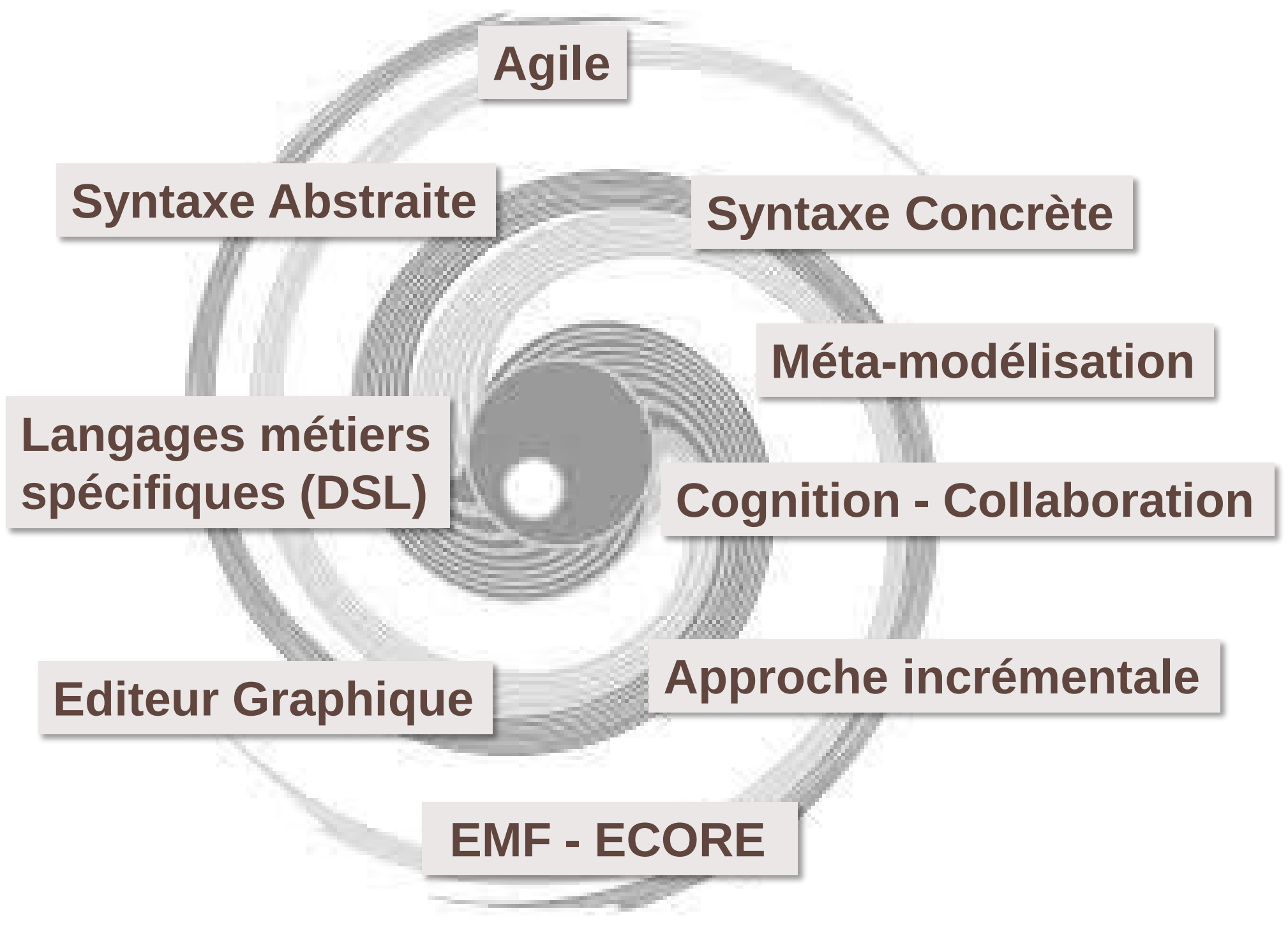
Future works

- To enable end-users to write directly their own interoperability requirements with a dedicated Domain Specific Language.
- To propose interoperability solutions relative to the identification of not checked requirements.

Colloque IMT, Numérique: grande échelle et complexité
Mars 2014

Contact nicolas.daclin@mines-ales.fr
Vincent.chapurlat@mines-ales.fr
Site web http://lgi2p.mines-ales.fr

Ingénierie des langages de modélisation: syntaxes concrètes graphiques avec DIAGRAPH


Agile
Syntaxe Abstraite
Syntaxe Concrète
Méta-modélisation
Langages métiers spécifiques (DSL)
Cognition - Collaboration
Approche incrémentale
Editeur Graphique
EMF - ECORE

Parties prenantes

Auteurs

François Pfister
Vincent Chapurlat

LGI2P, Ecole des Mines d'Alès, site de Nîmes, Parc Scientifique G. Besse, 30000 Nîmes

Partenaires

Marianne Huchard
Clémentine Nebut

LIRMM, CNRS – Université Montpellier 2, 161 rue Ada, 34095 Montpellier Cedex 5, France

Problématique

Comment définir aisément des notations graphiques en IDM (Ingénierie Dirigée par les Modèles) ?

Modéliser des systèmes sociotechniques complexes nécessite des langages :

- UML, DSML, ontologies, simulation multi-physique, etc..

Pour exprimer des points de vue :

- statique, dynamique, fonctionnel, architectural, exigences, etc…

Nous nous intéressons aux DSML (Domain Specific Modeling Language) graphiques en proposant dans ce contexte une méthodologie outillée : DIAGRAPH, consistant en :

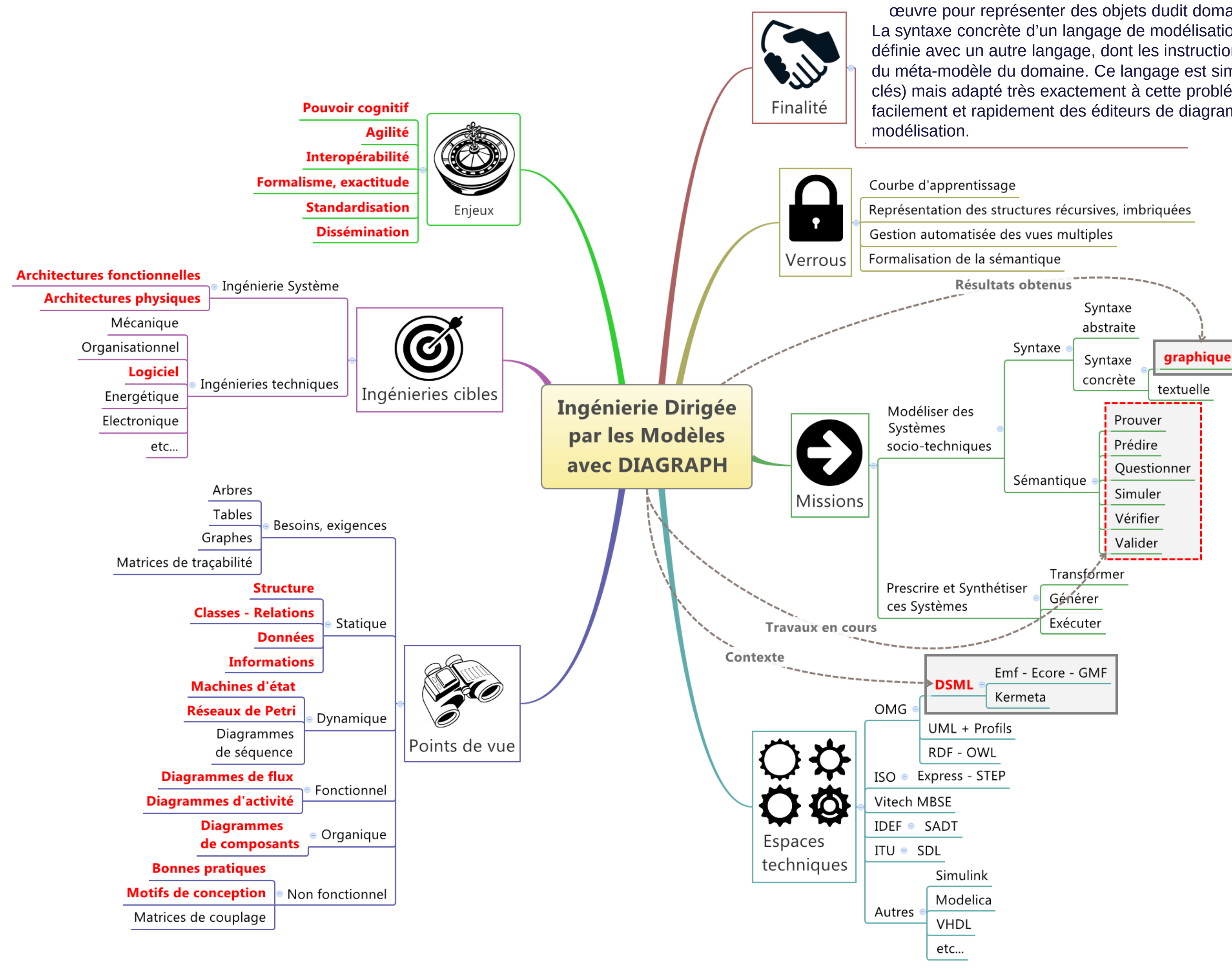
- Un processus de définition du langage, faisant actuellement défaut. (Les pratiques sont centrées sur la conception de méta-modèles, mais la création d'éditeurs graphiques est un point difficile).
- Un outillage compatible avec les environnements déjà existants dans l'écosystème Eclipse/EMF.

Contexte

Diagraph est un environnement de méta-modélisation permettant de créer des langages de modélisation graphiques adaptés à des domaines spécifiques (DSML, Domain Specific Modeling Languages).
La particularité de cet environnement est de traiter en ***un artefact unique***

- la « physique du domaine » avec un méta-modèle.
- la définition d'une syntaxe graphique (diagramme) qui sera mise en œuvre pour représenter des objets dudit domaine.

La syntaxe concrète d'un langage de modélisation graphique est alors définie avec un autre langage, dont les instructions annotent les éléments du méta-modèle du domaine. Ce langage est simple (moins de 20 mots-clés) mais adapté très exactement à cette problématique : créer facilement et rapidement des éditeurs de diagramme pour tout langage de modélisation.



Ingénierie Dirigée par les Modèles avec DIAGRAPH

- Finalité
- Enjeux
  - **Pouvoir cognitif**
  - **Agilité**
  - **Interopérabilité**
  - **Formalisme, exactitude**
  - **Standardisation**
  - **Dissémination**
- Ingénieries cibles
  - Ingénierie Système
    - **Architectures fonctionnelles**
    - **Architectures physiques**
  - Ingénieries techniques
    - Mécanique
    - Organisationnel
    - **Logiciel**
    - Energétique
    - Electronique
    - etc…
- Points de vue
  - Besoins, exigences
    - Arbres
    - Tables
    - Graphes
    - Matrices de traçabilité
  - Statique
    - **Structure**
    - **Classes - Relations**
    - **Données**
    - **Informations**
  - Dynamique
    - **Machines d'état**
    - **Réseaux de Petri**
    - Diagrammes de séquence
  - Fonctionnel
    - **Diagrammes de flux**
    - **Diagrammes d'activité**
  - Organique
    - **Diagrammes de composants**
  - Non fonctionnel
    - **Bonnes pratiques**
    - **Motifs de conception**
    - Matrices de couplage
- Verrous
  - Courbe d'apprentissage
  - Représentation des structures récursives, imbriquées
  - Gestion automatisée des vues multiples
  - Formalisation de la sémantique
- Missions
  - Modéliser des Systèmes socio-techniques
    - Syntaxe
      - Syntaxe abstraite
      - Syntaxe concrète
        - **graphique**
        - textuelle
    - Sémantique
      - Prouver
      - Prédire
      - Questionner
      - Simuler
      - Vérifier
      - Valider
  - Prescrire et Synthétiser ces Systèmes
    - Transformer
    - Générer
    - Exécuter
- Espaces techniques
  - OMG
    - **DSML**
      - Emf - Ecore - GMF
      - Kermeta
    - UML + Profils
    - RDF - OWL
  - ISO
    - Express - STEP
  - Vitech MBSE
  - IDEF
    - SADT
  - ITU
    - SDL
  - Autres
    - Simulink
    - Modelica
    - VHDL
    - etc…

Résultats obtenus
Travaux en cours
Contexte

Contributions

Id	Aspect	Description
1	open-source	Publié avec une licence libre et gratuite.
2	intégration des standards	Intégré dans l'écosystème Emf/Ecore.
3	interaction utilisateur	Editeur de diagramme de classes étendu permettant la saisie des rôles graphiques pour les entités du méta-modèle.
4	indépendant de la plateforme	Les diagrammes cibles sont abstraits et indépendants d'une plateforme spécifique.
5	abstraction	La structure d'un diagramme est définie par un méta-modèle publié.
6	artefact grammatical	La syntaxe concrète graphique et la syntaxe abstraite sont définies dans un artefact unique (une grammaire)
7	composition	Les langages peuvent être composés grâce à un mécanisme de fusion.
8	navigation entre les vues	Un mécanisme de vues multiples, incluant la navigation entre les vues, fait partie intégrante de la syntaxe concrète.
9	support des hiérarchies	Un support natif des hiérarchies est fourni.
10	grammaire positionnelle	La partie positionnelle de la grammaire est indépendante de la partie stylistique.
11	sémantique statique basée sur des règles	La sémantique statique est supportée par un langage de type OCL.
12	héritage graphique	L'héritage graphique est implémenté (propagation automatique des compositions graphiques).
13	inférence de constructions	Des constructions basées sur des motifs sont inférées

Un langage pour décrire des diagrammes

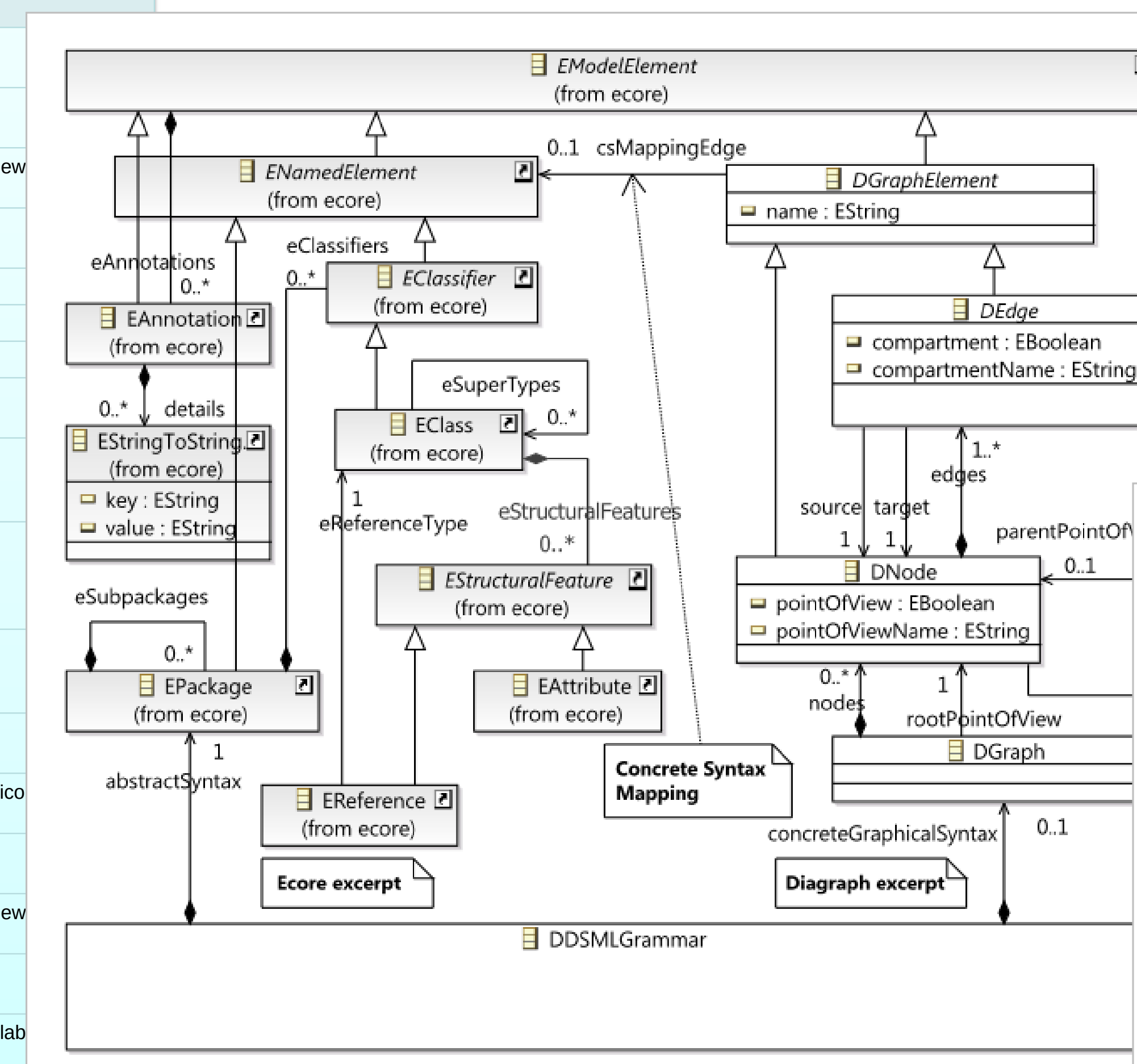
Les mots-clés

(sa syntaxe concrète)

Mot-clé	Sémantique	Correspondance méta-modèle
node	Une figure graphique (polygone, ellipse?) qui joue le rôle de la classe associée.	DNode
link	Une ligne vers la classe cible (notée sur la classe associante).	DEdge
pov	Un diagramme (et non une figure) joue le rôle de la classe.	DNode.pointOfView
lnk	Identique à link, mais notée sur la classe source.	DEdge
lsrc	Définit la source d'un link.	DEdge.source
ltrg	Définit la cible d'un link.	DEdge.target
cont	Définit le conteneur d'un link.	DNode.edges
cref	Un compartiment graphique contenant le nœud cible.	DEdge.source
kref	Identique à cref, mais plusieurs compartiments sont créés, à raison d'un compartiment par type d'association	DEdge.source
ref	Une ligne vers la classe cible. Référence associante (par conséquent, pas d'étiquette possible)	DEdge
afx	Définit une ligne affixée à travers un compartiment adjacent externe au nœud source (paradigme du port).	DEdge.source
shape	Définit une figure pour le nœud (node) ou une forme de flèche (pour link ou ref).	DShape
icon	Définit une icône associée au nœud (node) ou à la ligne (link ou ref).	DGraphElement.ico
ordered	Définit la mise en page (empilée ou libre) dans un compartiment.	DNode.layout
view	Identifie le diagramme propriétaire du nœud (dans le cas multi-vues).	DNode.pointOfView
nav	Un hyperlien de navigation vers un autre diagramme (une autre vue).	DEdge.target
label	Une étiquette graphique décorant le nœud (node) ou la ligne (link). Joue le rôle de l'attribut de la classe indiqué par l'argument.	DGraphElement.lab
style	Référence vers un style déclaré dans le modèle stylistique.	DBaseStyle (méta-modèle Diastyle)

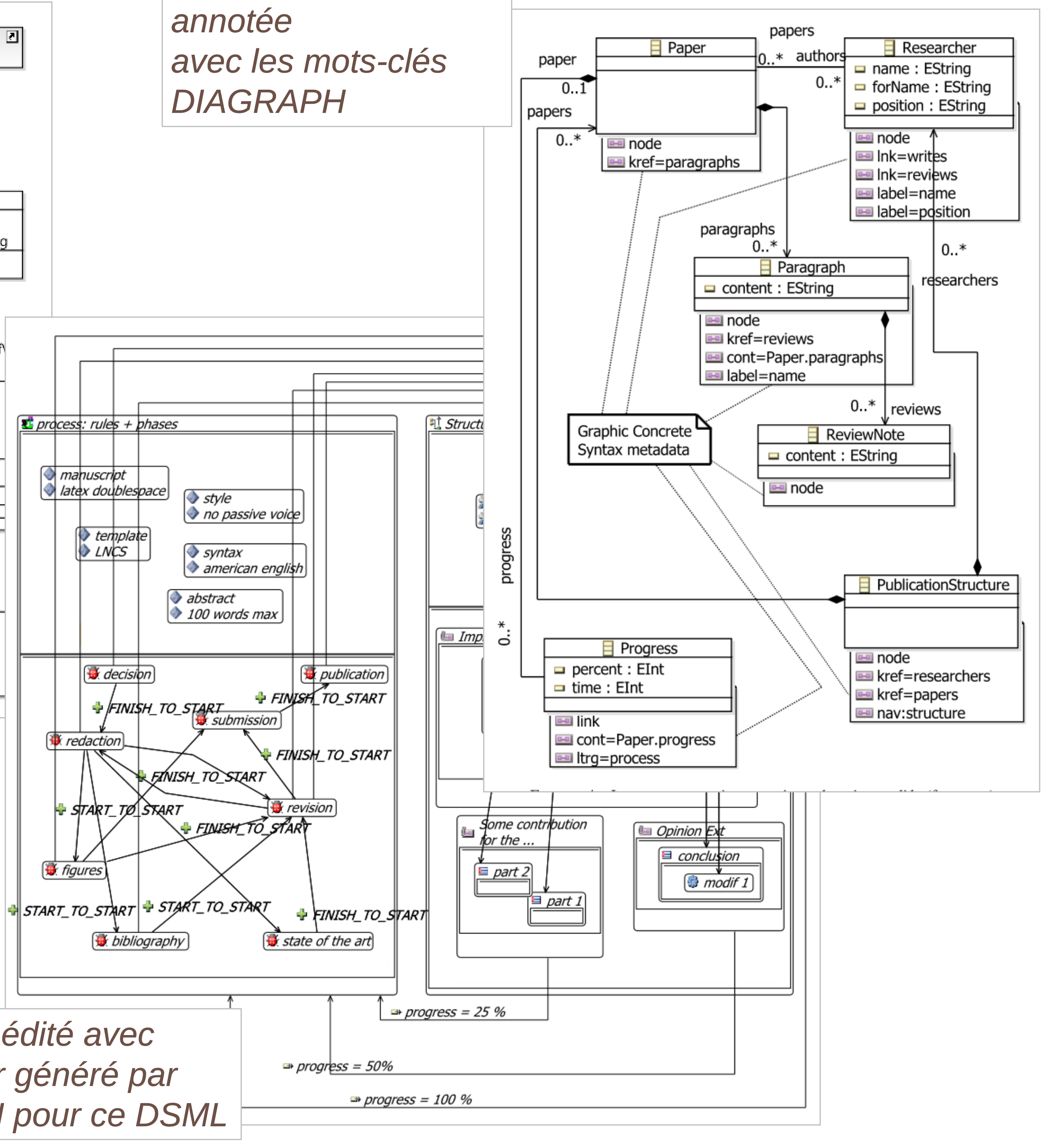
Le méta-modèle

(sa syntaxe abstraite)



Un DSML obtenu avec DIAGRAPH

sa syntaxe abstraite annotée avec les mots-clés DIAGRAPH



Un modèle édité avec le modeleur généré par DIAGRAPH pour ce DSML

Mise à disposition de l'environnement DIAGRAPH : Logiciel libre (Licence Eclipse EPL)
https://code.google.com/p/diagraph/

Colloque IMT : « Numérique : Grande échelle et complexité »
Mars 2014

Contact francois.pfister@mines-ales.fr Site web: http://lgi2p.mines-ales.fr

Guide pour la modélisation et la vérification de la cohérence modes de fonctionnement / scénarios opérationnels en Ingénierie Système : **GEMOS**

Parties prenantes

Laboratoire de Génie Informatique et d'Ingénierie de Production

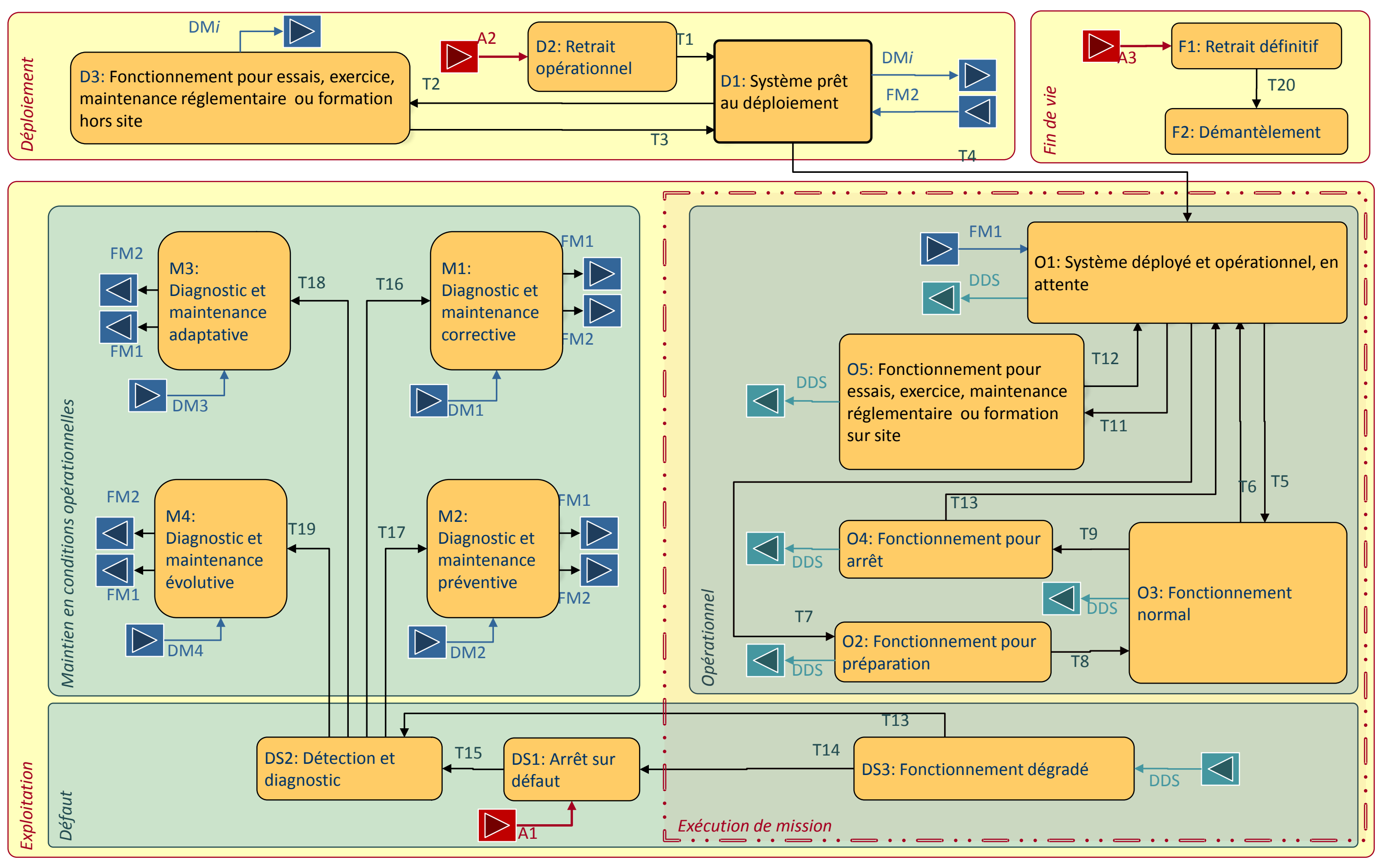
Auteurs

Chapurlat Vincent
Daclin Nicolas

Contexte

Problématique

- Guider la modélisation des modes de fonctionnement d'un système complexe en cours de conception
- Aider à l'émergence d'une ou plusieurs architectures fonctionnelles correspondantes
- S'assurer de la cohérence de l'ensemble



Objectifs : un guide outillé, le GEMOS…

- Définir un langage de modélisation du comportement d'un système liant naturellement les modes de fonctionnement, les scénarios opérationnels et l'architecture fonctionnelle résultante
- Gagner du temps de modélisation par réutilisation d'un modèle de comportement de référence applicables à tous les systèmes
- Gagner en confiance par une vérification assistée de la qualité du modèle du système en tenant compte de divers types d'exigences
- Disposer d'un outillage support

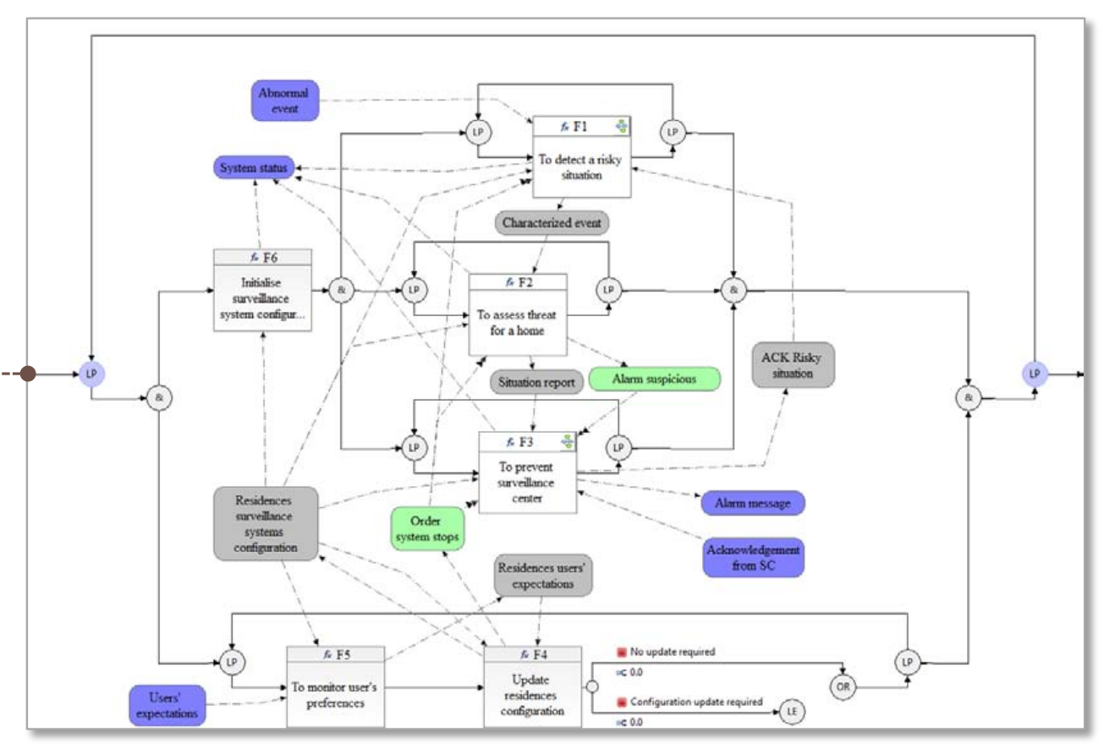
Modélisation

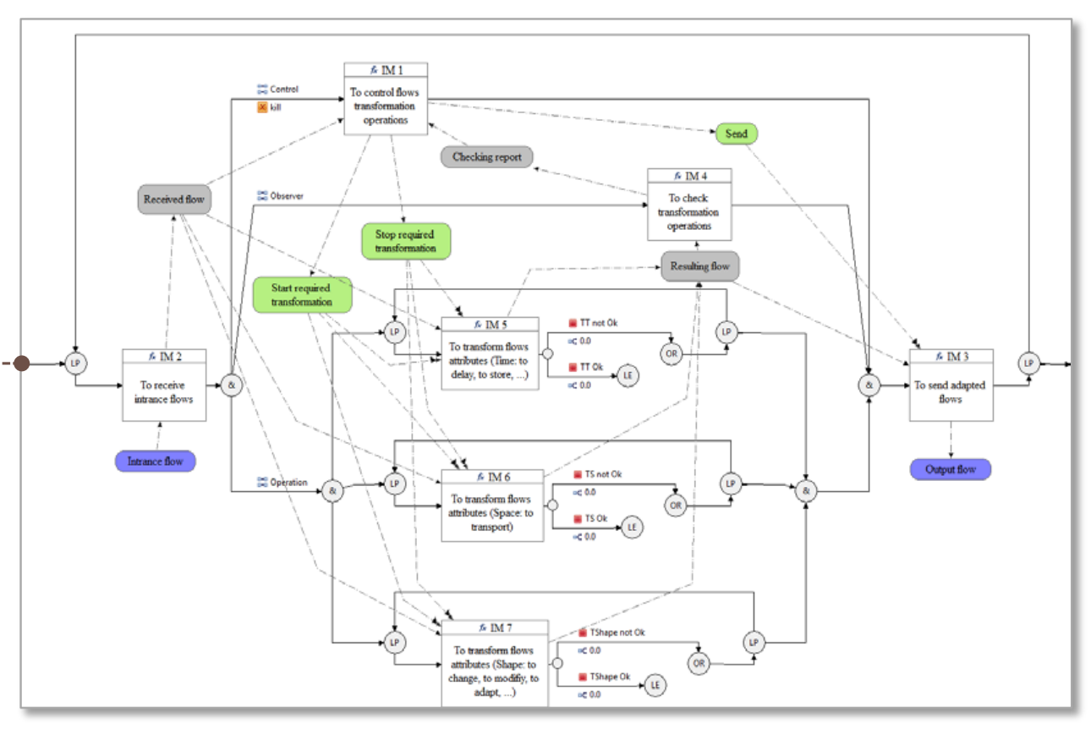
Phases génériques / *Modes génériques*

System deployment		D1	System is ready and waiting for deployment
		D2	Operational retirement
		D3	System functions for tests, maintainance, or traning out of operational site
System exploitation	Operationnal	O1	System is deployed and operational on site,
		O2	Preparing the system to assume its mission in nominal mode
		O3	System functions in nominal mode
		O4	Preparing the system to end normally its mission
		O5	System functions for tests, maintainance, or traning on operational site
	Dysfunctioning / security	DS1	Stop after a default or dysfunction
		DS2	Diagnosis for default detection
		DS3	System functions in non-nominal mode
	Maintainance	M1	Diagnosis and Corrective Maintenance
		M2	Diagnosis and Preventive Maintenance
		M3	Diagnosis and Adaptative Maintenance
		M4	Diagnosis and Evolutive Maintenance
Cessation and dismantling		C1	Retract
		C2	Dismantling

■ **Principe** : passer du **modèle comportemental de S** = {Phases}…

↱	D1	D2	D3	O1	O2	O3	O4	O5	DS1	DS2	DS3	M1	M2	M3	M4	F1	F2
D1		A2	T2	T4					A1			DM1	DM2	DM3	DM4	A3	
D2	T1								A1			DM1	DM2	DM3	DM4	A3	
D3	T3	A2							A1							A3	
O1		A2			T7	T5		T11	A1		DDS					A3	
O2		A2				T8			A1		DDS					A3	
O3		A2		T6			T9		A1		DDS					A3	
O4		A2		T10					A1		DDS					A3	
O5		A2		T12					A1		DDS					A3	
DS1		A2								T15						A3	
DS2		A2							A1			T16	T17	T18	T19	A3	
DS3		A2							T14	T13						A3	
M1	FM2	A2		FM1					A1							A3	
M2	FM2	A2		FM1					A1							A3	
M3	FM2	A2		FM1					A1							A3	
M4	FM2	A2		FM1					A1							A3	
F1																	T20
F2																	





Transition = {event, condition, action}

requires / induces / changes on

Phase = {sous-phases, **modes**, **transitions**}

Scenario Opérationnel = {**fonctions**, **flux**}

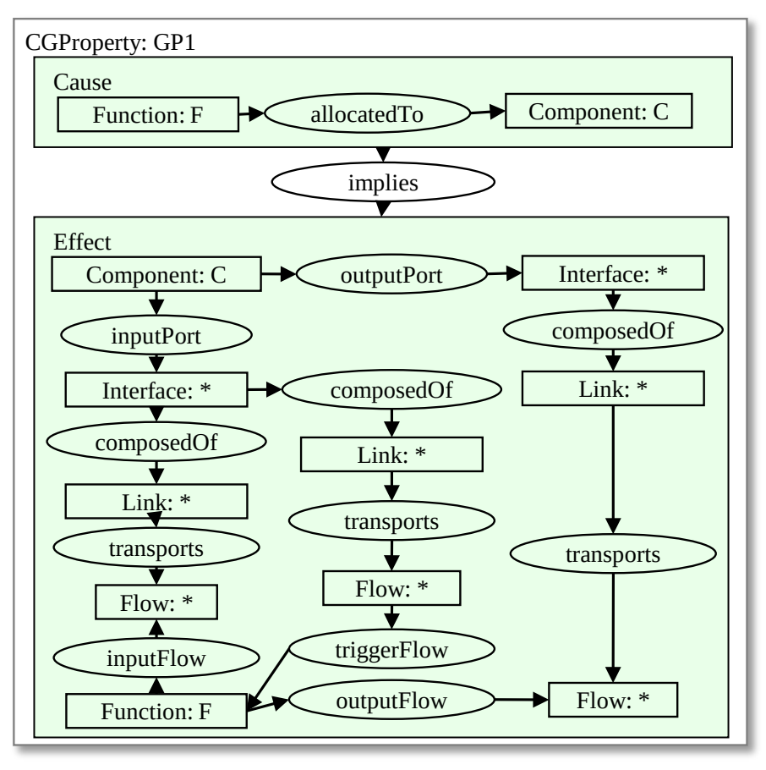
Mode = {sous-modes, sous-transitions, **configurations**, **transitions**, **scenarios**}

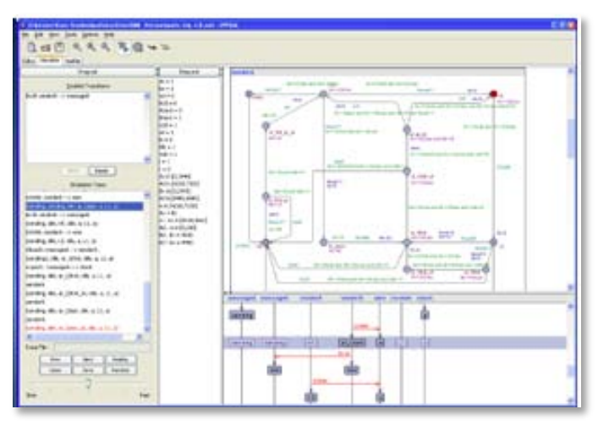
requires / induces / changes on

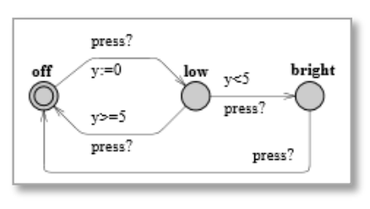
Configuration = {sous-configurations, sous-transitions}

… à une ou plusieurs alternatives d'**Architectures fonctionnelles de S**

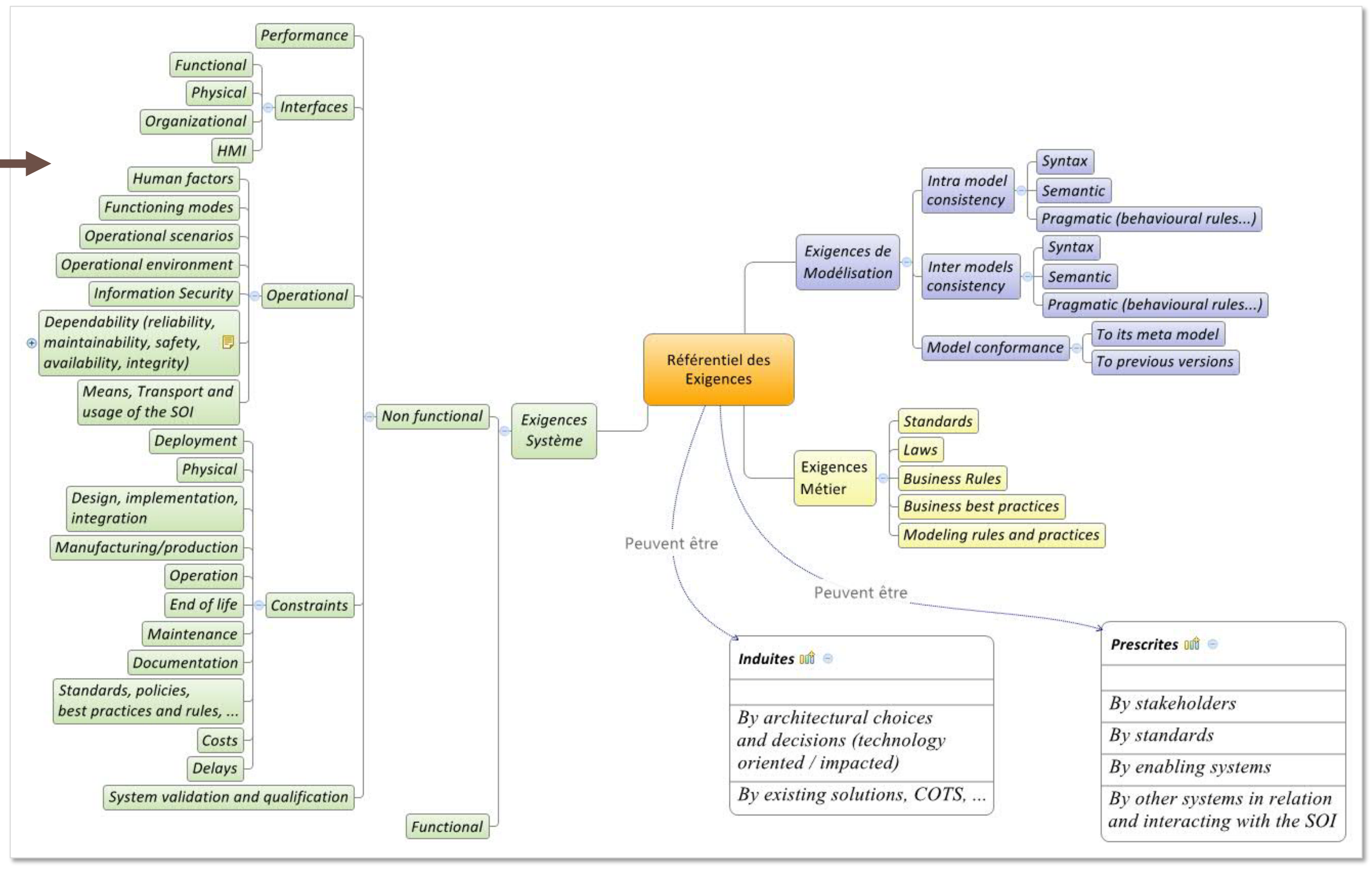
Vérification







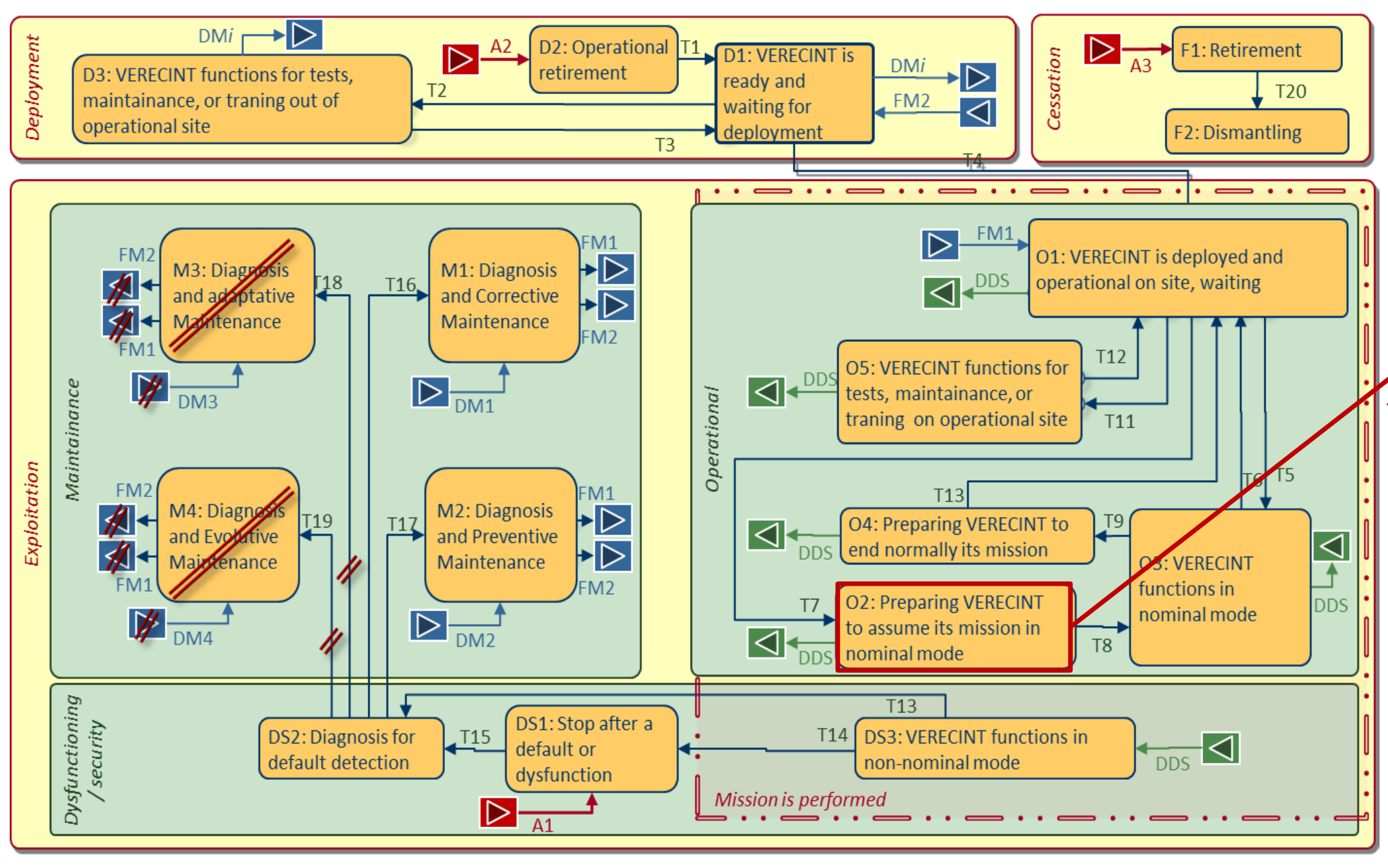
- **Principe** : traduire les **exigences de modélisation**, **métier** et **système** sous forme de **propriétés** et les prouver
- **Prouver les propriétés** : transformation vers le model checker UPPAAL (TCTL et Automates à états temporisés) et/ou vers COGITANT (Graphes Conceptuels)
- **Simuler le comportement** : sémantique opérationnelle du GEMOS

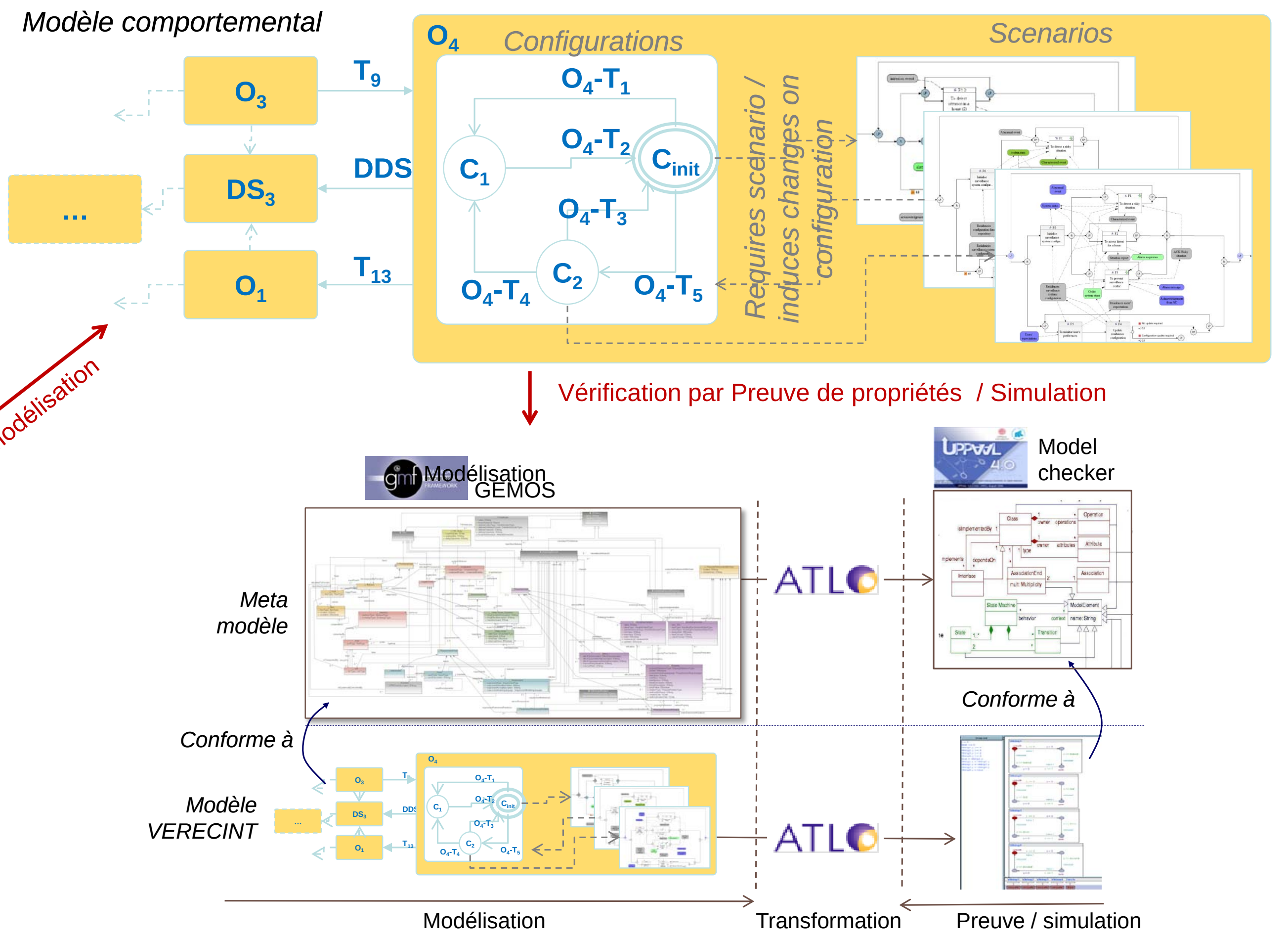


Processus opératoire actuel

http://pompiersdechateauneuf.com/lexique.php

VERECINT :
véhicule de recherche et d'intervention NRBC





Partenaires

Colloque Institut MInes-Télécom "Numérique : Grande échelle et complexité"

26/25 Mars 2014

Contact : Vincent.Chapurlat@mines-ales.fr

Site web : www.lgi2p.ema.fr

Contribution à l’évaluation d’architectures en Ingénierie des Systèmes Mécatroniques

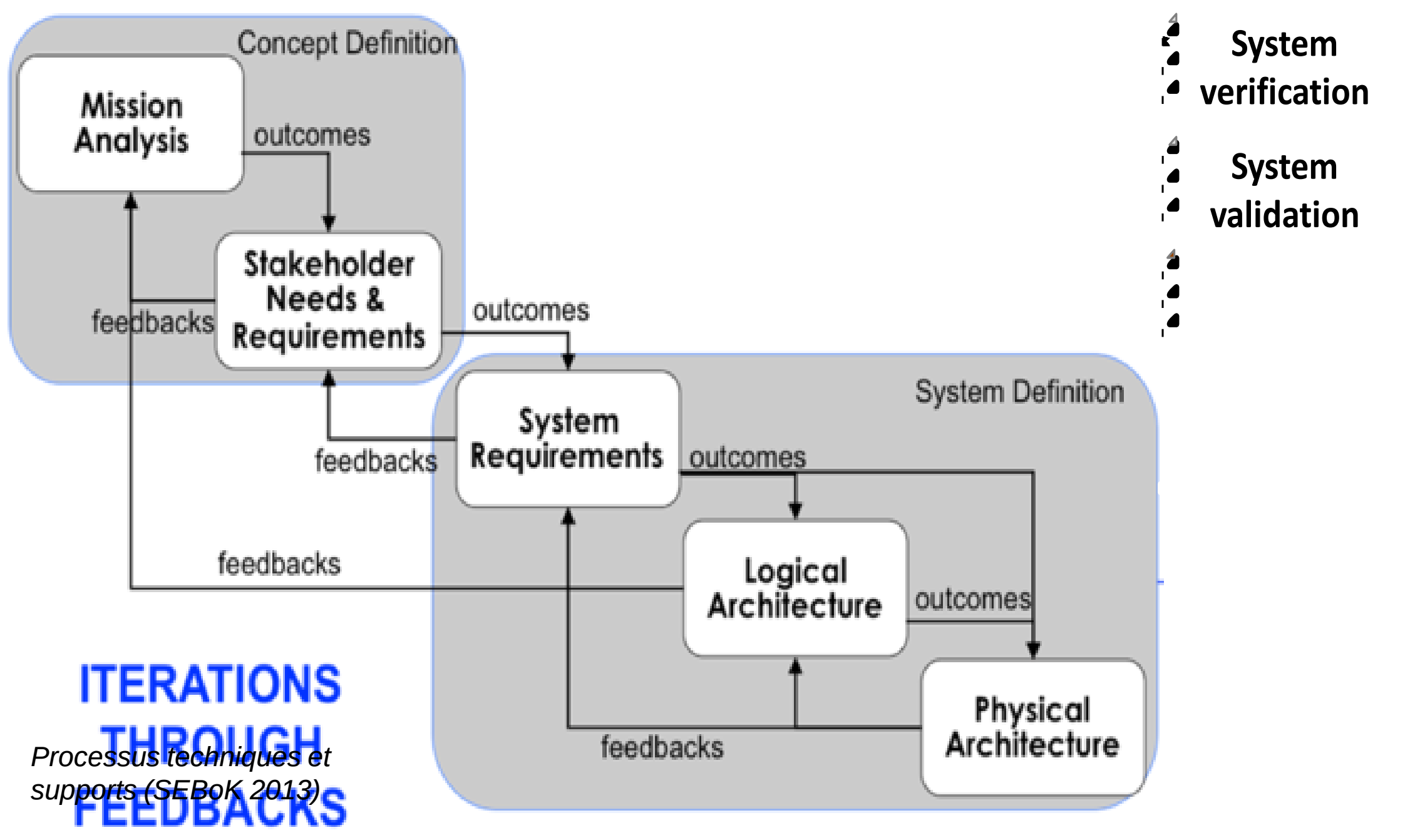
PROBLEMATIQUE ET VERROUS

Ingénierie Système (IS) et Mécatronique

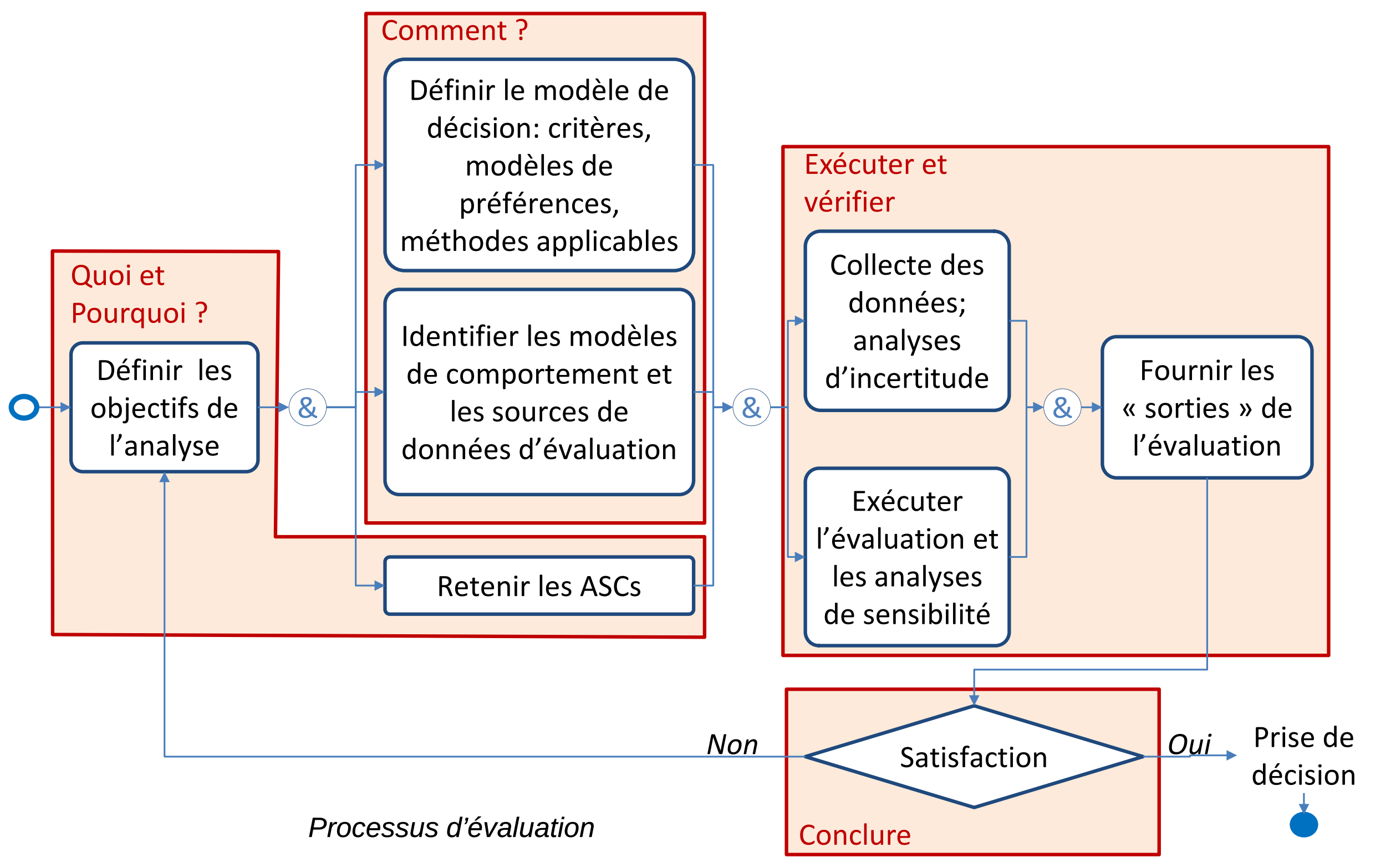
- Ingénierie interdisciplinaire / couplage fort entre disciplines
- Evaluation d’architectures organiques multi technologies : limites des modèles, des méthodes applicables et des outils.
- Nombreux paramètres / nombreux critères dont certains antagonistes
- Caractère itératif de la conception

Verrous

- Manque de vision partagée et unifiée de l’évaluation d’architectures candidates en IS : **System Analysis**
- Prise en compte et traçabilité des exigences et des connaissances métier de la Mécatronique au plus tôt
- Incertitude et imprécision des modèles d’analyse dans les premières phases de conception



Processus techniques et supports (SEBoK 2013)



Processus d’évaluation

Parties prenantes

Laboratoire de Génie Informatique et d’Ingénierie de Production

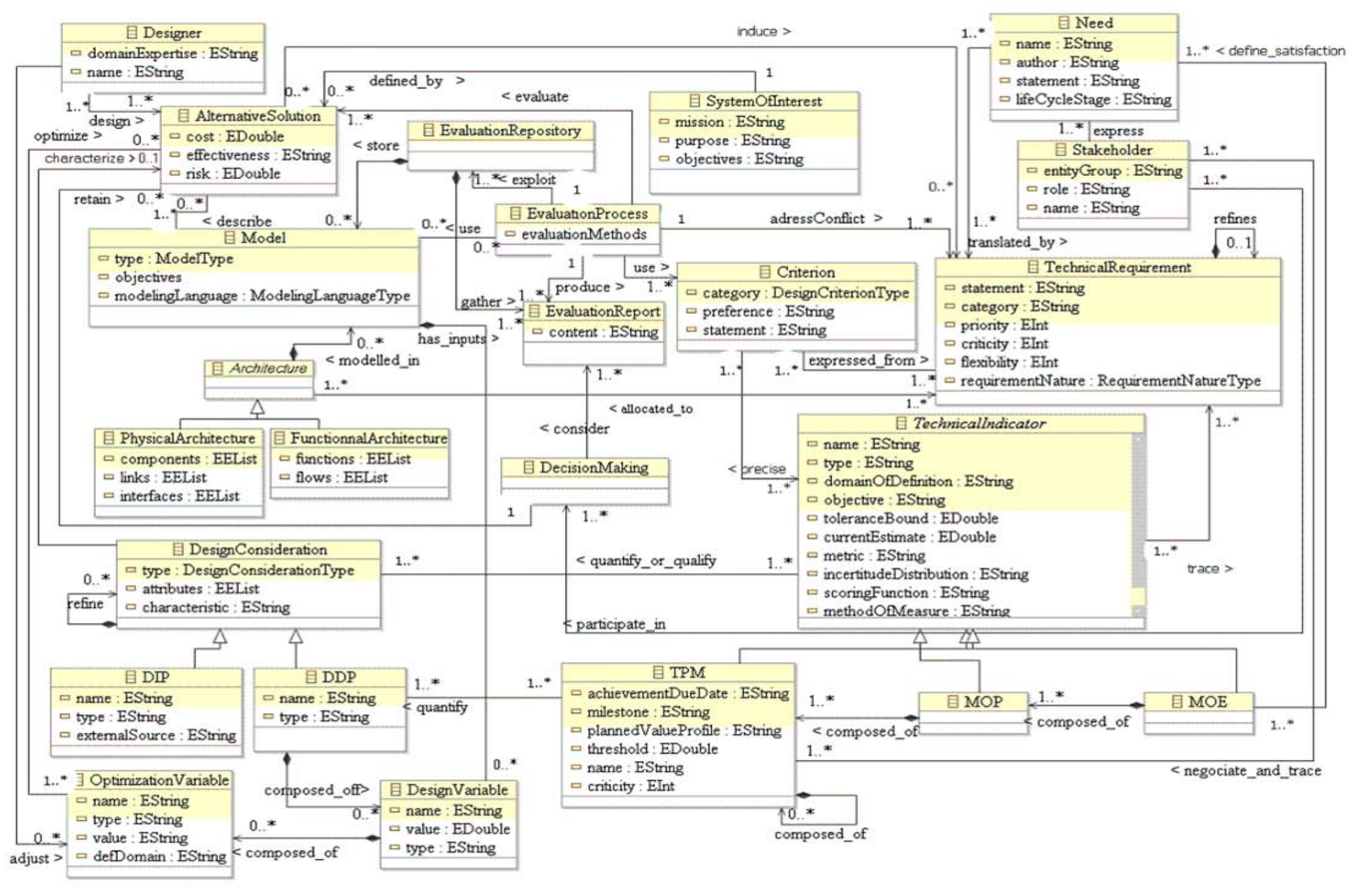
Auteurs

Couturier Pierre
Chapurlat Vincent
Lô Mambaye

CONTRIBUTIONS *(Lô et al. 2013)*

Conceptuelle

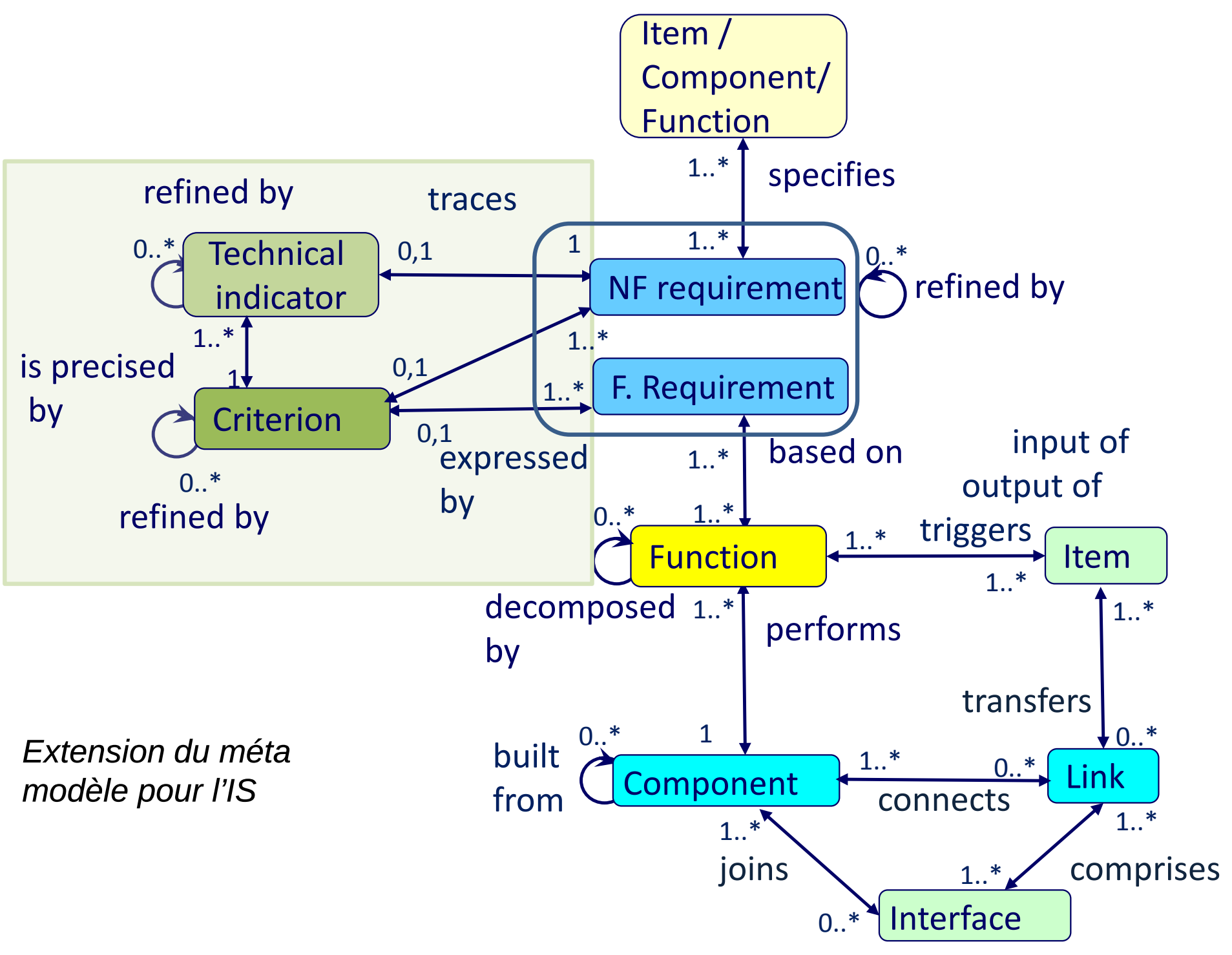
- Modèle conceptuel des données pour l’évaluation d’architectures en IS



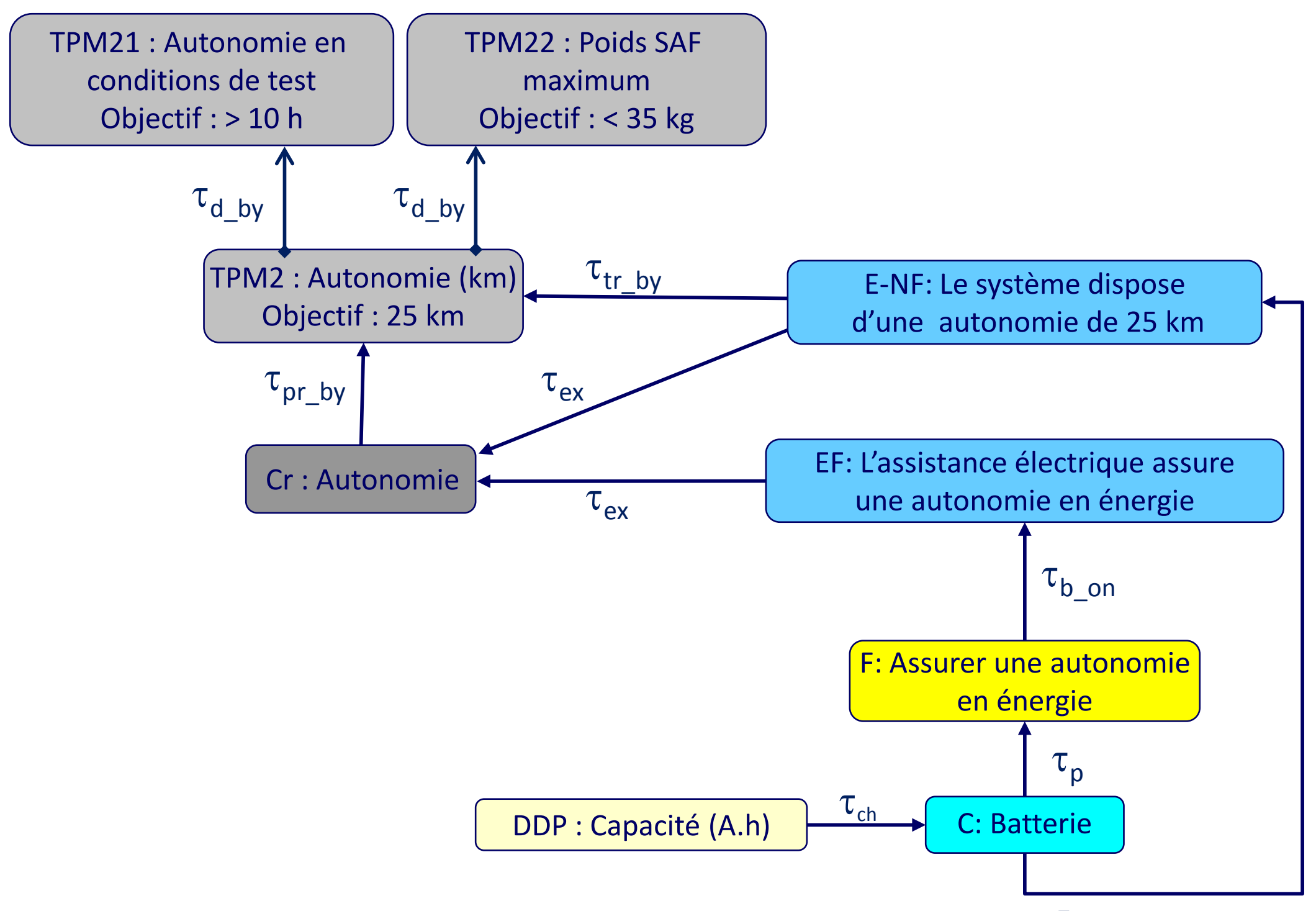
Modèle conceptuel des données pour l’évaluation

Méthodologique

- Traçabilité des choix de conception sur les modèles issus d’un projet de conception en IS
- Méthode qualitative d’analyse multicritère



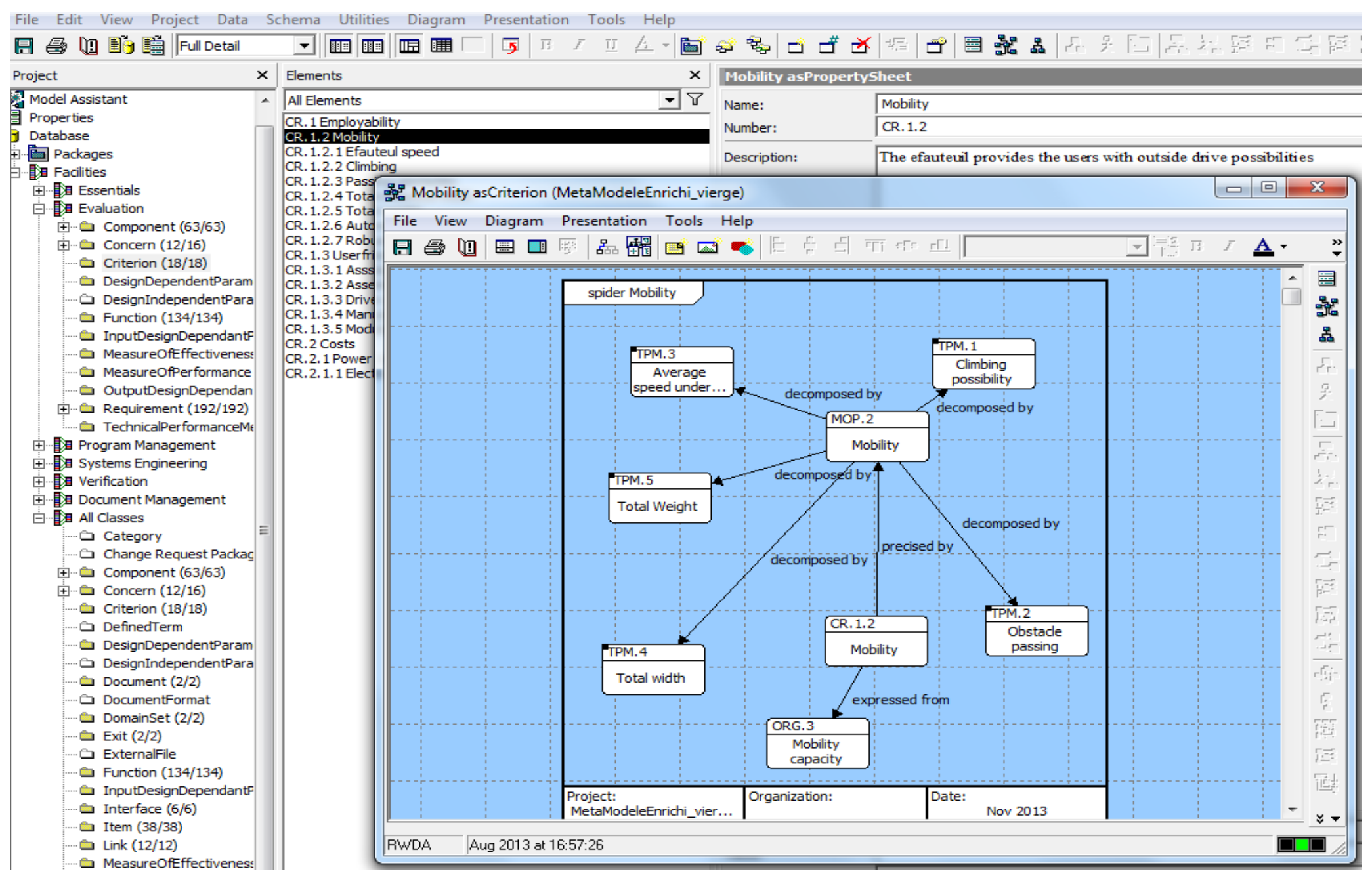
Extension du méta modèle pour l’IS

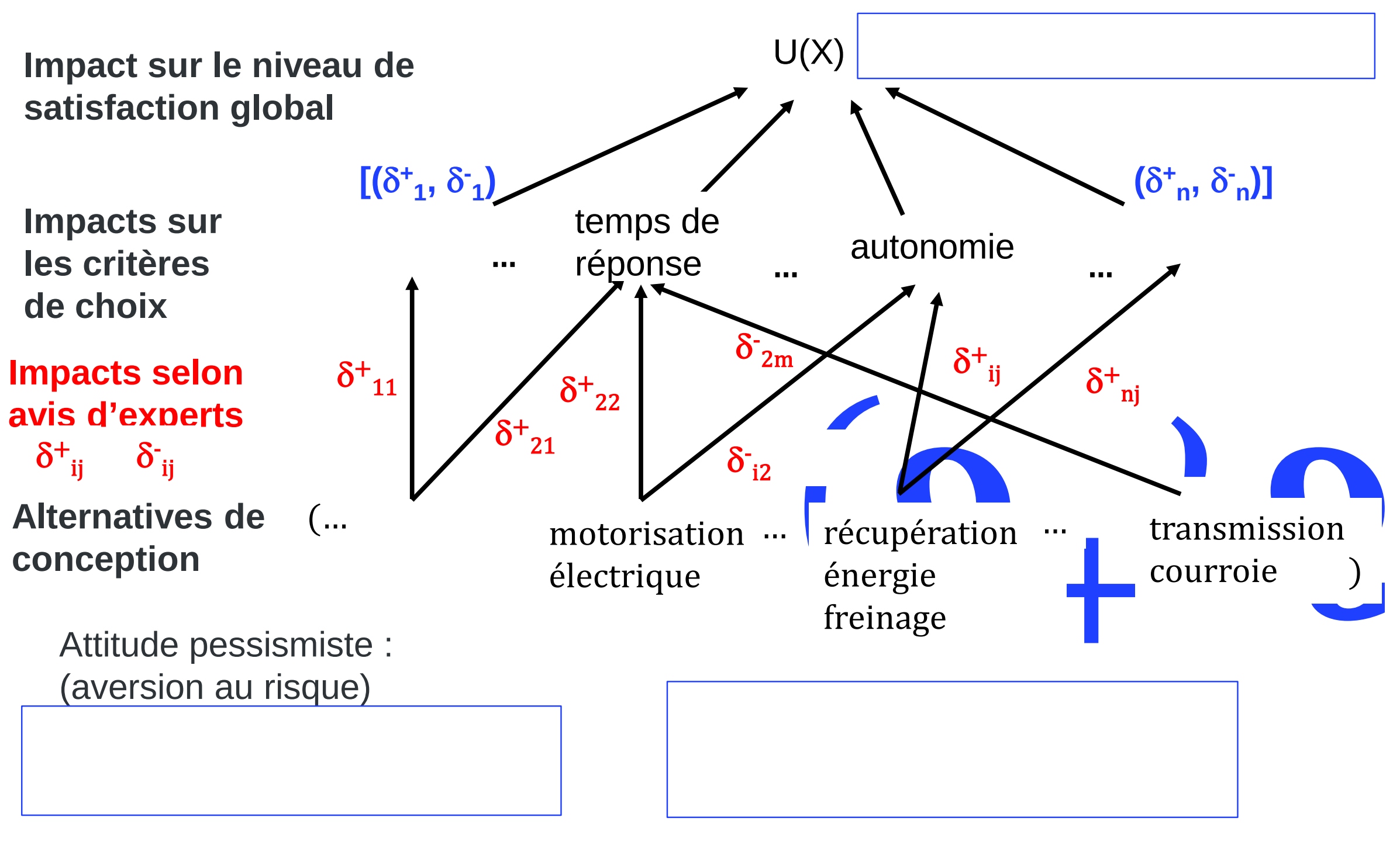


Liens de traçabilité

Technique

- Enrichissement de l’atelier d’IS CORE pour l’évaluation d’alternatives de solutions





Analyse qualitative multicritère

Partenaires

APPLICATION

Conception d’une assistance électrique pour fauteuil roulant

- Permettre l’accessibilité ou le maintien dans l’emploi de personnes à mobilité réduite
- Déclenchement de l’assistance électrique lors de la poussée sur la main courante
- Conduite habituelle d’un fauteuil roulant conservée
- La méthode d’évaluation permet d’identifier les solutions prometteuses.

Numérique : Grande échelle et complexité
26-27 mars 2014

Contact Pierre.Couturier@mines-ales.fr
Site web www.lgi2p.ema.fr

La robotique bio-inspirée : quand la robotique s'inspire du vivant.

Parties prenantes

Auteurs

Frédéric BOYER
Professeur
Ecole des Mines de Nantes

Vincent LEBASTARD
Maitre assistant
Ecole des Mines de Nantes

Mathieu POREZ
Maitre assistant
Ecole des Mines de Nantes

Partenaires

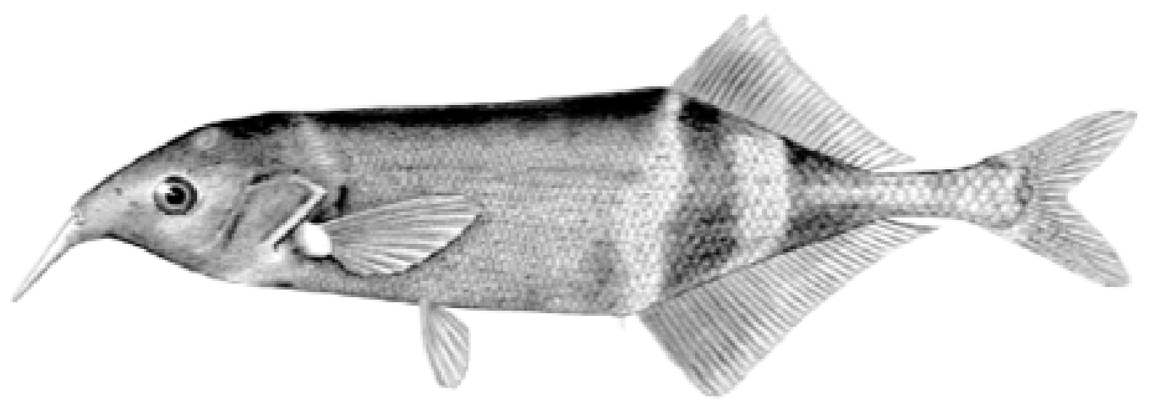
La bio-inspiration : un nouveau paradigme pour la robotique …

■ D'un point de vue conceptuel, la bio-inspiration est un modèle de pensée dans lequel la conception de nouvelles technologies est basée sur l'étude de la nature ou du vivant.

➞ En particulier, pour la robotique, ce paradigme consiste à s'inspirer des animaux pour lever les verrous de l'autonomie : c-à-d l'aptitude à percevoir, interpréter, décider et agir sur son environnement de manière adaptée sans interventions d'une volonté humaine extérieure. Dans ce contexte, l'autonomie est conçue comme le produit de :

- ■ l'intelligence incarnée dans la morphologie du corps ;
- ■ l'intelligence collective.

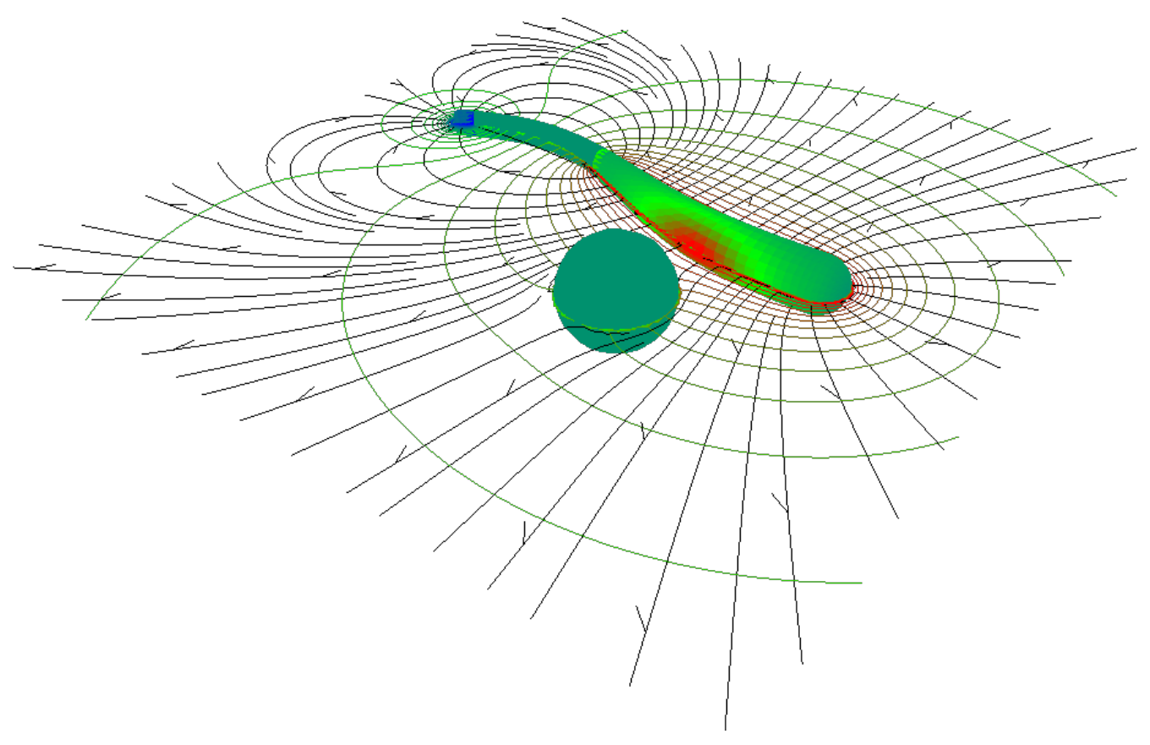
Observer



Gnathonemus petersii

↓

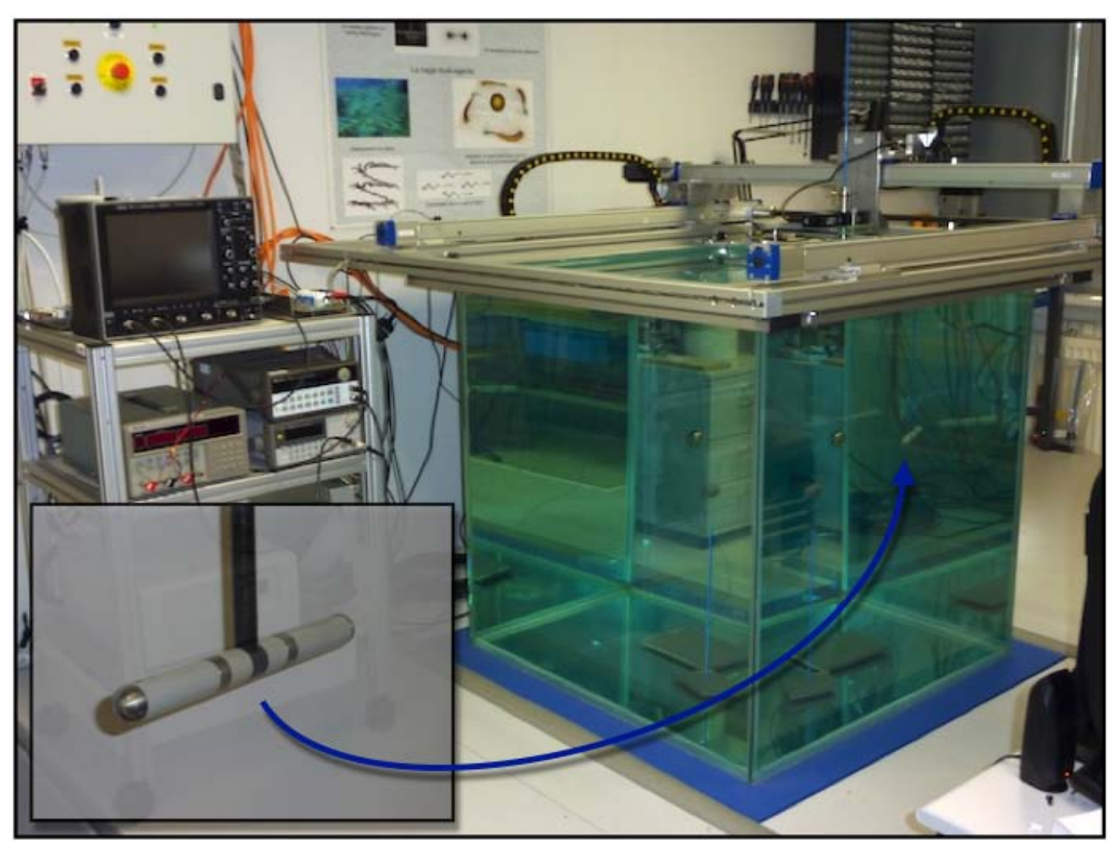
Comprendre



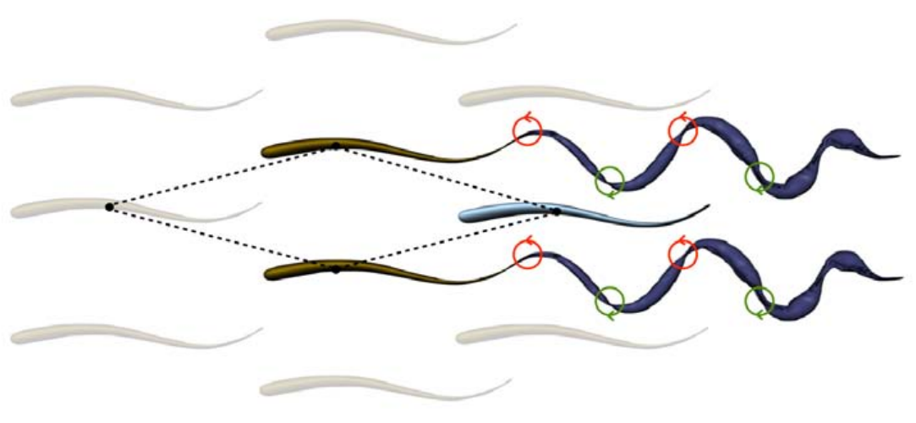
Modélisation FEM.

↓

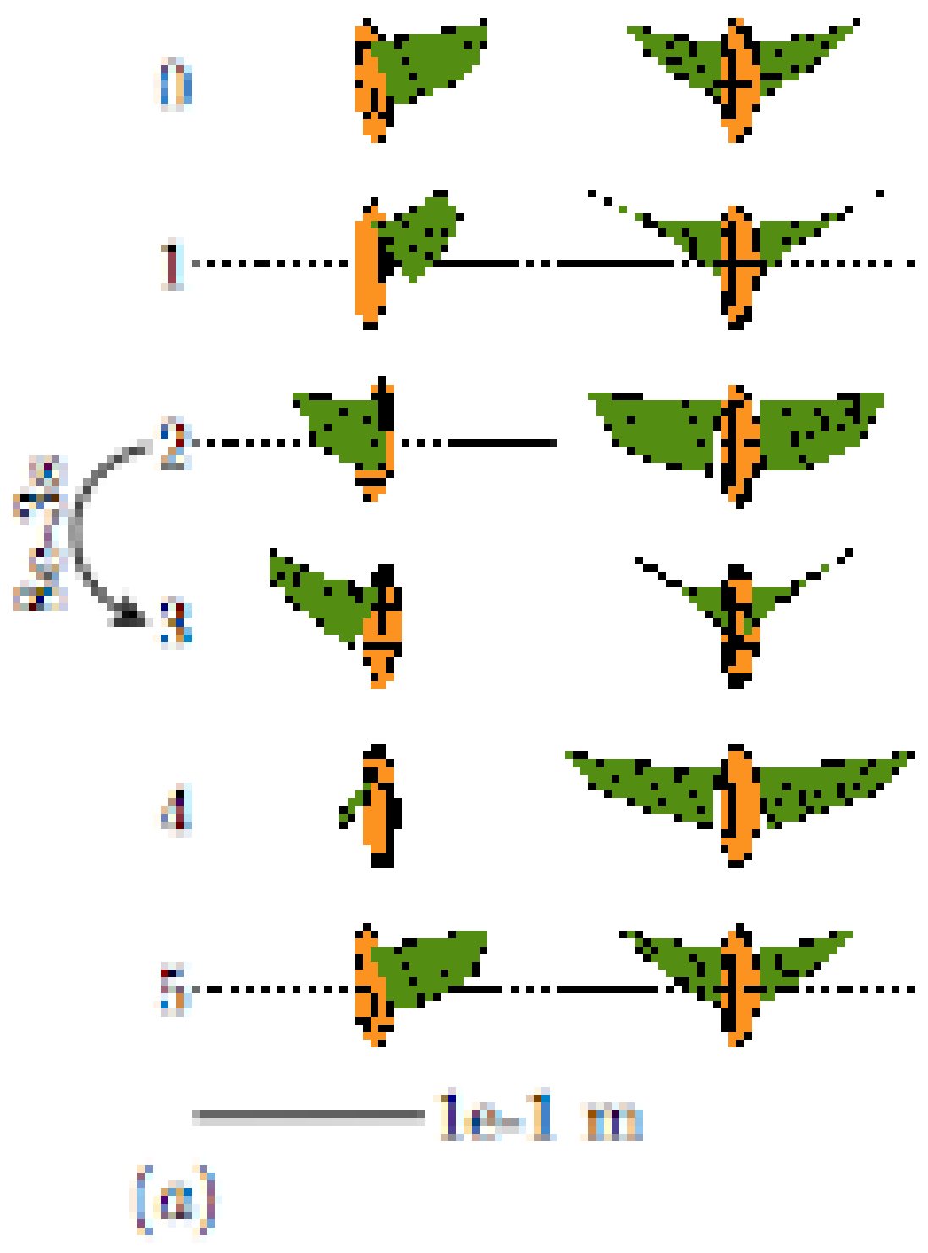
Reproduire



Capteur actif d'électrolocation
(WO Patent App. PCT/FR2012/051,764, 31 janvier, 2013.).



La nage en banc.



Le vol battant des insectes.

Nos thèmes de recherche :

■ La locomotion (depuis 2003) :

- ■ élaboration d'une théorie générale de la locomotion bio-inspirée en robotique ;
- ■ conception d'outils de modélisation et de simulation dédiés à la commande ;
- ➞ applications à la nage des poissons, la reptation des serpents, le vol battant des insectes, etc …

■ La perception (depuis 2007) :

- ■ la perception inspirée des poissons électriques ;
- ■ modélisation du sens électrique ;
- ■ conception de capteurs innovants pour la robotique ;
- ■ commande pour la navigation de robots sous-marins ;
- ➞ brevet WO Patent App. PCT/FR2012/051,764, 31 janvier, 2013.

Les projets (passés et présents) :

■ CNRS ROBEA Robot Anguille (2003-2006) - Etude et réalisation d'un robot anguille.

■ ANR PSIRob RAAMO (2007-2011) - Etude et réalisation d'un robot anguille autonome doté du sens électrique.

■ FP7 FET ANGELS (2009-2012) - Etude des interactions entre morphologie, perception et locomotion : application à la robotique sous-marine.

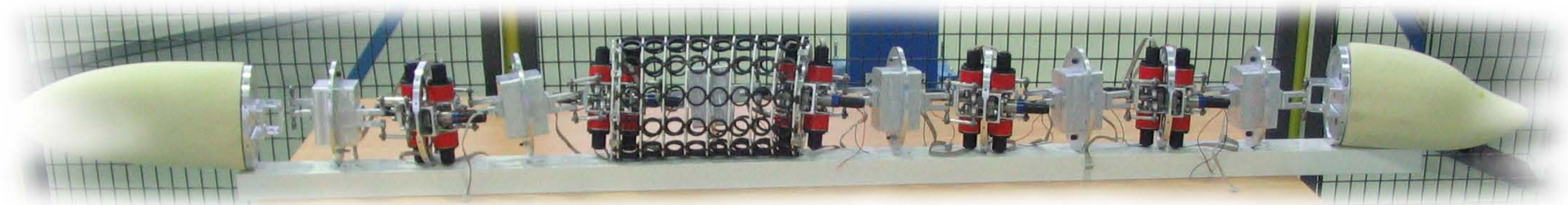
■ ANR Blanc EVA (2008-2013) - Etude et réalisation d'un robot volant autonome inspiré de l'insecte.

■ Projet Région et Carnot (2012-2013) - Equipements de laboratoire.

■ Projet Région CEA-Tech (2014-2017) - Télé-manipulation par retour électro-haptique dans l'eau et l'air.

Un module du robot ANGELS.

Le laboratoire de robotique bio-inspirée de l'Ecole des Mines de Nantes.



Le robot Anguille.

26 & 27 Mars 2014 Numérique : Grande échelle et complexité.

Contact frédéric.boyer@mines-nantes.fr Site web www.irccyn.ec-nantes.fr

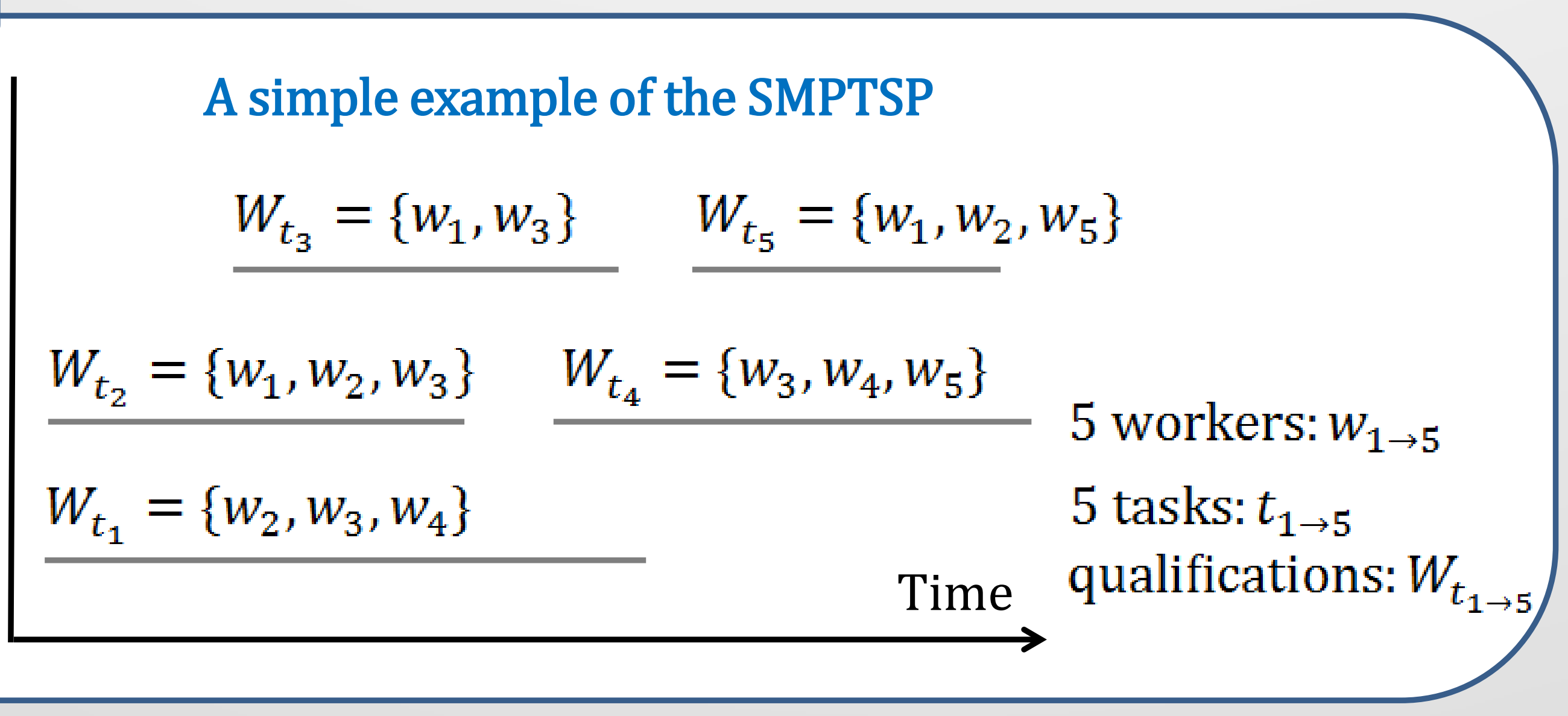
Filtering atMostNValue with difference constraints: application to the Shift Minimisation Personnel Task Scheduling Problem

Jean-Guillaume FAGES jean-guillaume.fages@mines-nantes.fr Tanguy LAPEGUE tanguy.lapegue@mines-nantes.fr

1) Problem & Applications

Objective: minimise resource consumption
C1: Overlapping tasks need different resources
C2: Tasks require qualified resources

Applications:
- Assignment of classes to rooms
- Assignment of fixed jobs to machines
- Assignment of fixed tasks to workers



A simple example of the SMPTSP

$W_{t_3} = \{w_1, w_3\}$ $W_{t_5} = \{w_1, w_2, w_5\}$
$W_{t_2} = \{w_1, w_2, w_3\}$ $W_{t_4} = \{w_3, w_4, w_5\}$
$W_{t_1} = \{w_2, w_3, w_4\}$

5 workers: $w_{1\to5}$
5 tasks: $t_{1\to5}$
qualifications: $W_{t_{1\to5}}$

2) Straightforward CP Model

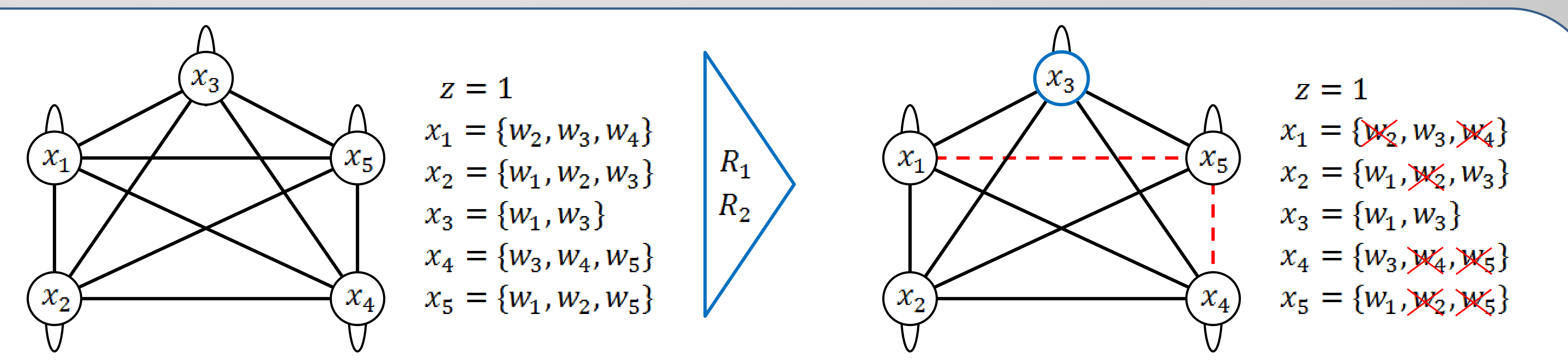
minimise(z)
s.t. : $AllDifferent(x_i | i \in K) \quad \forall K$ — No Communication
$AtMostNValue(X, z)$
$Dom(z) = [LB_{\neq}; |W|]$
$Dom(x_i) = W_{t_i} \quad \forall t_i \in T$

K: a maximal set of overlapping tasks
T: set of tasks W: set of workers
$LB_{\neq}$: size of the largest set K

3) AtMostNValue filtering

Given the intersection graph $G_I = (V, E_I)$ of the set of variables X, along with an independent set A in G_I

$R_1: \underline{z} \leftarrow \max(\underline{z}, |A|)$
$R_2: |A| = \overline{z} \Rightarrow \forall i \in V,\ Dom(x_i) \leftarrow Dom(x_i) \cap \bigcup_{a \in A} Dom(x_a)$

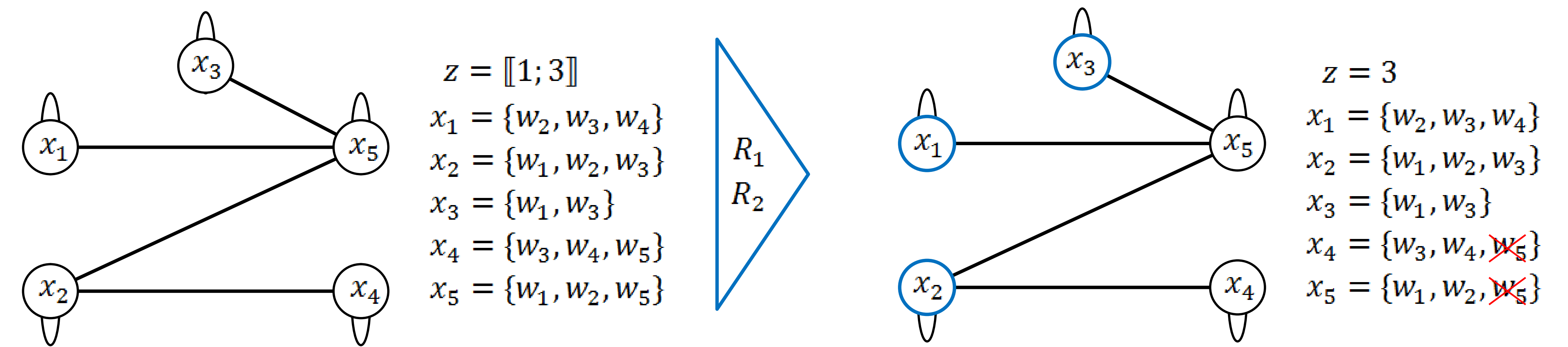


$z = 1$
$x_1 = \{w_2, w_3, w_4\}$
$x_2 = \{w_1, w_2, w_3\}$
$x_3 = \{w_1, w_3\}$
$x_4 = \{w_3, w_4, w_5\}$
$x_5 = \{w_1, w_2, w_5\}$

$z = 1$
$x_1 = \{\cancel{w_2}, w_3, \cancel{w_4}\}$
$x_2 = \{w_1, \cancel{w_2}, w_3\}$
$x_3 = \{w_1, w_3\}$
$x_4 = \{w_3, \cancel{w_4}, \cancel{w_5}\}$
$x_5 = \{w_1, \cancel{w_2}, \cancel{w_5}\}$

4) Improving filtering (Graph & Rules)

The constrained intersection graph G_{CI} of the set of variables X and the set of differences constraints C is deduced from G_I by removing edges (i, j) whenever $neq(i, j) \in C$

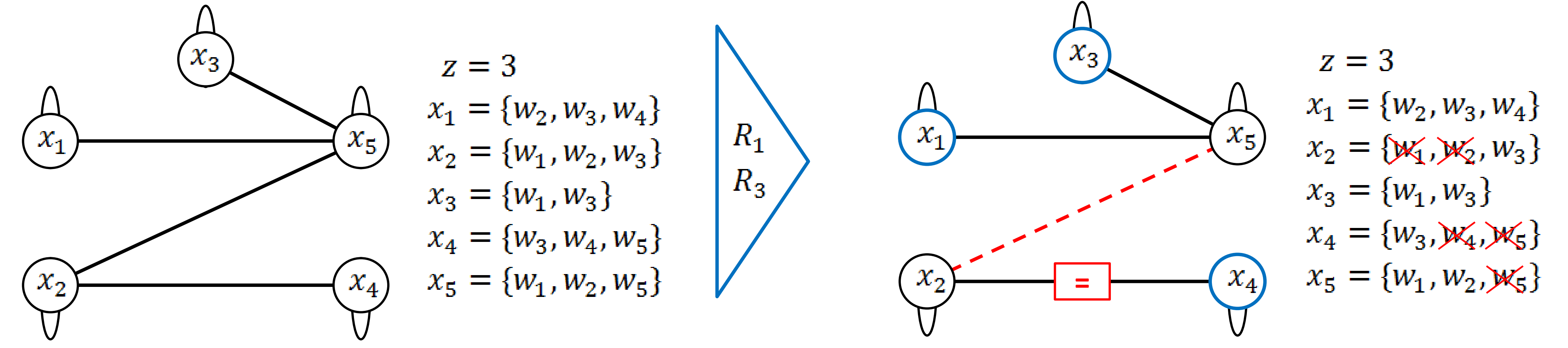


$z = [\![1; 3]\!]$
$x_1 = \{w_2, w_3, w_4\}$
$x_2 = \{w_1, w_2, w_3\}$
$x_3 = \{w_1, w_3\}$
$x_4 = \{w_3, w_4, w_5\}$
$x_5 = \{w_1, w_2, w_5\}$

$z = 3$
$x_1 = \{w_2, w_3, w_4\}$
$x_2 = \{w_1, w_2, w_3\}$
$x_3 = \{w_1, w_3\}$
$x_4 = \{w_3, w_4, \cancel{w_5}\}$
$x_5 = \{w_1, w_2, \cancel{w_5}\}$

Given an independent set A of $G_{CI} = (V, E_{CI})$

$R_3: |A| = \overline{z} \Rightarrow \forall i \in V \backslash A,$
$$\begin{cases} A_i = \{a\} \Rightarrow Dom(x_a) \leftarrow Dom(x_a) \cap Dom(x_i) \\ Dom(x_i) \leftarrow Dom(x_i) \cap \bigcup_{a \in A} Dom(x_a) \end{cases}$$



$z = 3$
$x_1 = \{w_2, w_3, w_4\}$
$x_2 = \{w_1, w_2, w_3\}$
$x_3 = \{w_1, w_3\}$
$x_4 = \{w_3, w_4, w_5\}$
$x_5 = \{w_1, w_2, w_5\}$

$z = 3$
$x_1 = \{w_2, w_3, w_4\}$
$x_2 = \{\cancel{w_1}, \cancel{w_2}, w_3\}$
$x_3 = \{w_1, w_3\}$
$x_4 = \{w_3, \cancel{w_4}, \cancel{w_5}\}$
$x_5 = \{w_1, w_2, \cancel{w_5}\}$

5) Diversifying filtering

Rules rely on independent sets:
- Finding **large** independent sets is important
- Finding **different** independent sets is important

What is done in the literature?
→ minDegree algorithm (MD)
  1) Fast
  2) Effective
  3) Deterministic → No diversification

How to get diversification?

1) Breaking ties randomly in MD? (no impact)
2) Computes k pseudo-random independent sets? (not effective)
3) Computes k random independent sets: R^k (improves filtering)
- Complements MD
- Provides control over the tradeoff time/filtering

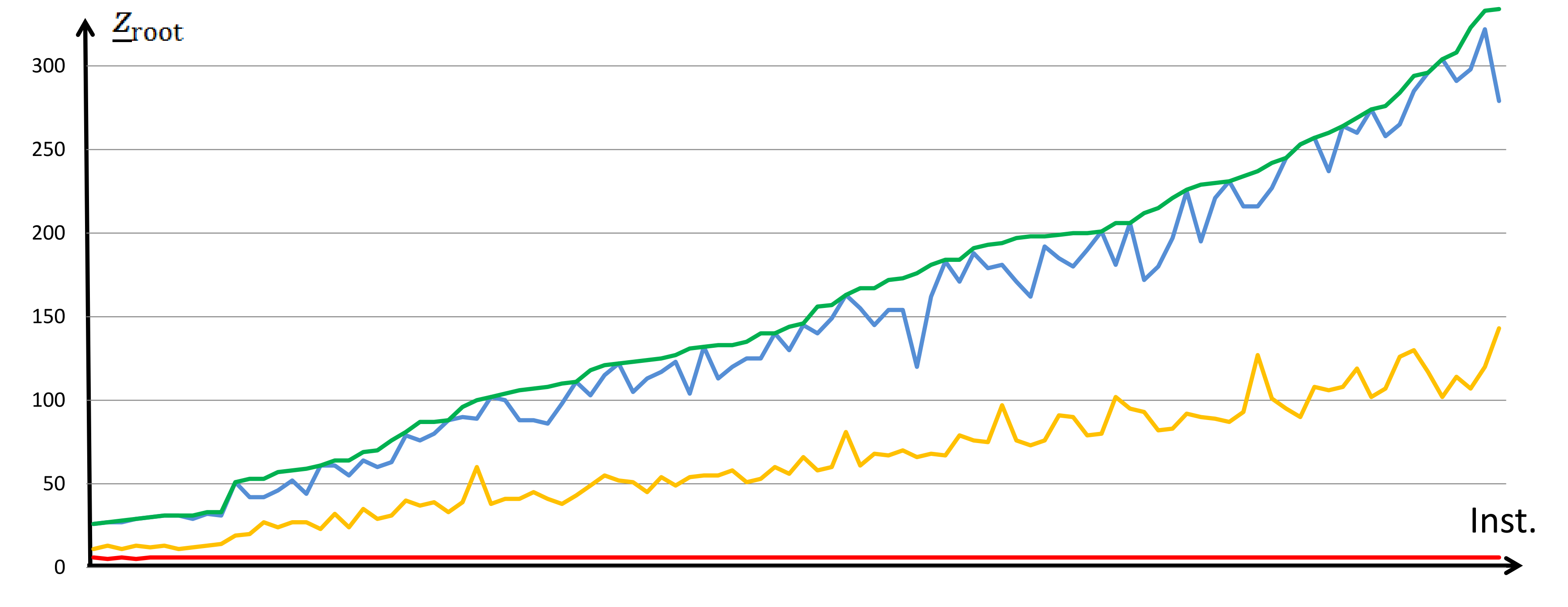
↳ Improvements opportunity

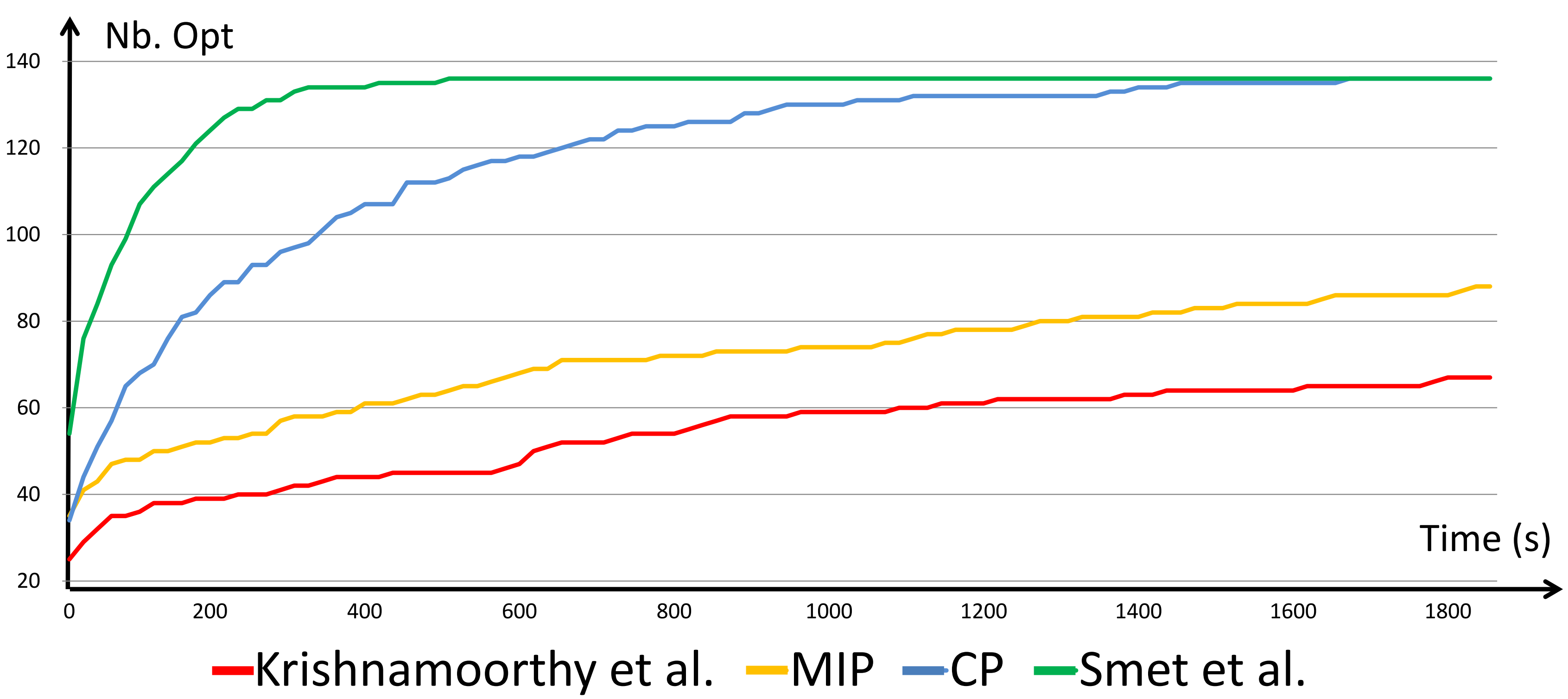
	Data 137				
k	10	30	50	70	90
Nb. Opt	101	**109**	106	99	95

	Data 100				
k	100	200	400	800	1600
Nb. Opt	19	22	28	**35**	31

Results after 5 min with $\mathrm{AMNV} < G_{CI} | \mathrm{MD}, R^k | R_{1,3} >$

6) Results & Literature





INSTITUT Mines-Télécom
MINES Nantes
LABORATOIRE D'INFORMATIQUE DE NANTES ATLANTIQUE

Programme Gaspard Monge pour l'Optimisation et la Recherche Opérationnelle

Learning Constraints for Reducing Combinatorics

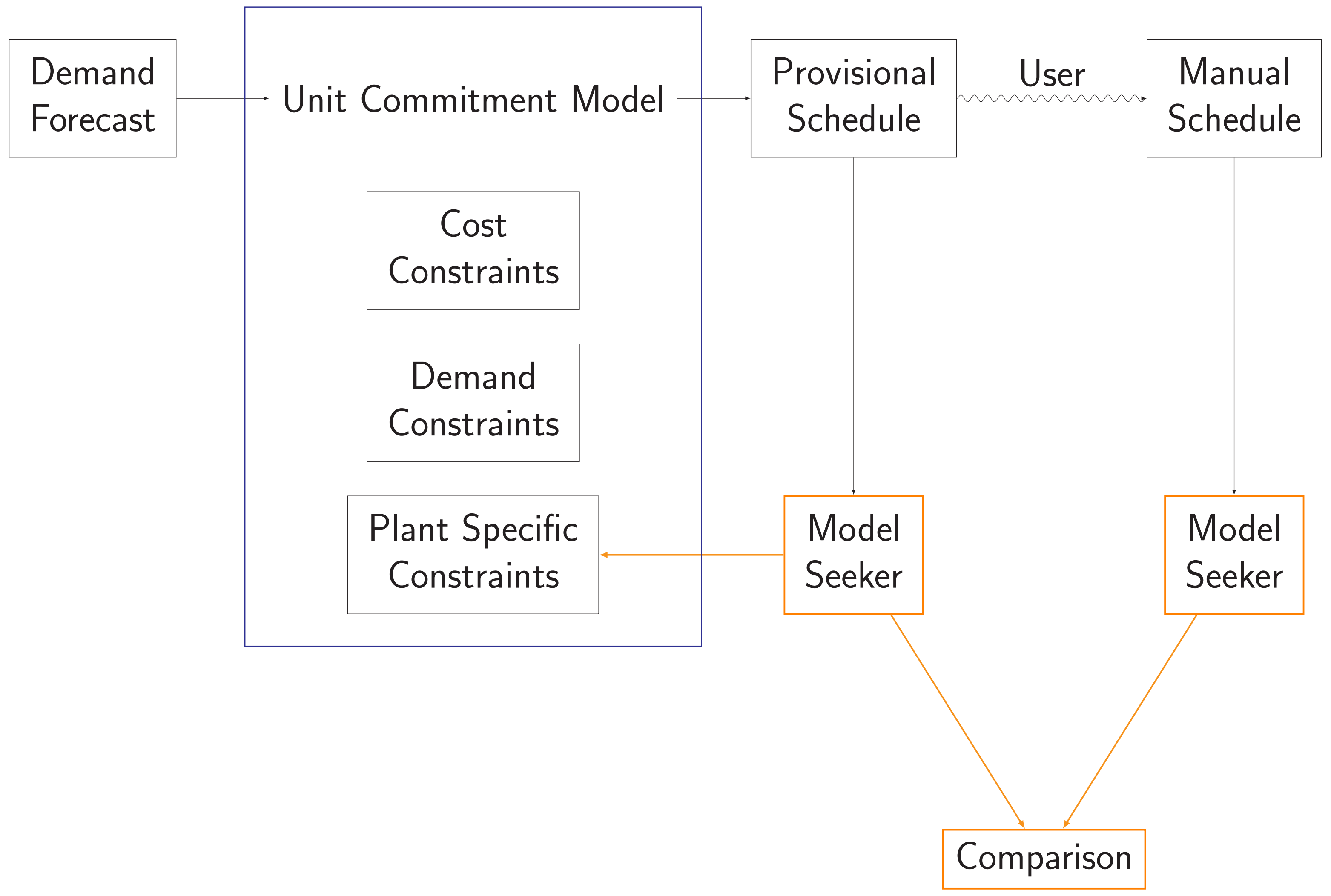
Nicolas Beldiceanu, Helmut Simonis, Georgiana Ifrim, Arnaud Lenoir, Jean-Yves Lucas

TASC team (CNRS/INRIA), Mines de Nantes, France
Cork Constraint Computation Centre, University College Cork, Ireland
EDF Research, France

Why, What, How?

- Learn plant specific constraints from production planning data
- Discover known or perhaps new, hidden constraints
- Use output of Unit Commitment Problem (UCP)
- Consider different plant types (nuclear, thermal, hydro)
- Learn from both the provisional schedule and manually modified solutions; compare
- Using, adapting and extending existing ConstraintSeeker and ModelSeeker tools
- New, specialized UCP-ModelSeeker tool combining Constraint Programming and Machine Learning
- Adding new global constraints to Global Constraint Catalogue
- Run on large datasets (1.5 million samples)

Learning in the EDF Unit Commitment Problem


Demand Forecast

Unit Commitment Model

Cost Constraints

Demand Constraints

Plant Specific Constraints

Provisional Schedule

User

Manual Schedule

Model Seeker

Model Seeker

Comparison

Main Components of UCP-ModelSeeker

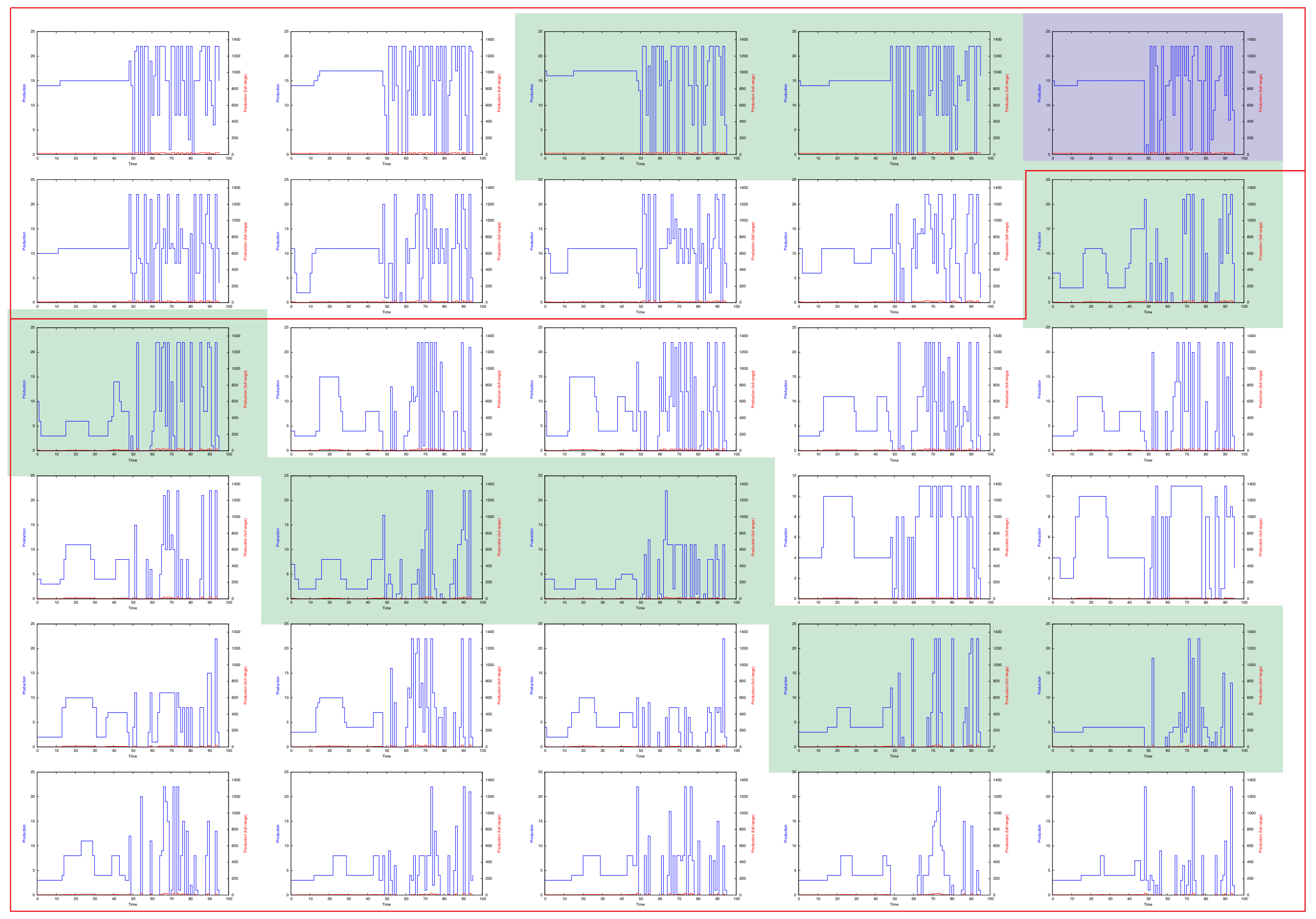
CLUSTERING TIME SERIES
(to learn stronger constraints)

LEARNING DOMAINS
- For each period
- For each functional dependency parameter

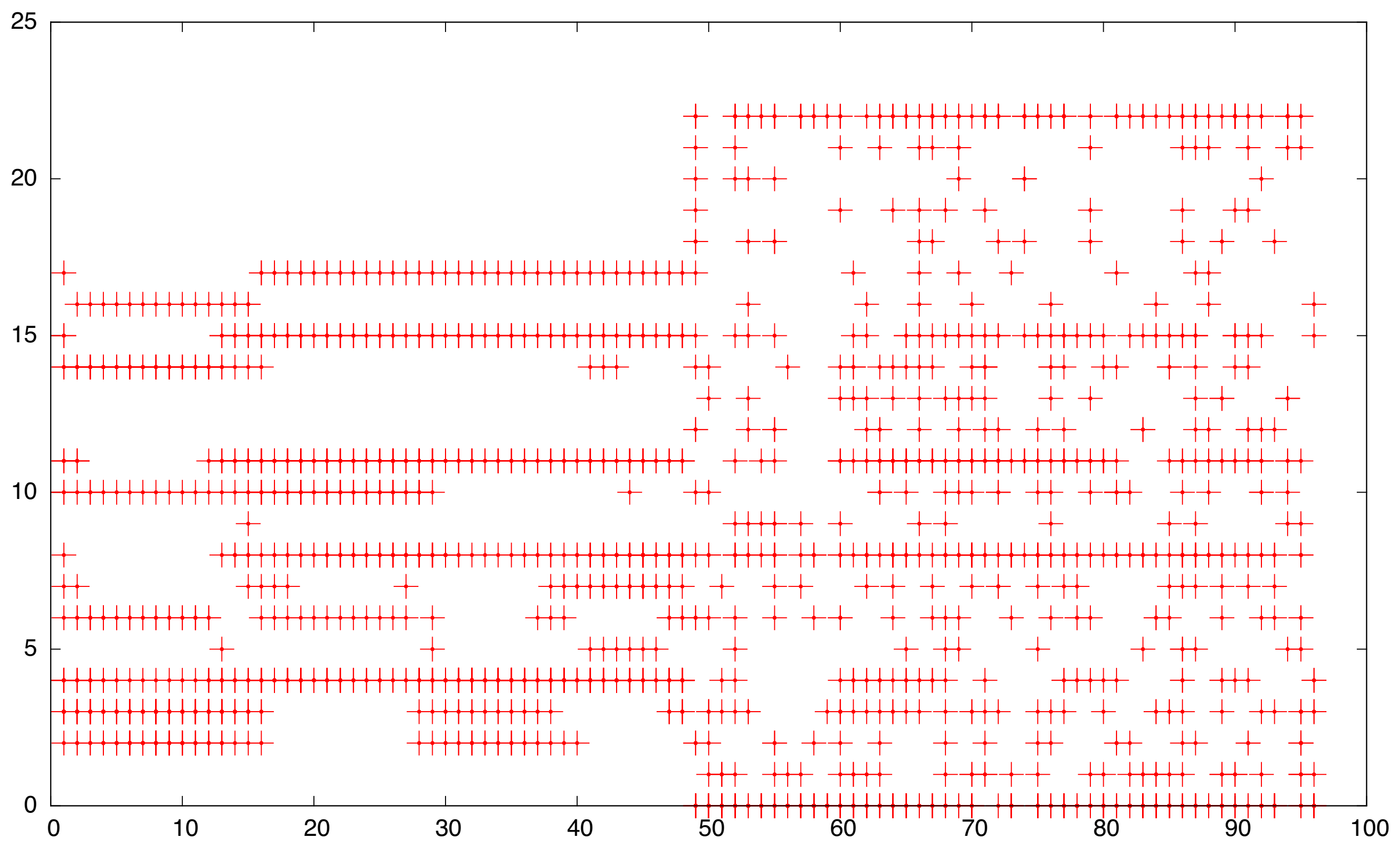
LEARNING CONSTRAINTS
- Functional dependency constraints
- Constraints without functional dependency
- Binary constraints (on consecutive periods)
- Variable partitions generator (find breakpoints)

Execute Model
(generate similar time series)

Clustering of Power Output of Example Plant, April 2010. (Clusters in red, Weekends/Holidays in green/violet)



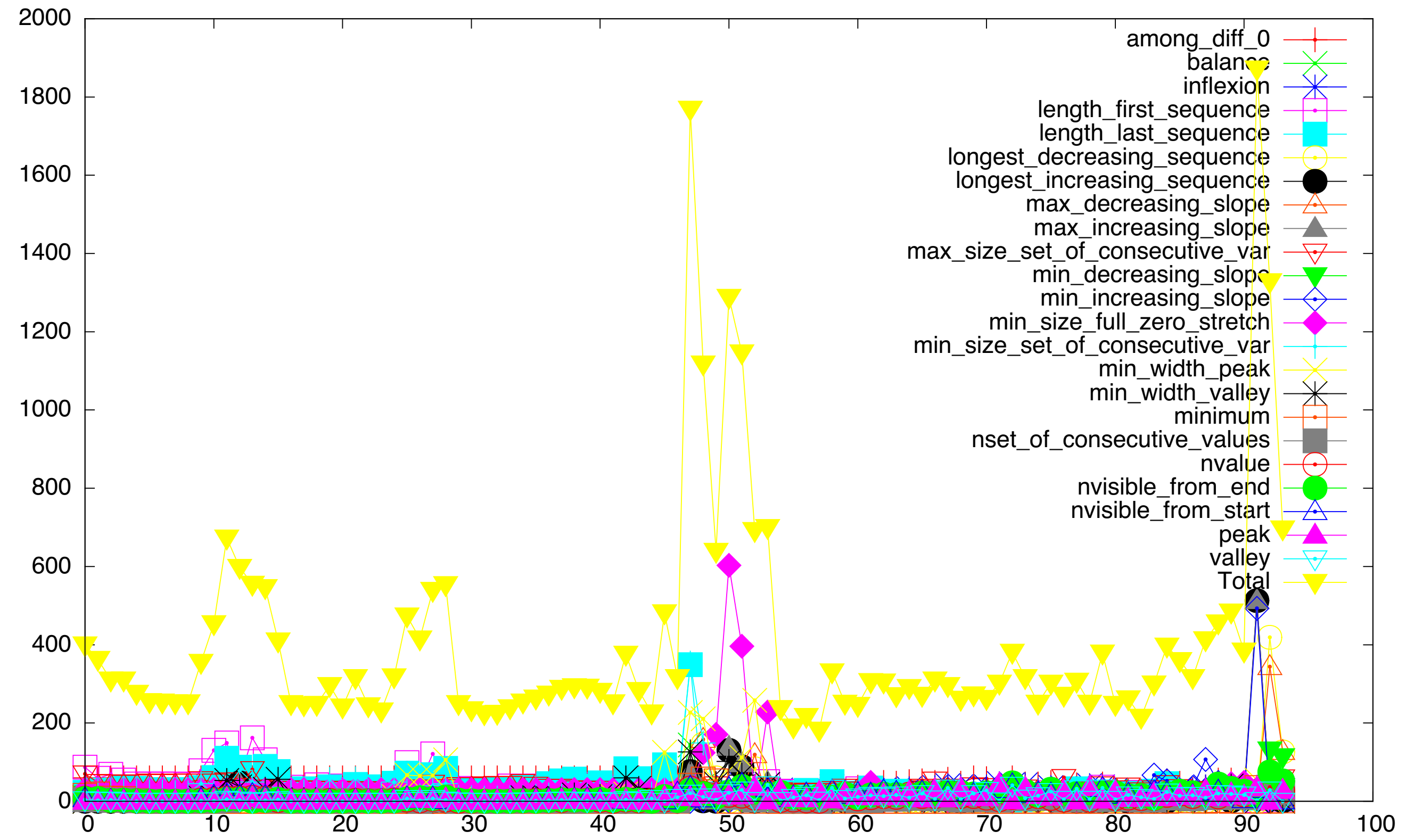
Variable Domains for Example Plant



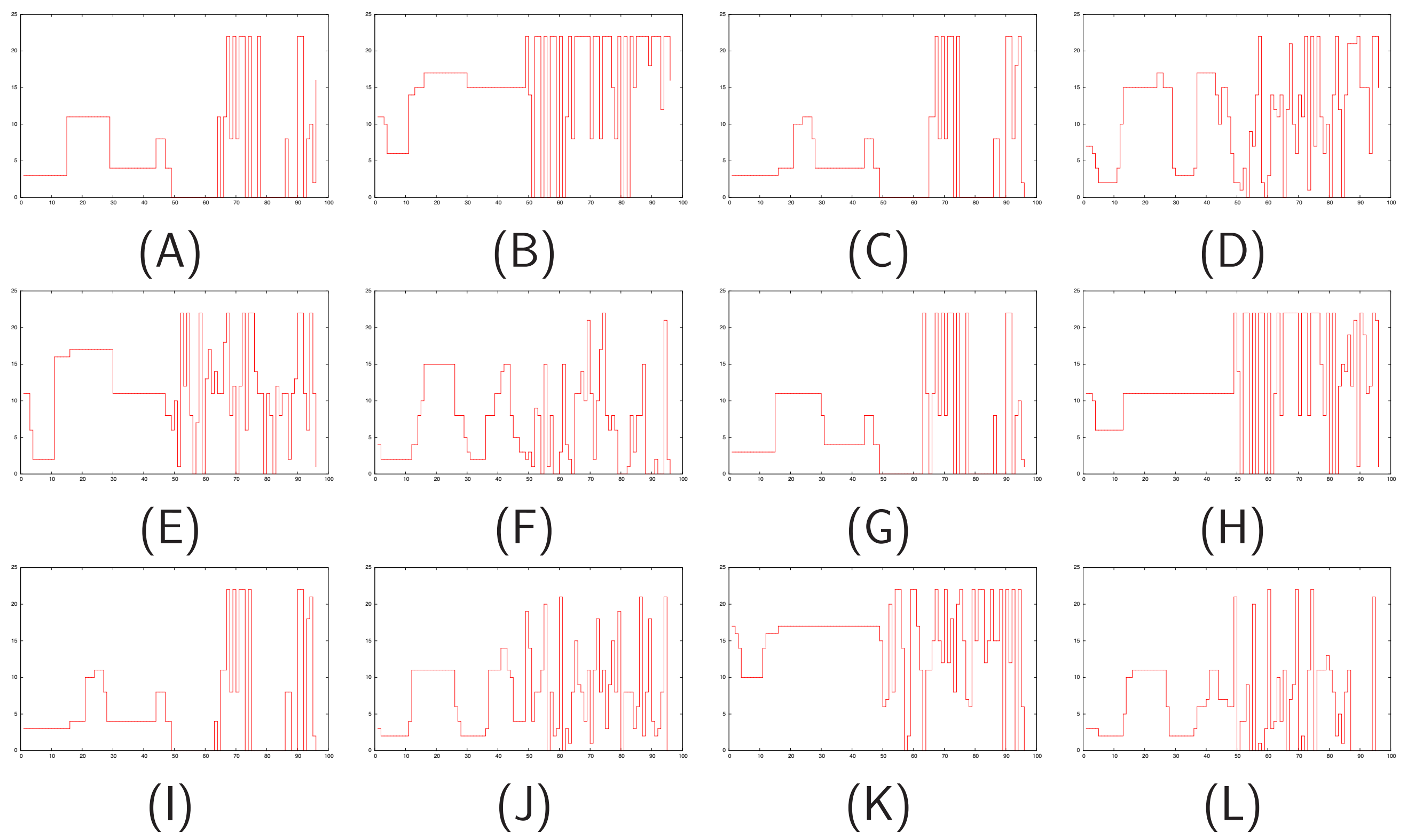
New Constraints to Characterize Structured TIme Series

among_diff_0	: *number of values different from* **0**,
max_nvalue	: *number of occurrences of the most used value,*
min_nvalue	: *number of occurrences of the least used value,*
balance	: *difference in count of the most and least used values,*
change	: *number of consecutive values that are different.*
peak	: *number of peaks,*
highest_peak	: *altitude of the highest peak,*
min_width_peak	: *smallest width of any peak,*
nvisible_from_start	: *number of peaks visible from the start,*
nvisible_from_end	: *number of peaks visible from the end,*
inflexion	: *number of peaks and valleys,*
min_dist_between_inflexion	: *minimum distance between consecutive inflexions,*
longest_increasing_sequence	: *range of the longest increasing subsequence,*
max_increasing_slope	: *maximum slope on the strictly increasing subsequences,*
min_increasing_slope	: *minimum slope on the strictly increasing subsequences,*
big_peak	: *number of big peaks of .*

Identify Timepoint(s) When Profile Was Manually Updated: Find Sub-Sequences with Different Behaviour



UCP-ModelSeeker Generated Profiles for Example Plant


(A) (B) (C) (D)

(E) (F) (G) (H)

(I) (J) (K) (L)

Comparing Constraints, Search Strategies and Solution Quality Relative to Input Data

Variant	**Search**	**Cluster**	**Split**	**MAE**	**MSE**	**Time (sec)**
A	Frequent	All	no	452.30	**87.53**	1.10
B	Frequent	1	no	449.67	104.62	0.53
C	Frequent	2	no	298.43	70.90	0.87
D	Random	All	no	649.97	114.20	1.43
E	Random	1	no	492.90	106.68	0.54
F	Random	2	no	422.33	82.48	0.89
G	Frequent	All	yes	**445.10**	87.82	2.30
H	Frequent	1	yes	**431.33**	**101.70**	1.03
I	Frequent	2	yes	**294.00**	**70.70**	1.74
J	Random	All	yes	547.37	97.23	2.32
K	Random	1	yes	510.22	111.71	1.03
L	Random	2	yes	397.86	78.38	1.73

For More Information

`http://4c.ucc.ie/~hsimonis/edfcp2013.pdf`
`http://4c.ucc.ie/~hsimonis/modelseeker.pdf`

Cork Constraint Computation Centre, University College Cork, Ireland   Mail: `h.simonis@4c.ucc.ie`   WWW: `http://4c.ucc.ie/~hsimonis`

INSTITUT
Mines-Télécom

ADVANCED IMAGE PROCESSING IN VISUAL INSPECTION

MINES ParisTech
ARMINES

Auteurs

Petr Dokladal

Center for Mathematical Morphology,
35, rue Saint-Honoré, 77 300
Fontainebleau,
http://cmm.mines-paristech.fr

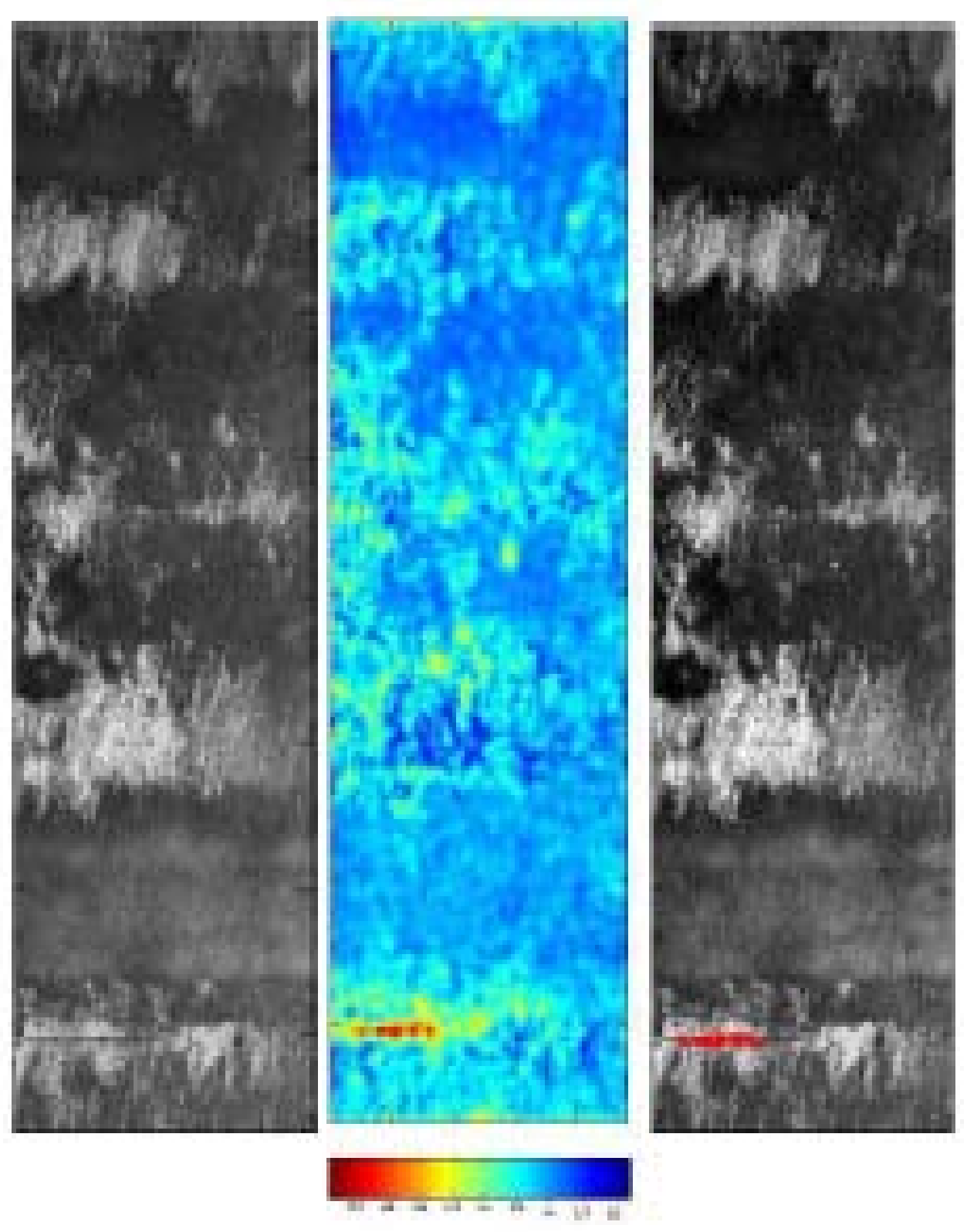
Real-Time Control of Metal Sheet Lamination Process

Defect Detection in Heavily Textured Surfaces

- Every pixel is described using a feature vector (30 linear and morphological filters and 26 curvelets).
- A statistical learning is used to discriminate the defects.
- The processing is optimized (cascaded) to verify the needs of real-time processing.

Cord A., Bach F., Jeulin D. Texture classification by statistical learning from morphological image processing. Application to metallic surfaces, Journal of Microscopy, 239, pp. 159-166, 2010

contact: Dominique.Jeulin@mines-paristech.fr


sheet image probability image defect

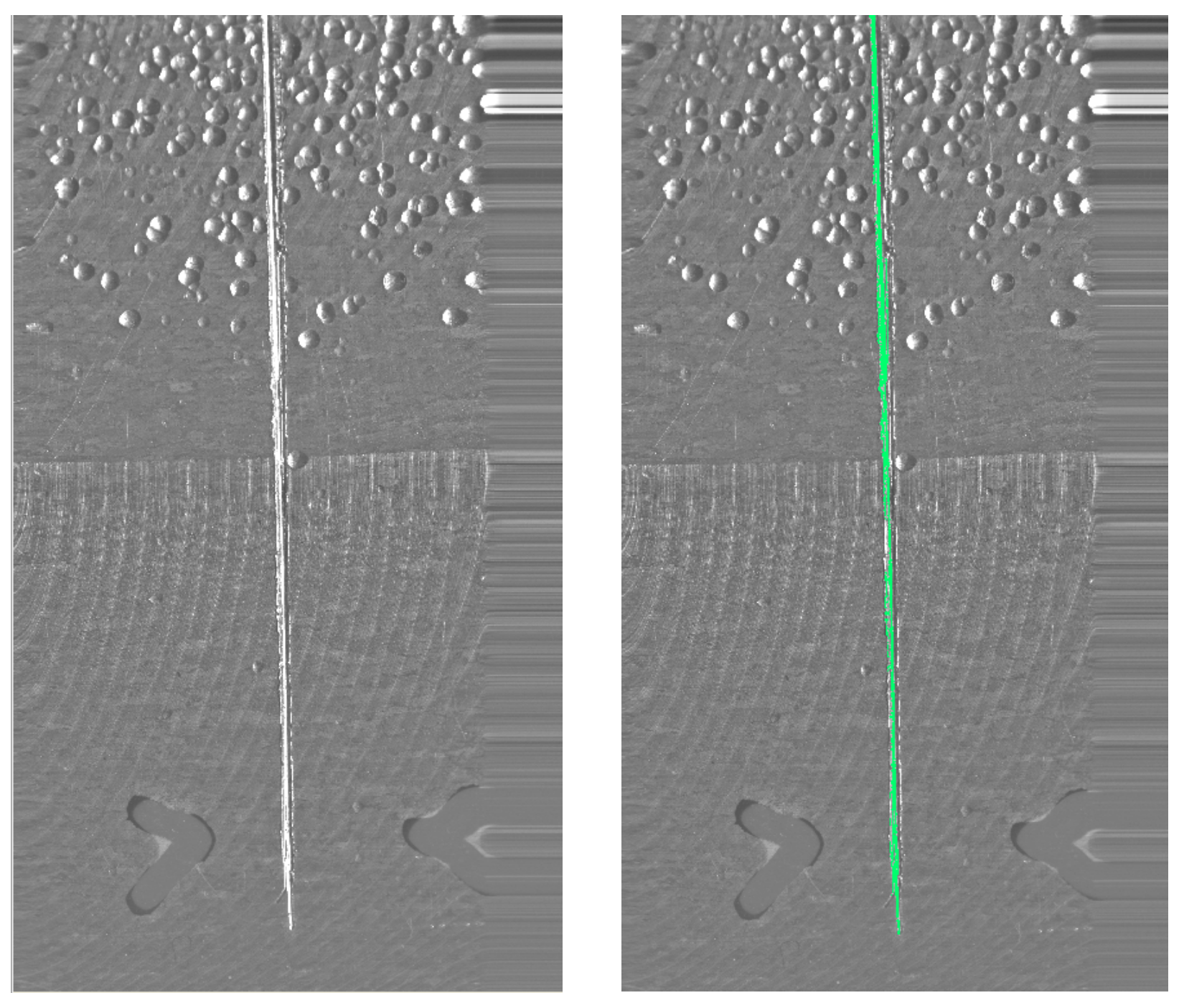
Automated Visual Inspection Of Industrial Parts

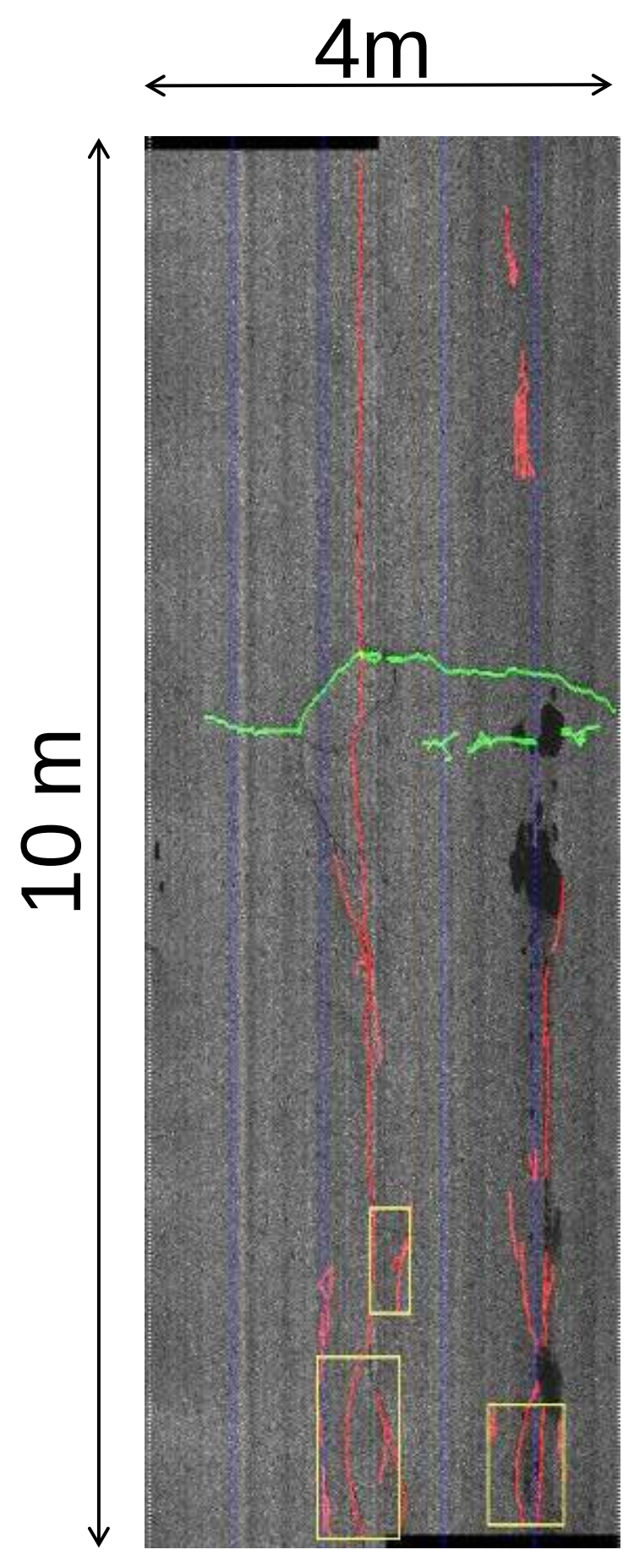
Adaptive, Cost-Optimal Defect detection

- We propose an original method to replace the dye penetrant inspection using toxic chemicals.
- The technique is fully adaptive and can detect fatal defects and ignore benign anomalies.
- Optimal algorithms have been developed to limit the processing time.

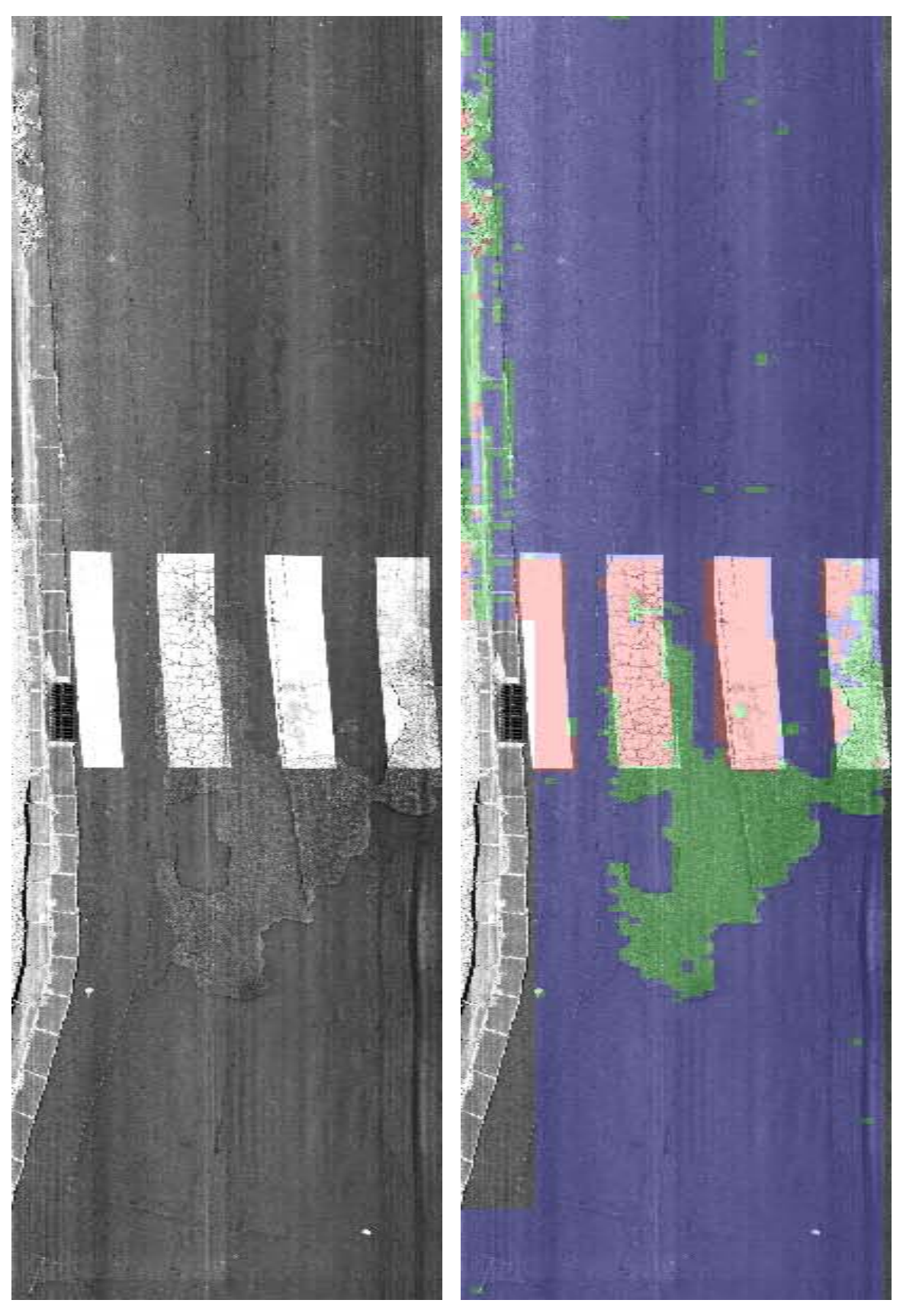
1. *Morard V, Dokládal P, Decencière E, Parsimonious path openings and closings. IEEE TIP, 2014*
2. *Morard V, Dokládal P, Decencière E, One-dimensional openings, granulometries and component trees in O(1) per pixel. JSTSP, 2012*

contact: Petr.Dokladal@mines-paristech.fr


crack detected defect



cracks


strippings

Paved-Road Aging Evaluation

Detection and analysis of cracks and strippings.

- Open and sealed cracks are separately detected and categorized according to : width, length, grouping and position and the cumulative length if reported for each category.
- Asphalt strippings are detected by means of texture analysis and classification.

E. Coquelle, J.-L. Gautier, P. Dokládal. Automatic Assessment of a Road Surface Condition, Surf 2012.

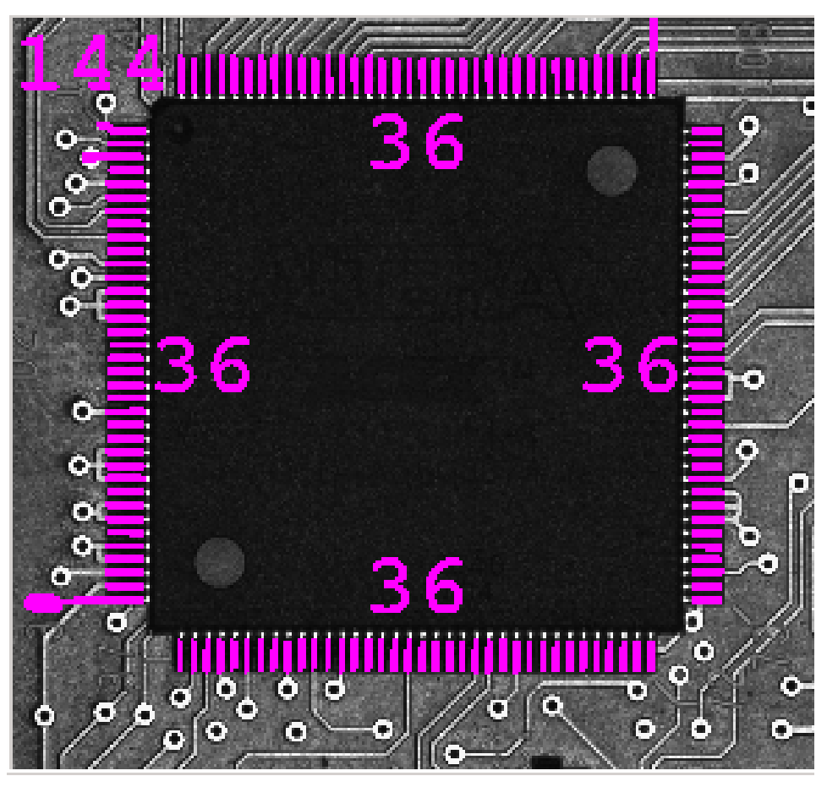
contact: Petr.Dokladal@mines-paristech.fr

Automated Visual Inspection of Electronic Cards

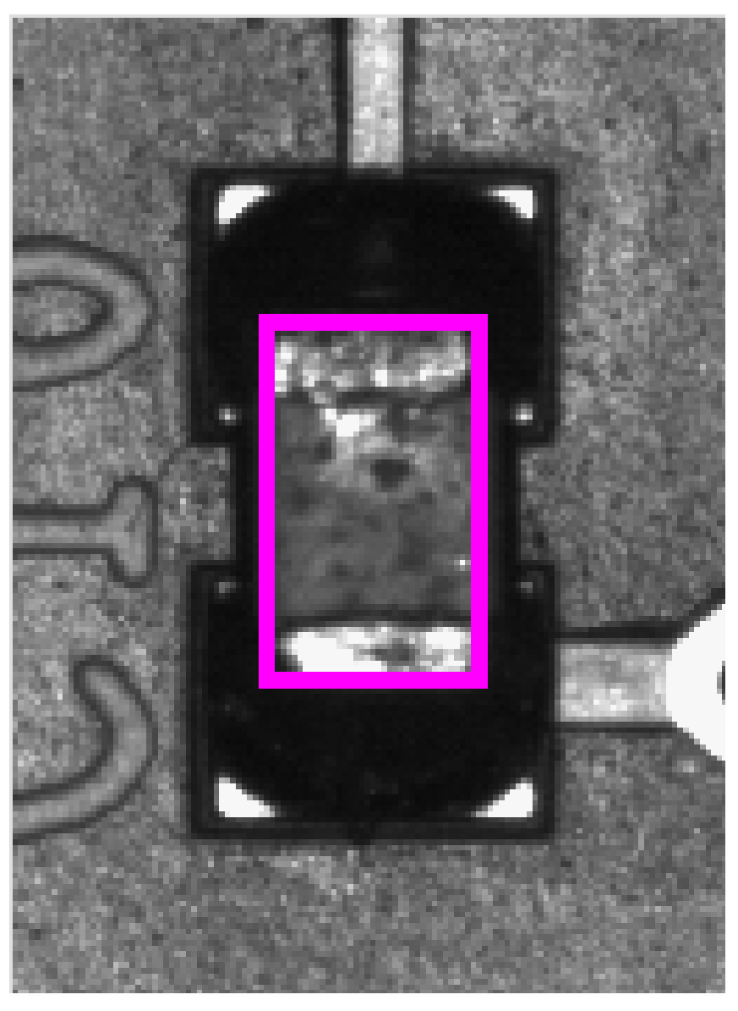
Detection of anomalies in IC mounted PCB cards

- Automatic detection of various IC housings.
- Detection of incorrectly placed or missing IC.
- Automatic detection of IC leads.

contact: Serge.Beucher@mines-paristech.fr



detected IC leads


detected IC housing

26-27 mars 2014 Numérique: Grande échelle et complexité

TELECOM SudParis
Institut Mines-Télécom

OpenETCS Project

s@movar UMR 5157

Authors

Phd student
João Santos

Post-Doc
Huu-Nghia Nguyen

Professor
Ana Cavalli

Context

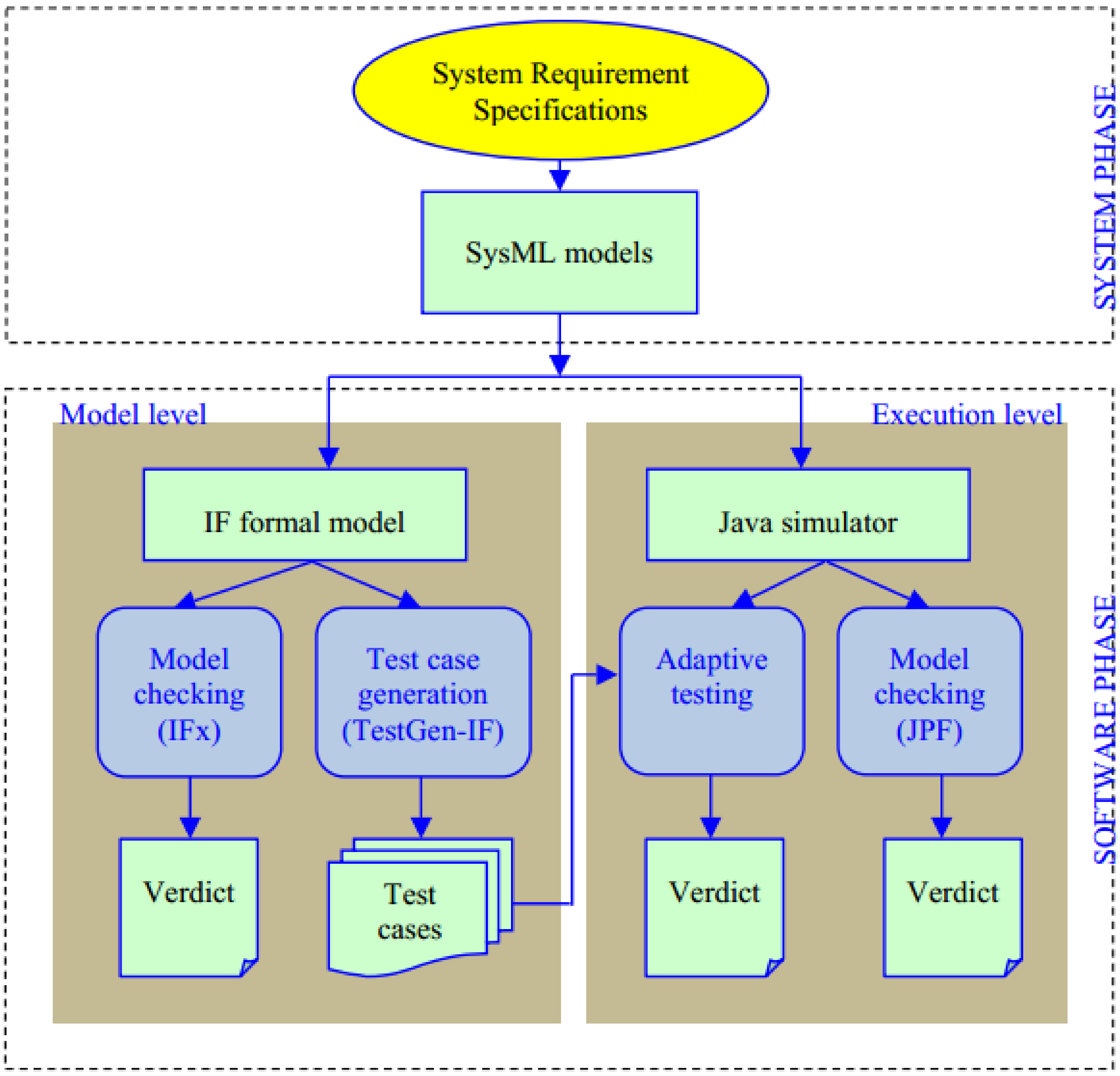
ITEA 2
INFORMATION TECHNOLOGY FOR EUROPEAN ADVANCEMENT

Objectives

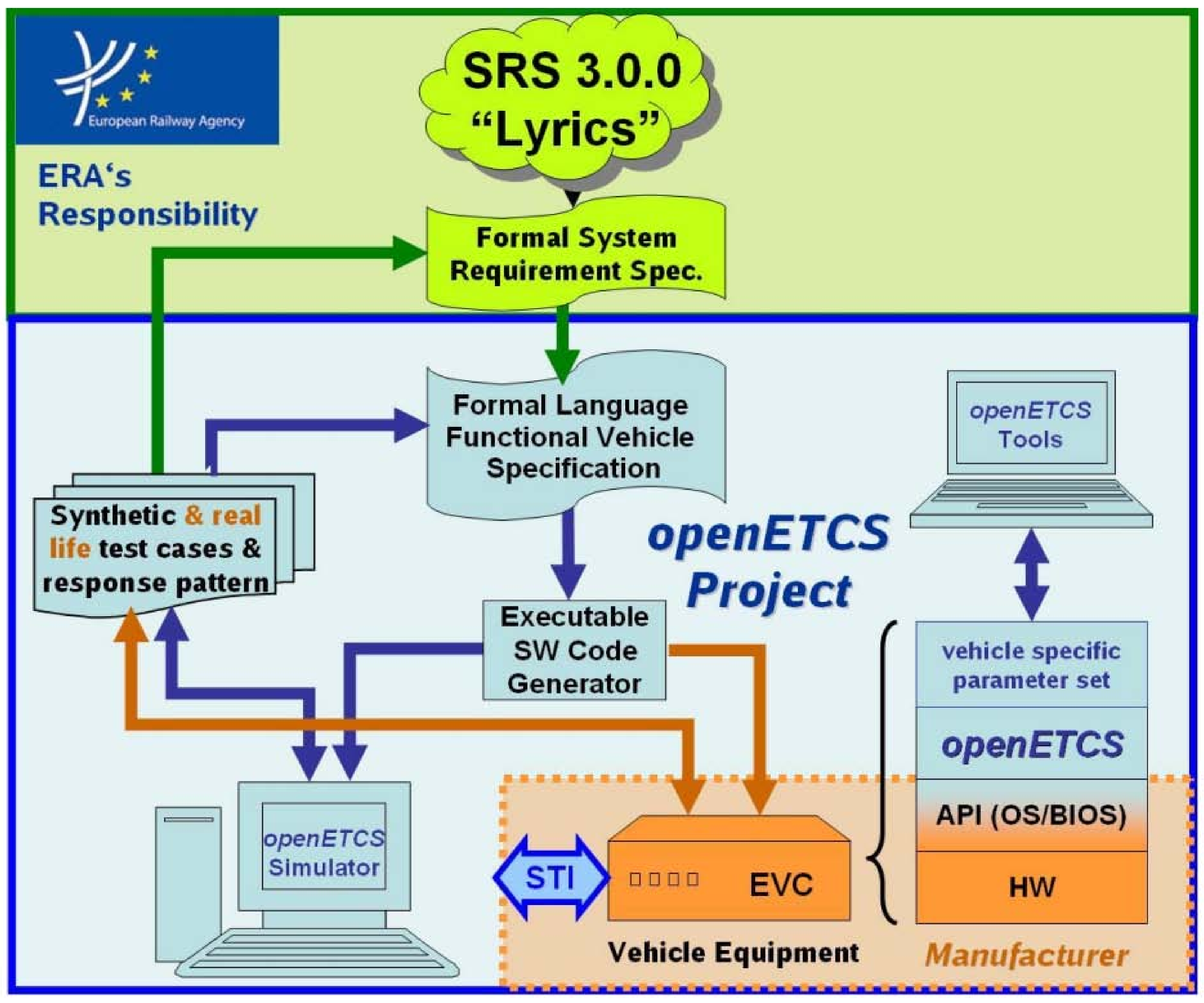
The main objective of the OpenETCS project is to develop an integrated modeling, development, validation and testing framework for leveraging the cost-efficient and reliable implementation of the European Train Control System (ETCS).

Our approach

Our work focuses on the validation and verification of an ETCS formal model. We resort to Model Checking, Simulation and Testing to achieve this goal.



OpenETCS Architecture



What is new?

❑ OpenETCS is based on Open Standards at all levels, including hardware and software, interfaces definition, design tools , verification and validation as well as embedded control software.

❑ The avionics sector has already developed its own source tools chain and created and eco-system. The similarity between the requirements of the railway and aviation safety equipment, make such tool chains a good basis for the project.

Partners