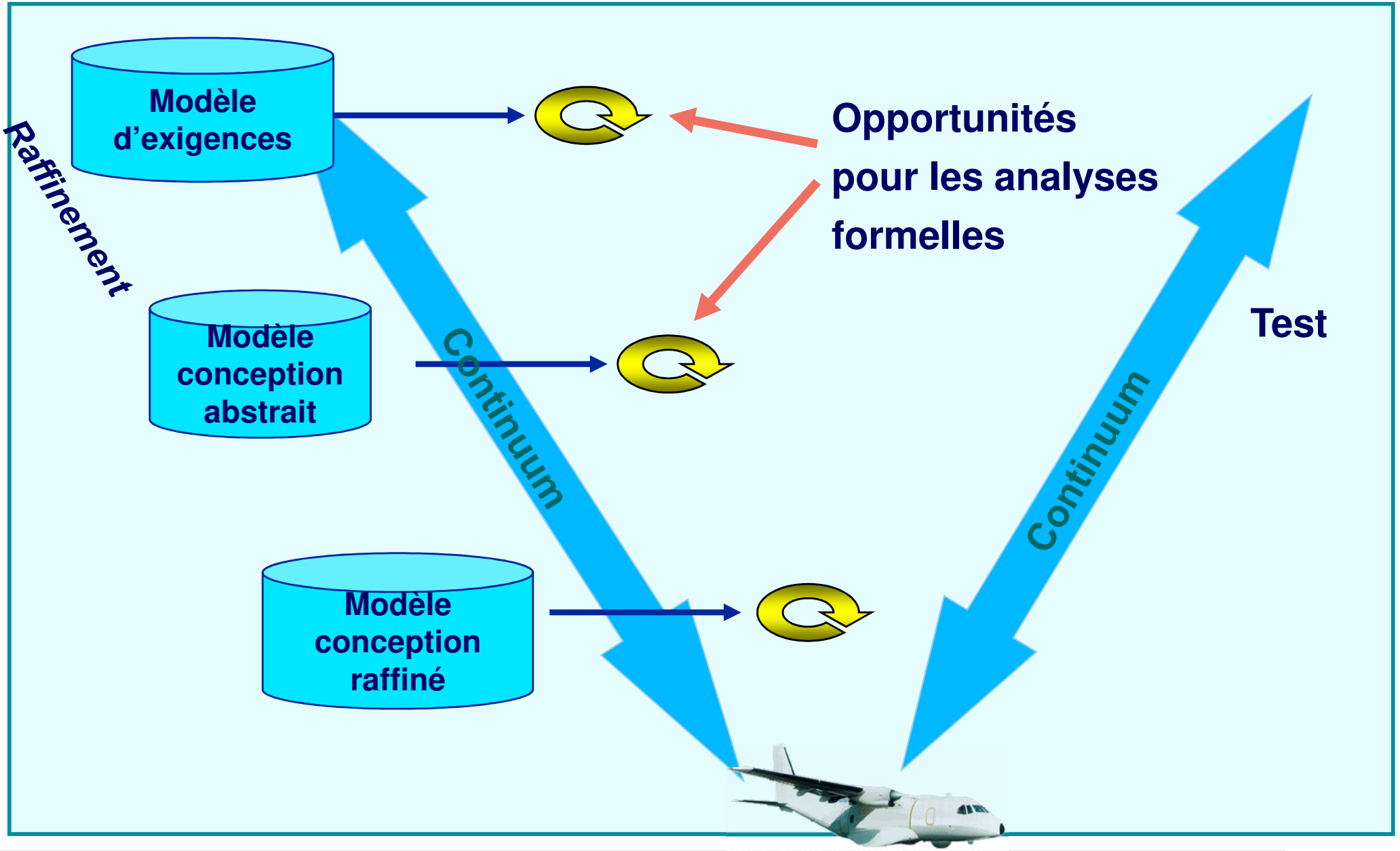
Approche pour la Vérification Formelle de Propriétés : Application au Développement Industriel de Logiciels Embarqués

Philippe Dhaussy

Univ. Européenne de Bretagne
Lab-STICC
UMR CNRS 6285
ENSTA-Bretagne, Brest.

philippe.dhaussy@ensta-bretagne.fr

Processus IDM et Analyses Formelles

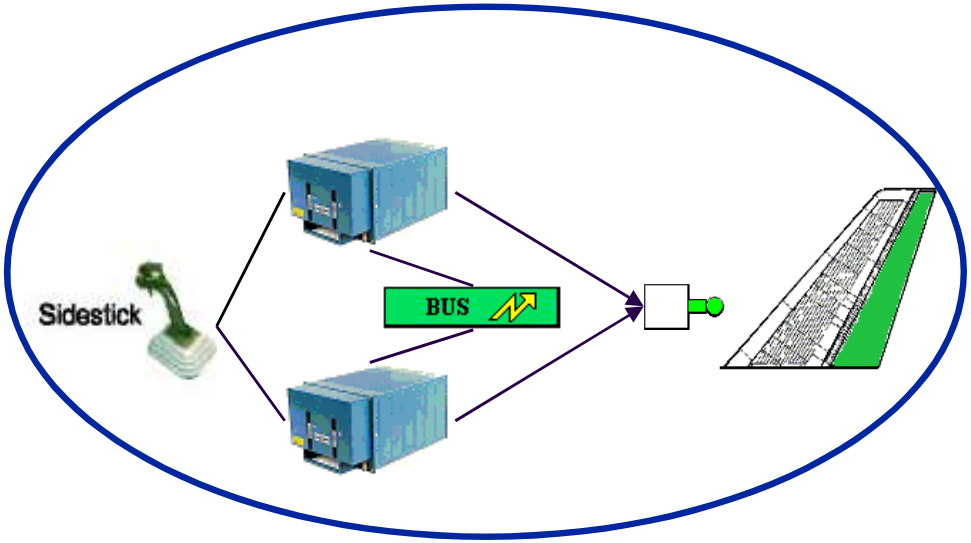


Plan

- Motivations
- Travaux, résultats, retour d'expérience
- Perspectives

Spécification des architectures fonctionnelles

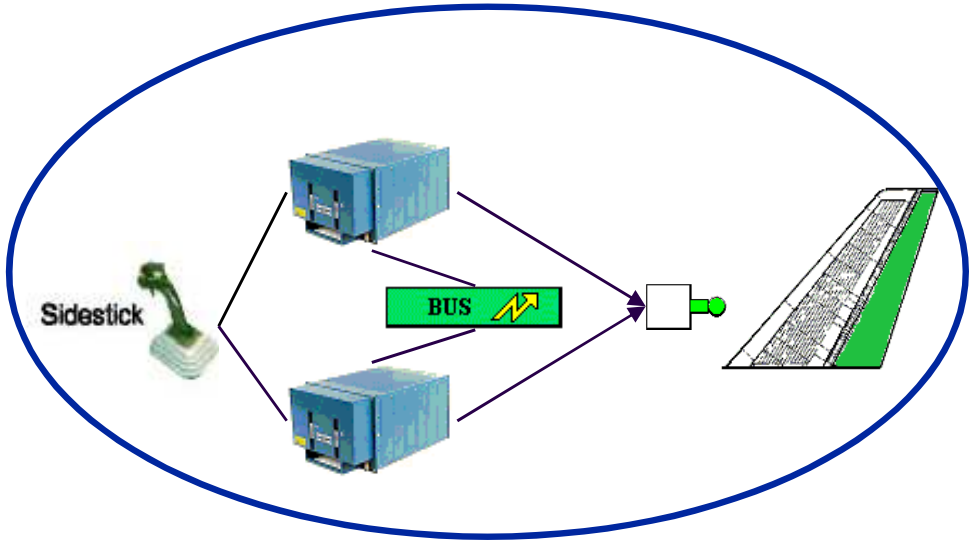
Modèle d'architecture fonctionnelle



Spécification des architectures fonctionnelles

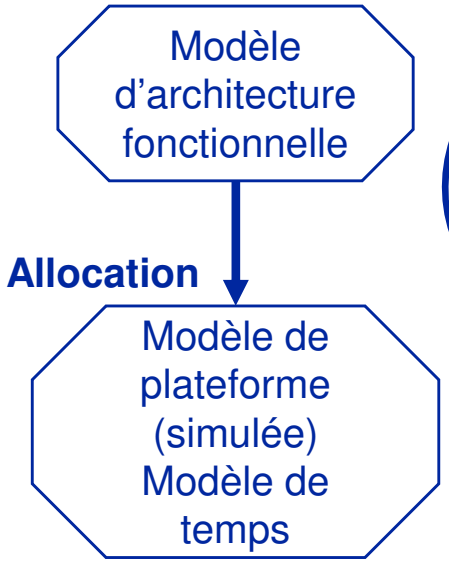
AADL, UML-MARTE (simplifié)

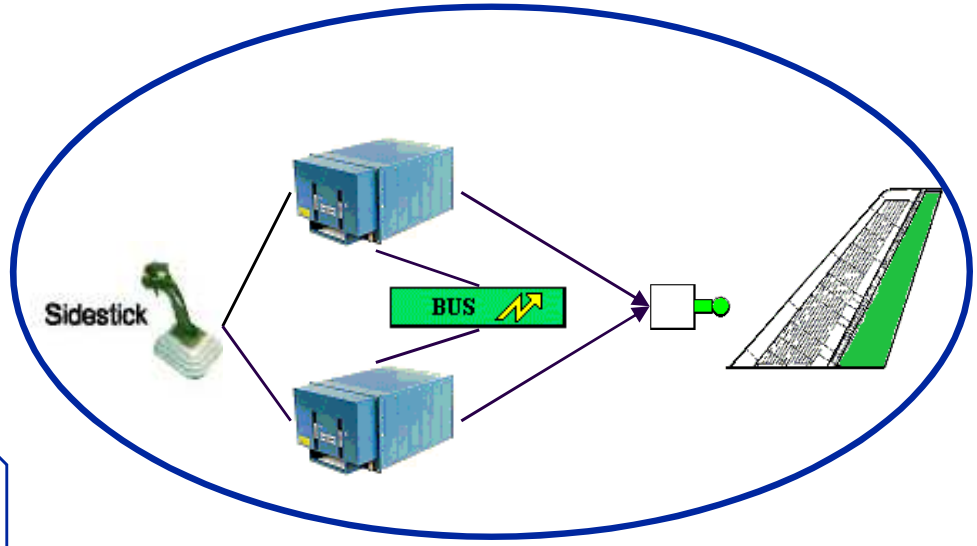
Modèle d'architecture fonctionnelle



Spécification des architectures fonctionnelles
Allocation sur des plateformes

AADL, UML-MARTE (simplifié)





Spécification des architectures fonctionnelles
Allocation sur des plateformes

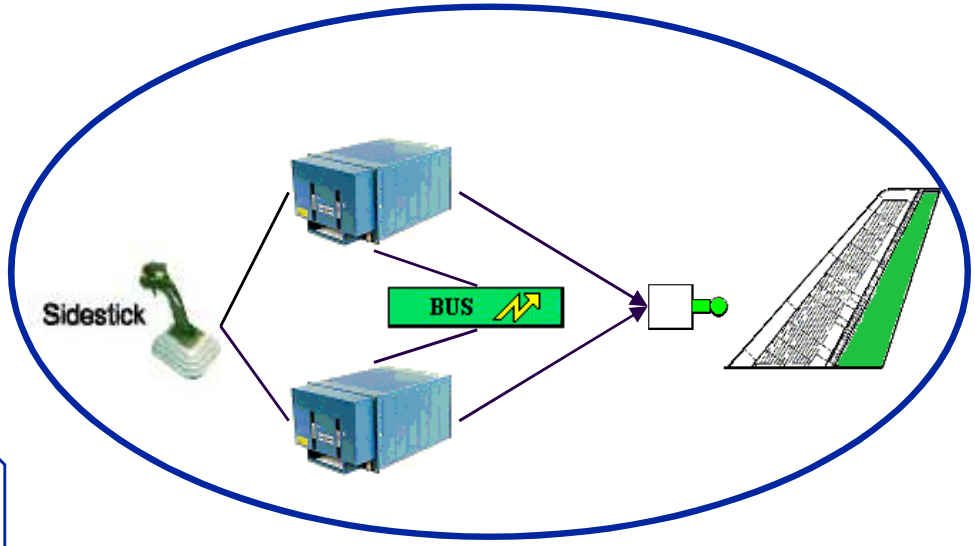
AADL, UML-MARTE (simplifié)

Modèle d'architecture fonctionnelle

Allocation

Modèle de plateforme (simulée)
Modèle de temps

AADL, UML-MARTE complet



Modèles d'analyse

AADL, UML-MARTE (simplifié)

Modèle d'architecture fonctionnelle

Extraction + enrichissement

Modèles dysfonctionnels

Modèles fonctionnels

Allocation

Modèle de plateforme (simulée) Modèle de temps

Modèles temporisés

Modèles de test

AADL, UML-MARTE complet

Analyse de défaillances

- Chaîne de Markov
- AltaRica

⇒ Calcul de probabilités d'événements redoutés

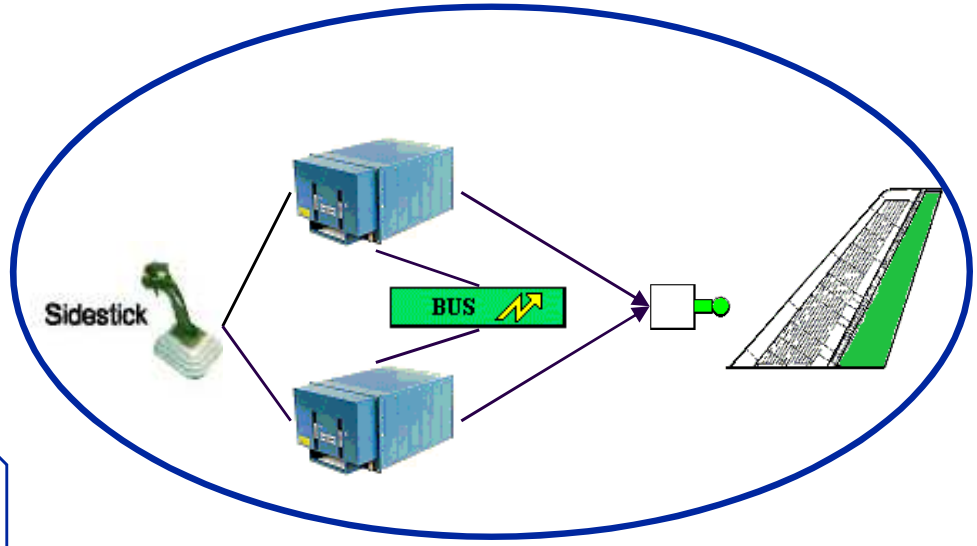
- automates,
- RdP, …

Analyse de propriétés fonctionnelles et non fonctionnelles

Modèles d'analyse

AADL, UML-MARTE (simplifié)

Modèle d'architecture fonctionnelle



Simulink

Modèle des lois de pilotage

Allocation

Modèle de plateforme (simulée) Modèle de temps

SCADE / ESTEREL (ou code C séquentiel)

Spécification des mécanismes (logiciels) de détection et de reconfiguration en cas de panne

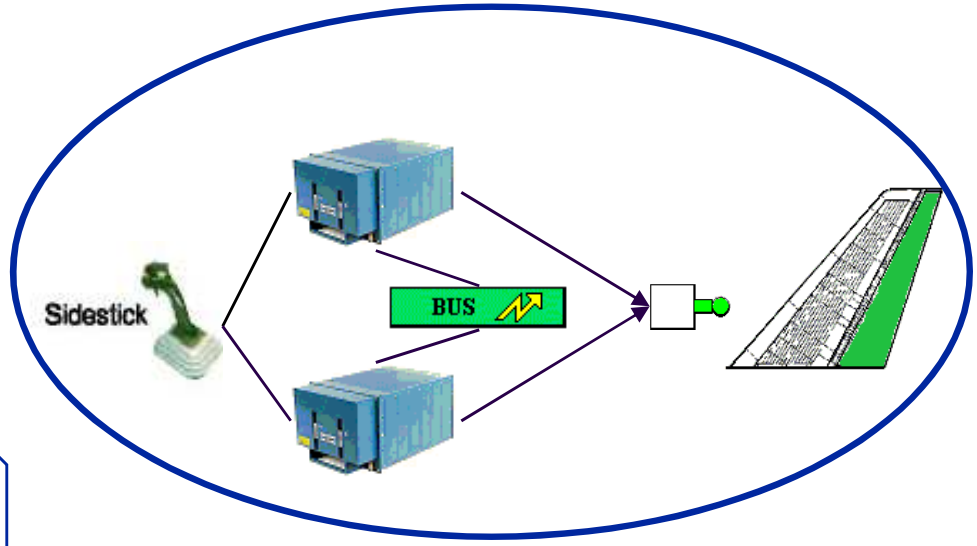
AADL, UML-MARTE complet

- Multi tâche + sémantique synchrone
- Time Triggered
- Interruptions, préemption

Modèles d'analyse

AADL, UML-MARTE (simplifié)

Modèle d'architecture fonctionnelle



Simulink

Modèle des lois de pilotage

Allocation

Modèle de plateforme (simulée) Modèle de temps

SCADE / ESTEREL (ou code C séquentiel)

Spécification des mécanismes (logiciels) de détection et de reconfiguration en cas de panne

AADL, UML-MARTE complet

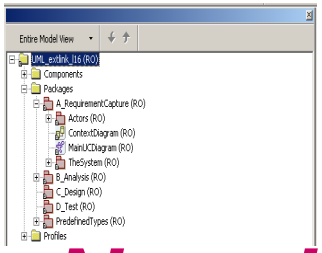
- Multi tâche + sémantique synchrone
- Time Triggered
- Interruptions, préemption

Génération / production de codes automatique correcte par construction

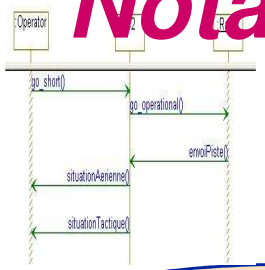
Codes

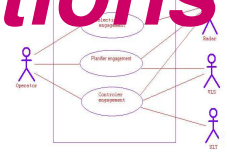
Plateforme Cible

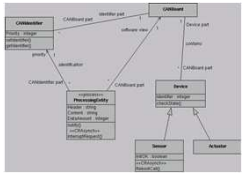
Enjeux pour les analyses formelles en contexte industriel

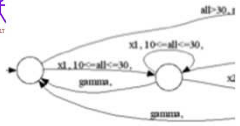


Notations









User Models
(UMLx, AADL, SDL, …)

Exigences

Processus

Comment intégrer des techniques d'analyse formelle dans l'ingénierie des exigences et du logiciel ?

Formalisation des contextes environnementaux et des propriétés
Réduction de l'explosion combinatoire

Techniques, Outils

Model-Checkers

[« *Research Directions in Requirements Engineering* », Cheng & Atlee, FOSE'07]

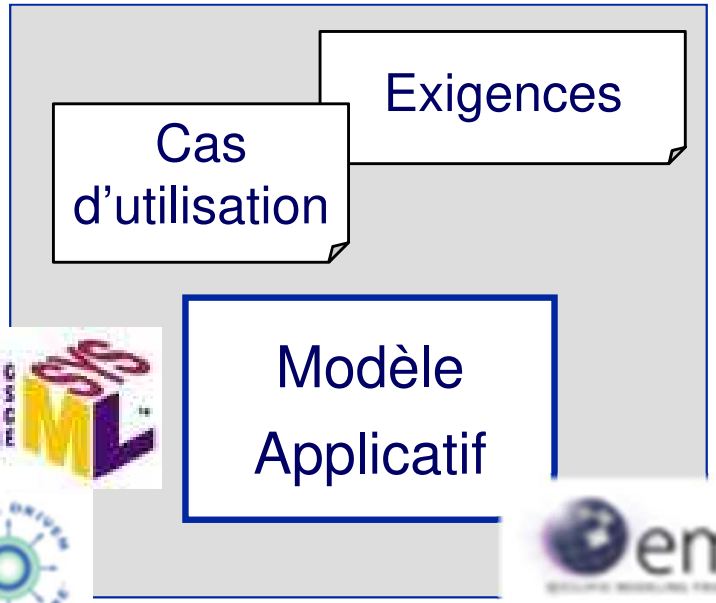
Requirements Tasks	Requirements Technologies		
	Notations	**Methodologies, Strategies, Advice**	**Techniques, Analyses, Tools**
Elicitation	Goals [19, 109, 173] Policies [18] Scenarios [1, 32, 47] Agents [106, 183] Anti-models [157, 166, 174] Nonfunctional requirements [28, 69]	Identifying stakeholders [152] Metaphors [133, 136] Persona [9, 34] Contextual requirements [33, 160] Inventing requirements [117]	Animation [84, 115, 170] Simulation [164] Invariant generation [93]
Modeling	Object models [89] Behavioral models [92, 167] Domain descriptions [11] Property languages [14, 50, 105] Notation Semantics [59, 120, 125, 163]	RE reference model [75, 77, 131] Model elaboration [169] Viewpoints [128, 155] Patterns [49, 54, 90, 99, 171] Modeling facilitators [6, 31, 72, 97, 129] Formalization heuristics [18, 67] Methodologies [15]	Model merging [147, 165] Model synthesis [3, 39, 107, 168, 182] Model composition [80]
Requirements Analysis		Negotiation [87] Aligning requirements with COTS [5, 144] Conflict management [143]	Linguistic analysis [16, 27, 176] Ontologies [95] Checklists [177] Consistency checking [58, 83, 123] Inspections [60, 132] Conflict analysis [25, 79] Obstacle analysis [114, 175] Risk management [62] Impact analysis [101] Causal order analysis [12] Prioritization [122] Variability analysis [74, 108, 110] Requirements selection [139, 159]
Validation & Verification	Model formalisms [22, 51]	???	Simulation [164] Animation [84, 115, 170] nvariant generation [93] Model checking [26, 55, 158] Model satisfiability [89]
Requirements Management	Variability modeling [21, 38, 140, 150]	Scenario management [2] Feature management [179] Global RE [44]	Traceability [30, 81, 146, 151] Stability analysis [23]

Motivation : Intégration des méthodes formelles dans les processus d'ingéniérie

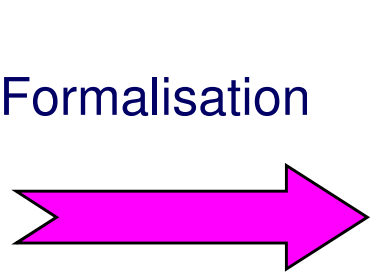
Notations

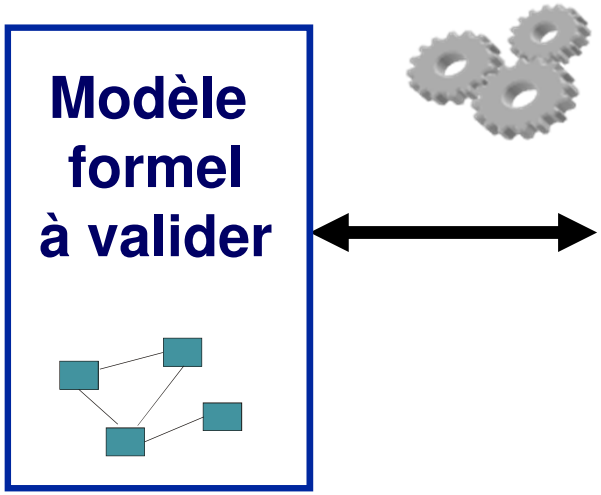
Processus

Outils



Formalisation





Objectif : *Etudier les hypothèses et les conditions opérationnelles pour rendre possible l'ntégration des méthodes dans les processus.*

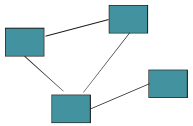
Disposer de modèles formels pour les outils de vérification

Principe de vérification de propriétés

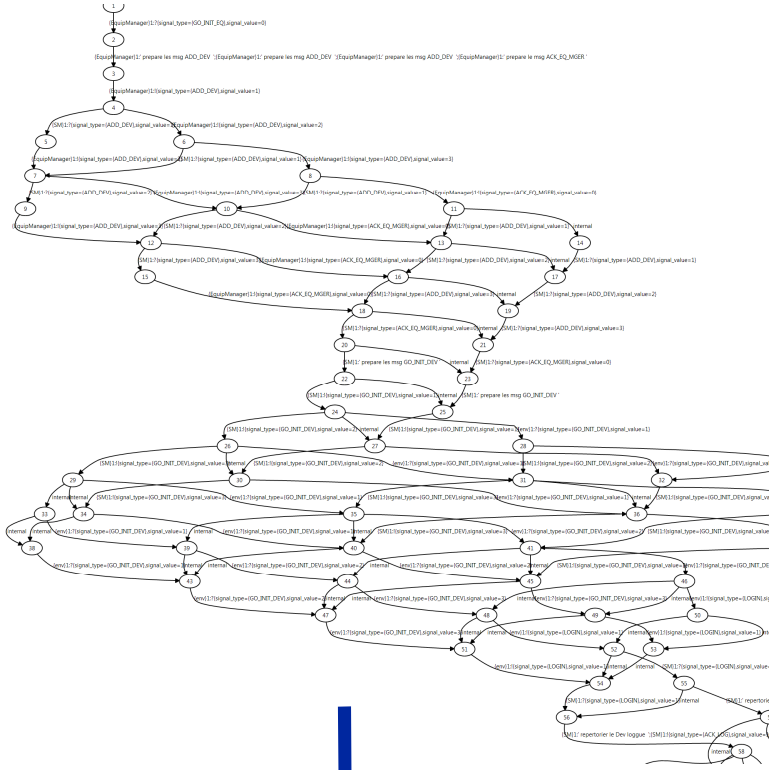
Système de transitions étiquetées

Propriétés formelles

Modèle formel à valider



Simulateur, Model-checker



trace

Verdict

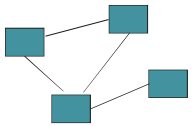
Analyse d'accessibilité

Principe de vérification de propriétés

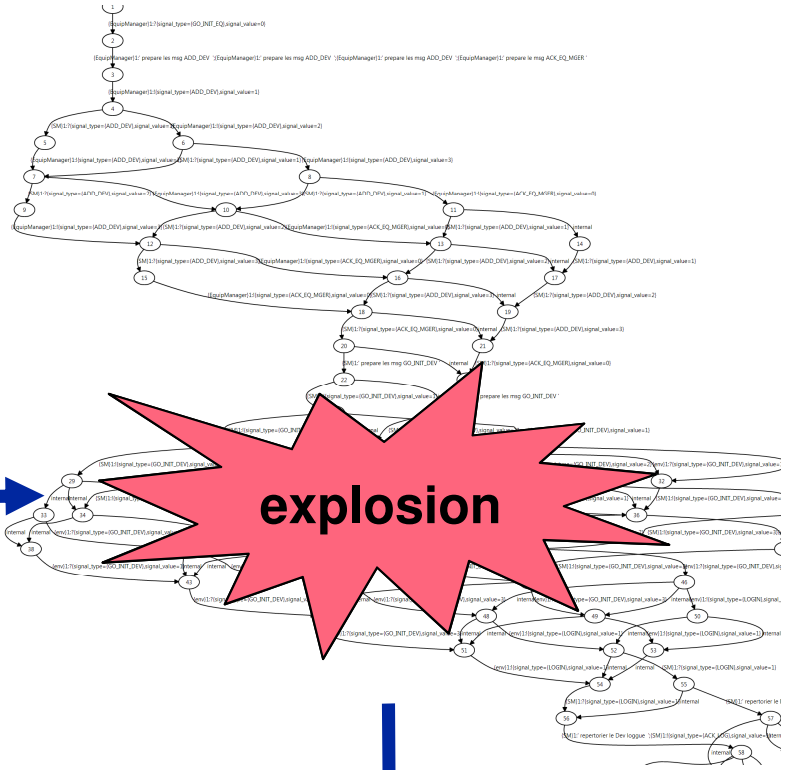
Système de transitions étiquetées

Propriétés formelles

Modèle formel à valider



Simulateur, Model-checker



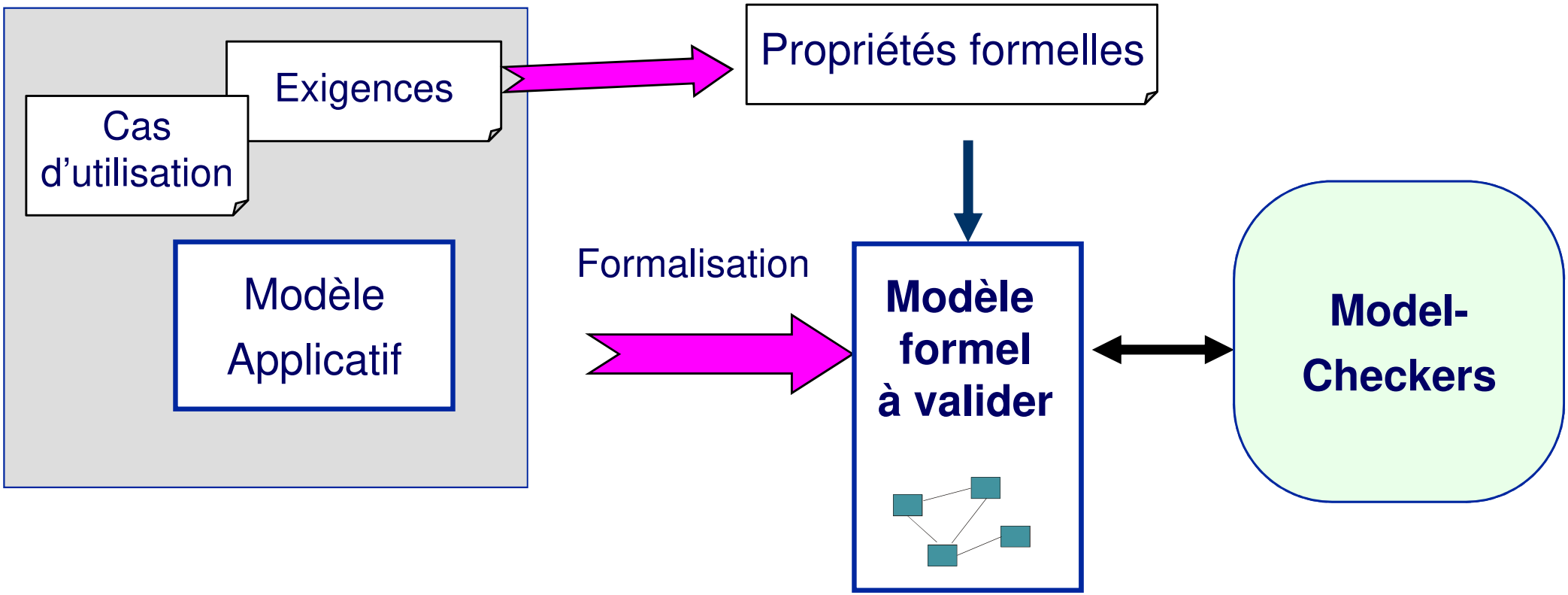
explosion

trace

Verdict

Analyse d'accessibilité

Difficulté: formalisation des propriétés



Gap entre le modèle à valider et le modèle formel

Logique temporelle : non adéquat

Difficulté: formalisation des propriétés

SYST-DP-REQ-6

During initialization procedure, the SYST_DP shall associate a generic equipment identifiers to one or several role in the system (MainSensor, OtherSensor, IFF, Actuator, ...). It shall also associate an identifier to each console.
The SYST_DP shall send an evtEquipmentRole message, in preparation mode, for each connected generic equipment, to each connected console.
Initialization procedure shall end successfully, when the SYST_DP has set all the generic equipment identifiers and all console identifiers and all evtEquipmentRole message have been sent.

End

SYST-DP-REQ-8

Once initialization is achieved, the SYST_DP shall send to each console an evtCurrentMission with curMission set to IDLE, to set current mission to idle, followed by an evtCurrentActivity with curActivity to LOGIN and status to TRUE to activate login.

End

Lien : Contexte – propriété

SYST-DP-REQ-6

During initialization procedure, the SYST_DP shall associate a generic equipment identifiers to one or several role in the system (MainSensor, OtherSensor, IFF, Actuator, ...). It shall also associate an identifier to each console.
The SYST_DP shall send an evtEquipmentRole message, in preparation mode, for each connected generic equipment, to each connected console.
Initialization procedure shall end successfully, when the SYST_DP has set all the generic equipment identifiers and all console identifiers and all evtEquipmentRole message have been sent.

End

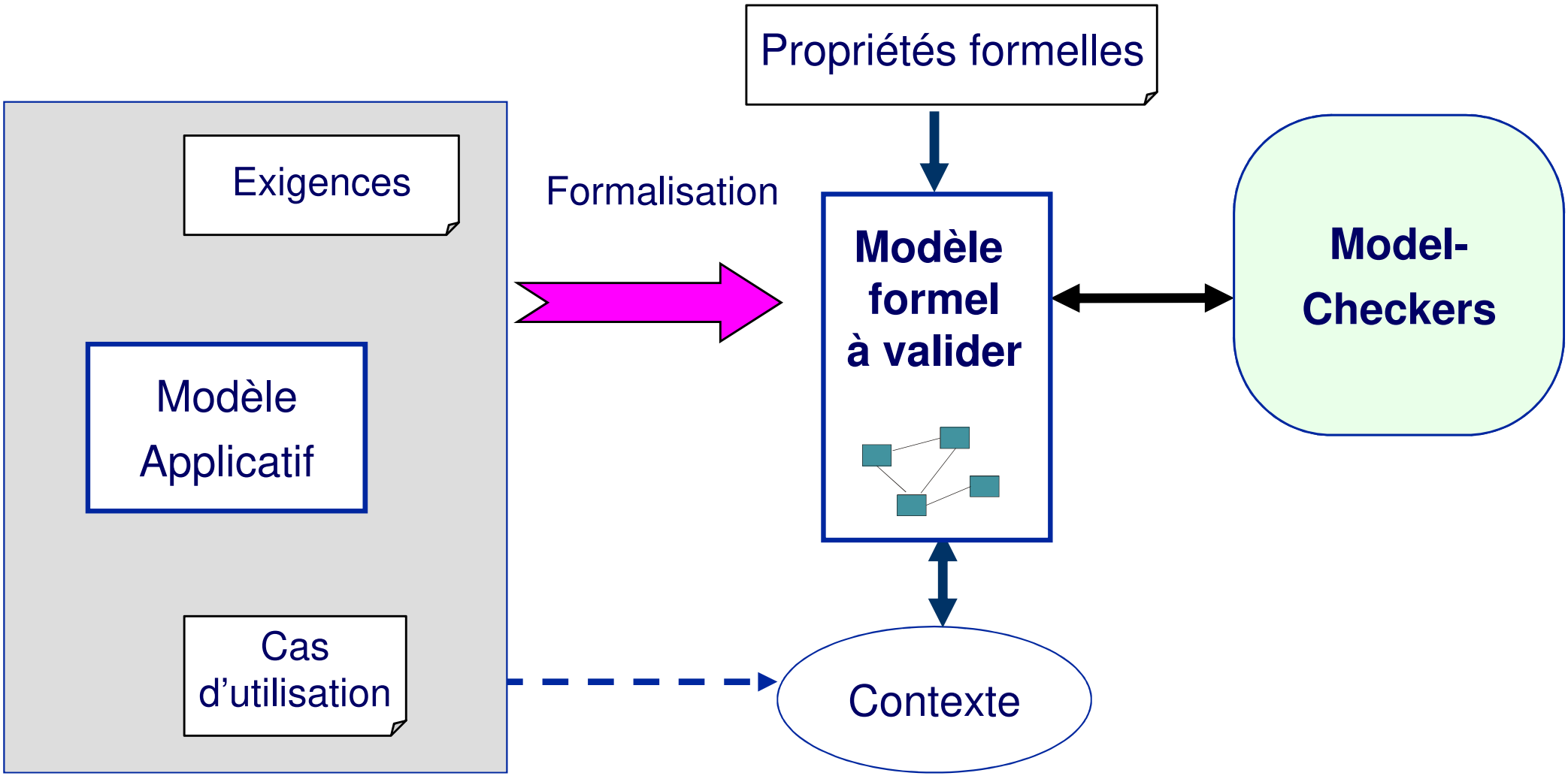
SYST-DP-REQ-8

Once initialization is achieved, the SYST_DP shall send to each console an evtCurrentMission with curMission set to IDLE, to set current mission to idle, followed by an evtCurrentActivity with curActivity to LOGIN and status to TRUE to activate login.

End

Contexte

Lien : Contexte – propriété



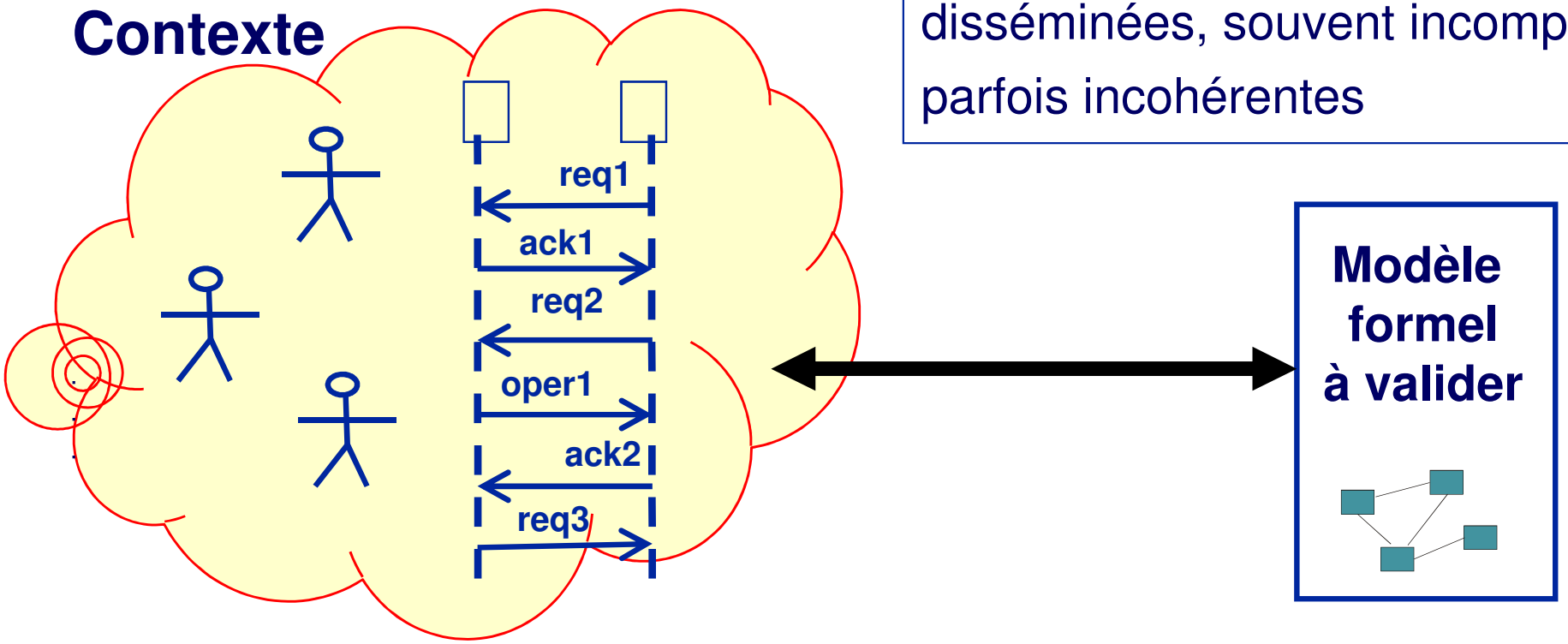
Vérifier les propriétés dans leur contexte

Expression des contextes

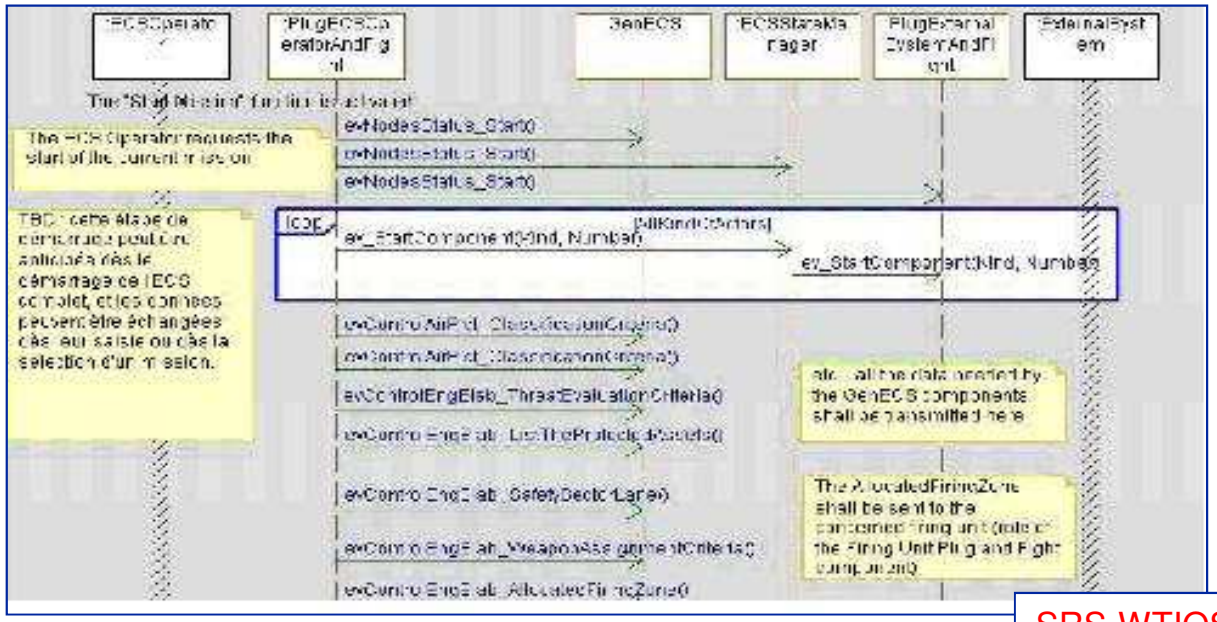
Représentent le comportement de l'environnement (Phases opérationnelles)

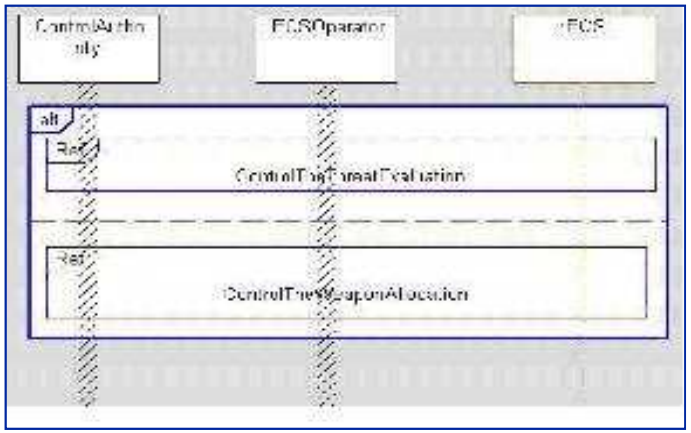
Initialisation, reconfiguration, modes dégradés, scénarios d'erreurs, etc.

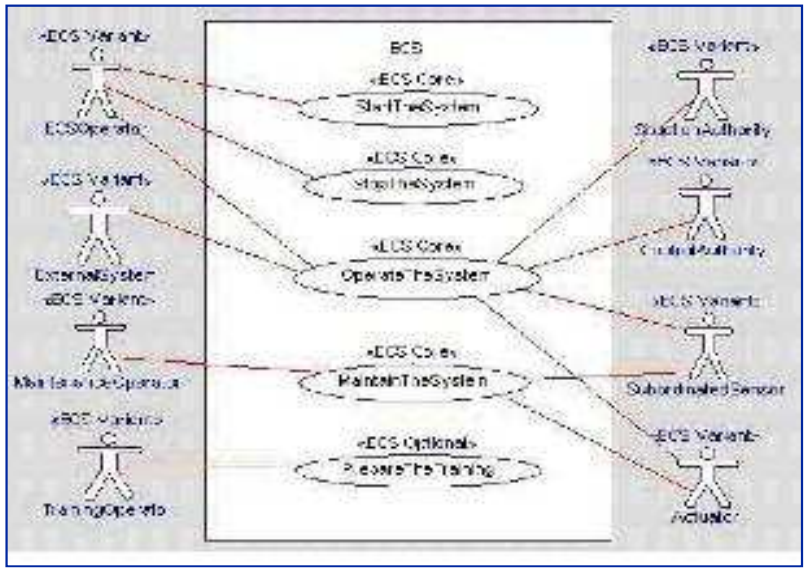
En réalité dans les spécifications :
Descriptions avec beaucoup d'implicite, disséminées, souvent incomplètes, parfois incohérentes



Expression des contextes

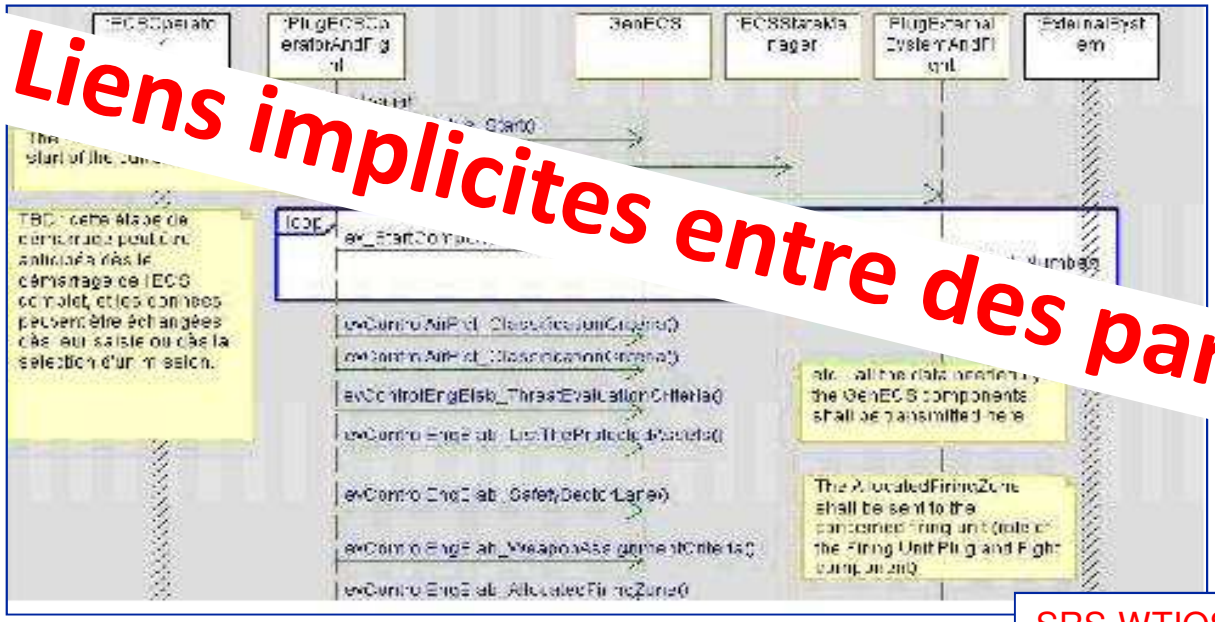


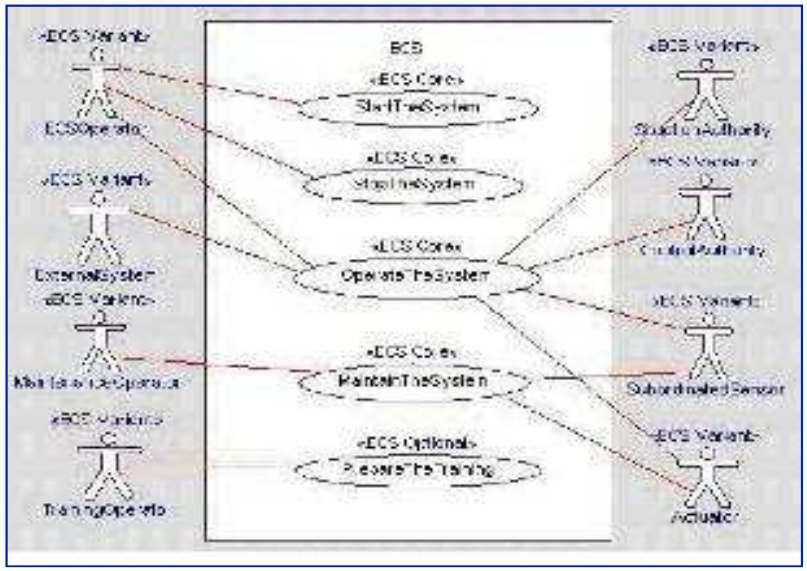




SRS-WTIOS-REQ-004
On receipt of a *MsgFieldMask* message from the COMM_WT, the WT_IOS shall set the WT_State to 'STANDBY' and transmit the *EvtTechnicalStateIos* message to the ECDP_DP with the following parameters:
equipmentId = equipmentId of the WT_IOS
roleId = roleId of the WT_IOS
state = STANDBY
If the requested WT_State is OPERATIONAL, the WT_IOS shall transmit the *MsgControlNetwork* message to the COMM_WT with the following parameters:
orderId
command = 'READY'
End Requirement

Expression des contextes

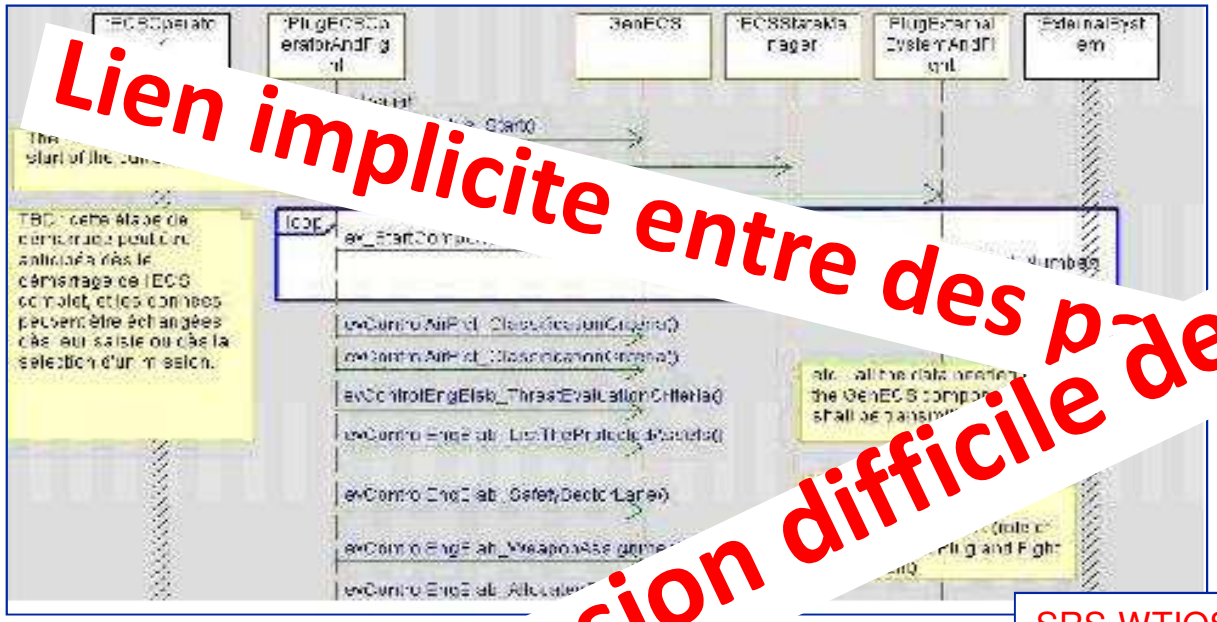


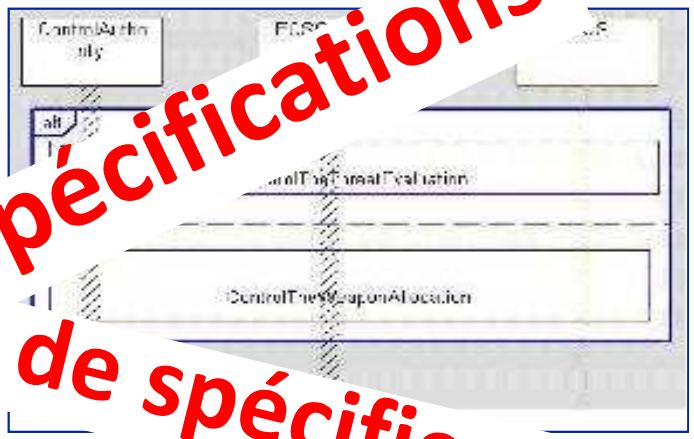


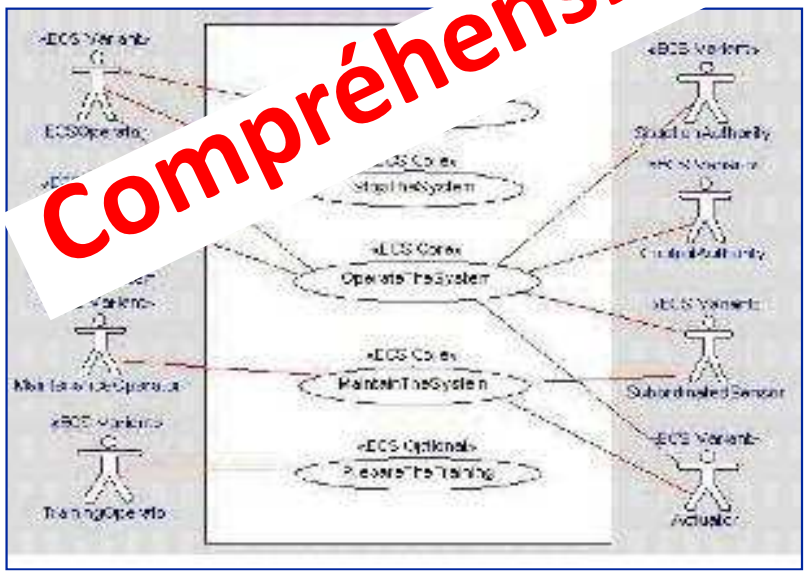
Liens implicites entre des parties de spécifications

SRS-WTIOS-REQ-004
On receipt of a *MsgFieldMask* message from the COMM_WT, the WT_IOS shall set the WT_State to ‘STANDBY’ and transmit the *EvtTechnicalStateIos* message to the ECDP_DP with the following parameters:
equipmentId = equipmentId of the WT_IOS
roleId = roleId of the WT_IOS
state = STANDBY
If the requested WT_State is OPERATIONAL, the WT_IOS shall transmit the *MsgControlNetwork* message to the COMM_WT with the following parameters:
orderId
command = ‘READY’
End Requirement

Expression des contextes





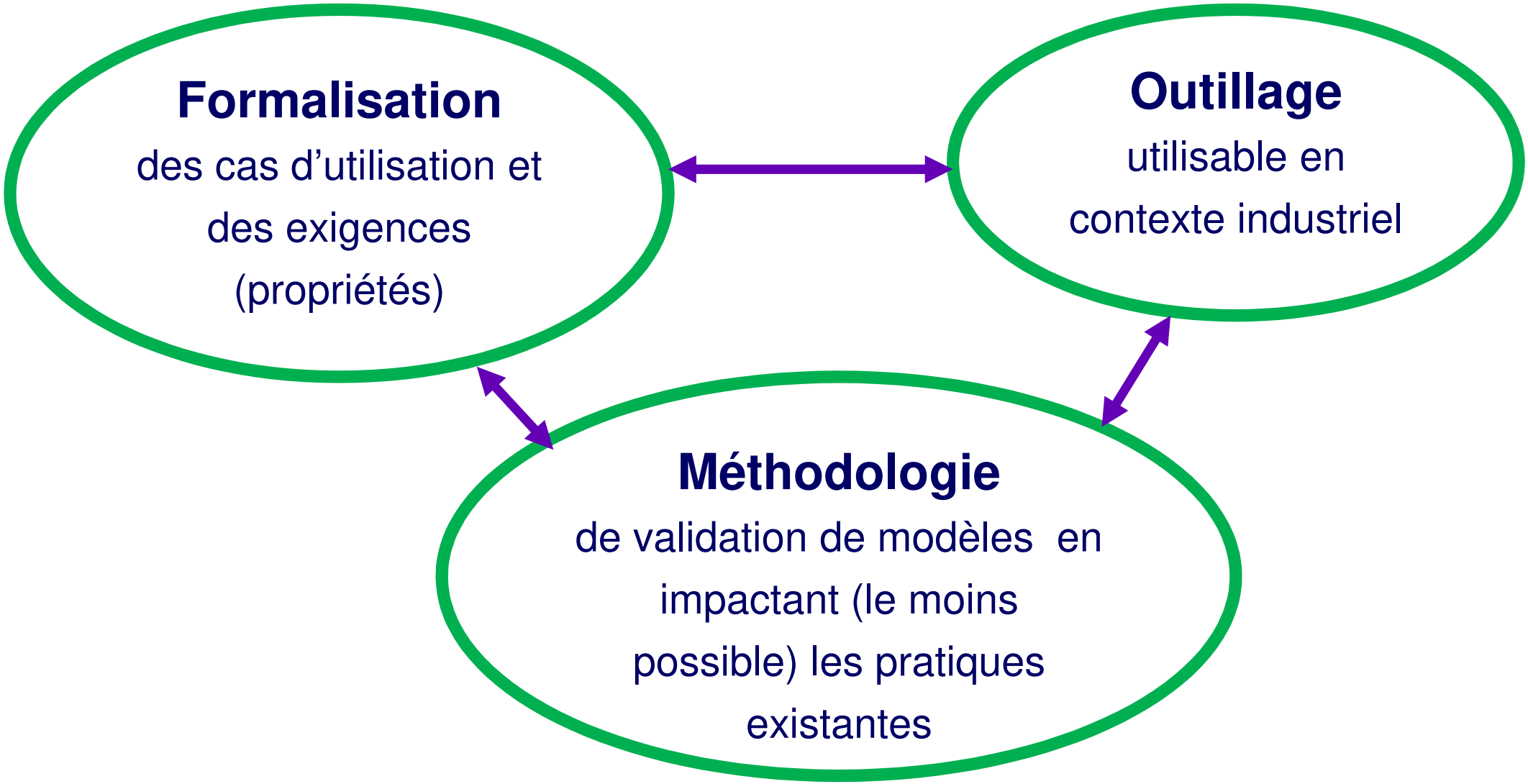


Lien implicite entre des p… …s de spécifications

Compréhension difficile des spécifications

SRS-WTIOS-REQ-004
On receipt of a *MsgFieldMask* message from the COMM_WT, the WT_IOS shall set the WT_State to ‘STANDBY’ and transmit the *EvtTechnicalStateIos* message to the ECDP_DP with the following parameters:
equipmentId = equipmentId of the WT_IOS
roleId = roleId of the WT_IOS
state = STANDBY
If the requested WT_State is OPERATIONAL, the WT_IOS shall transmit the *MsgControlNetwork* message to the COMM_WT with the following parameters:
orderId
command = ‘READY’
End Requirement

Une approche pragmatique



Plan

- Motivations
- Travaux, résultats, retour d'expérience
- Perspectives

Travaux de l'équipe

Observer-Based Prover

Context Description Language

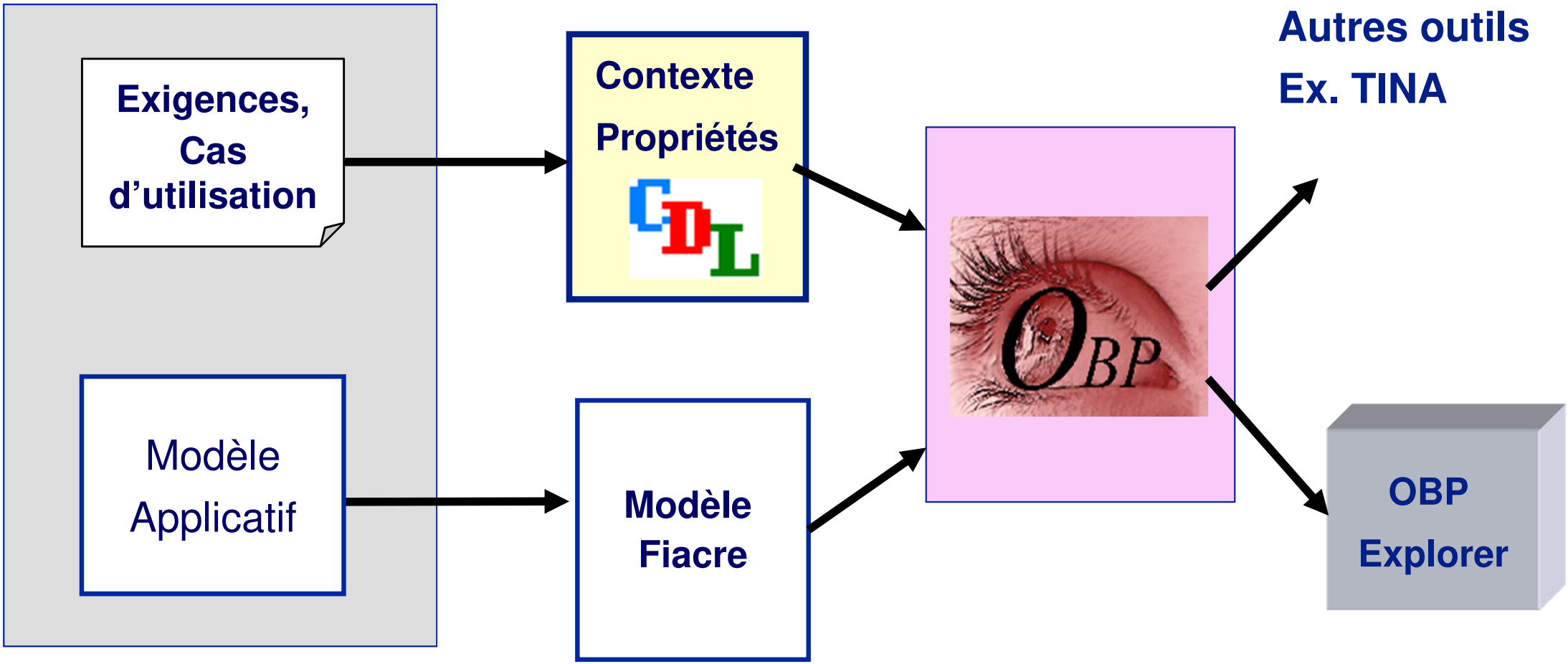
Expérimentations industrielles

UCM & Méthodologie

Optimisation, parallélisation OBP

Chaîne de transformation

L'outillage OBP et le langage CDL

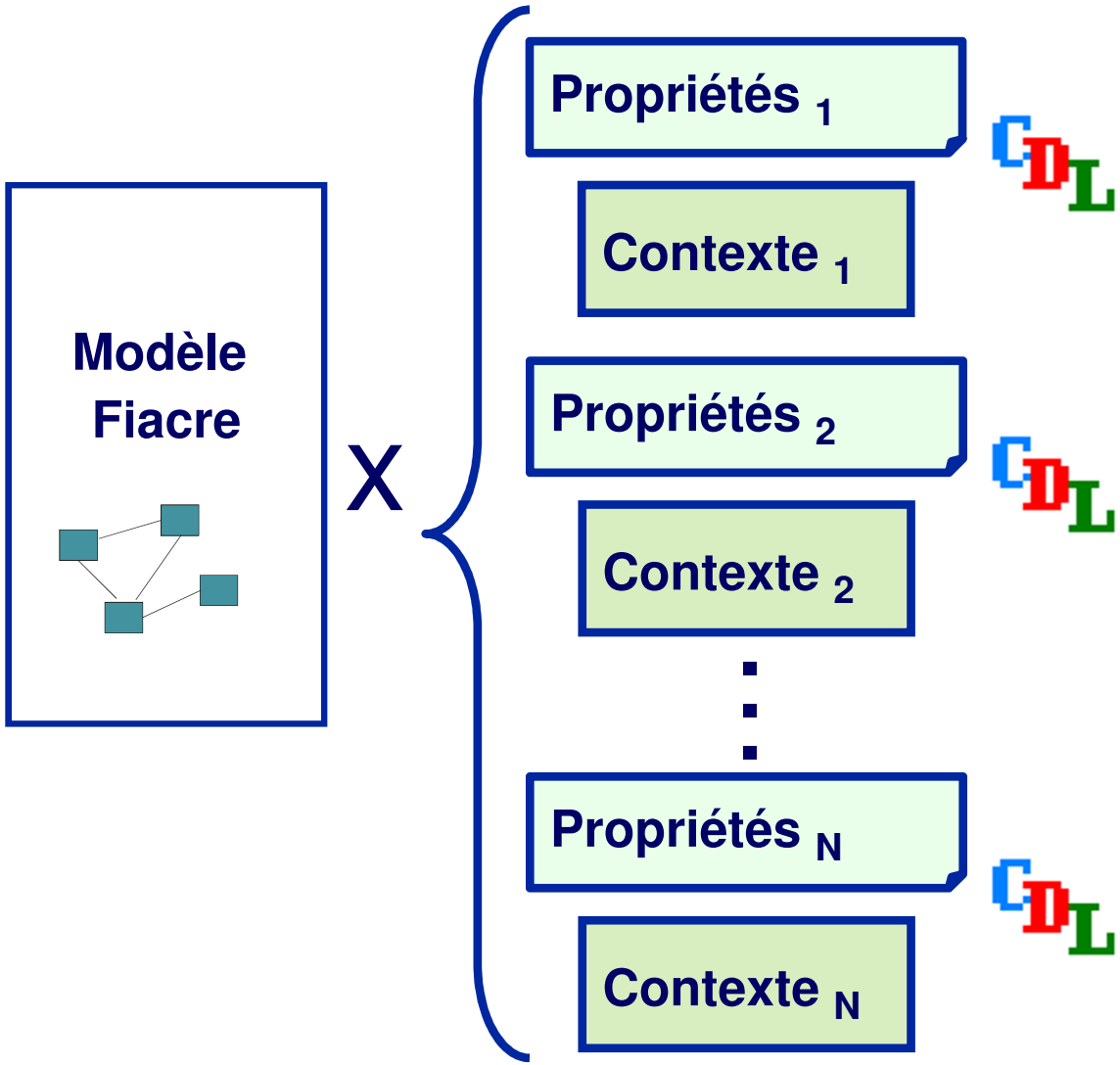


Diffusion : www.obpcdl.org

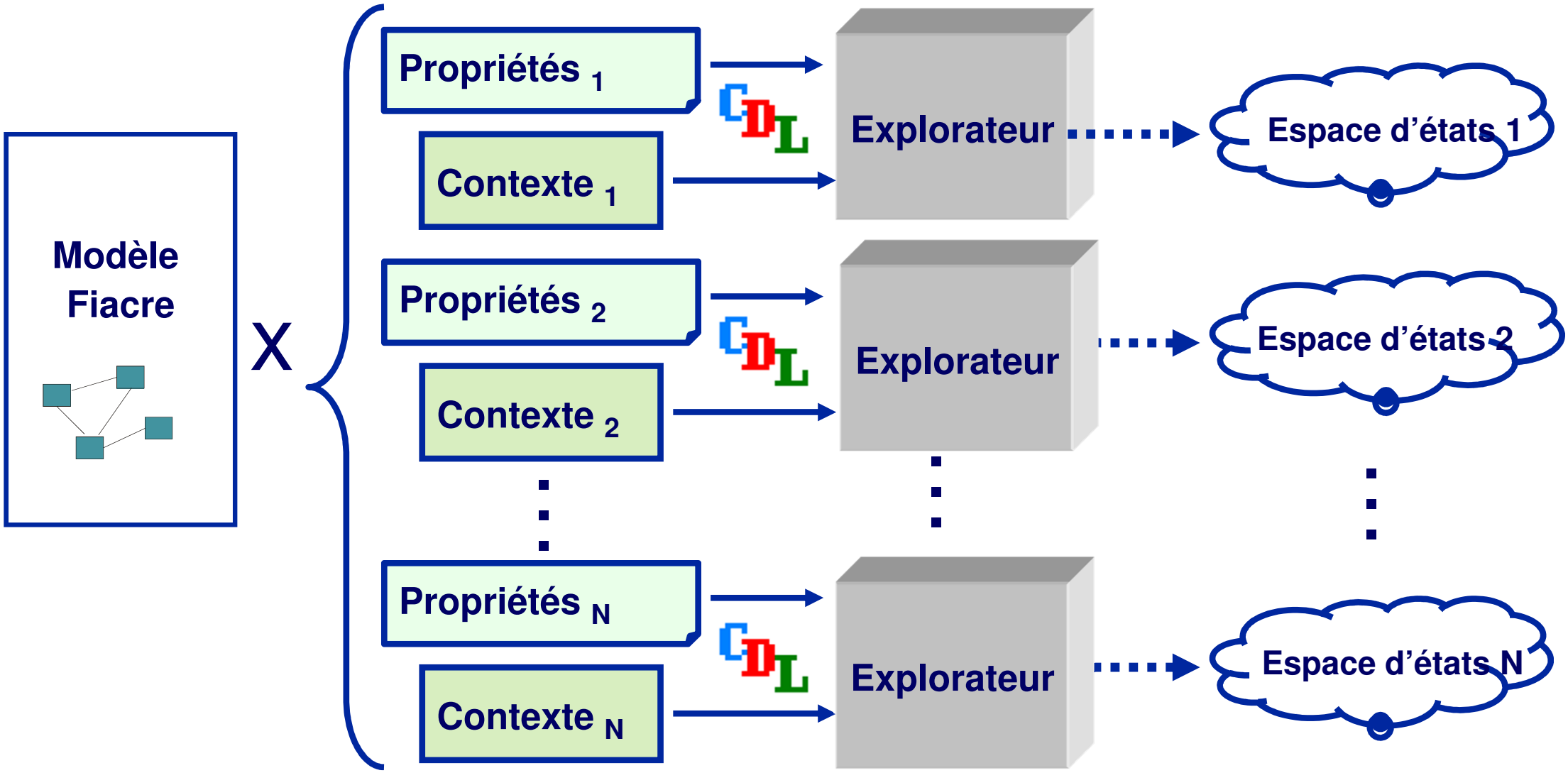
Langage 

- ***Gestion des contextes***
- ***Gestion des propriétés***

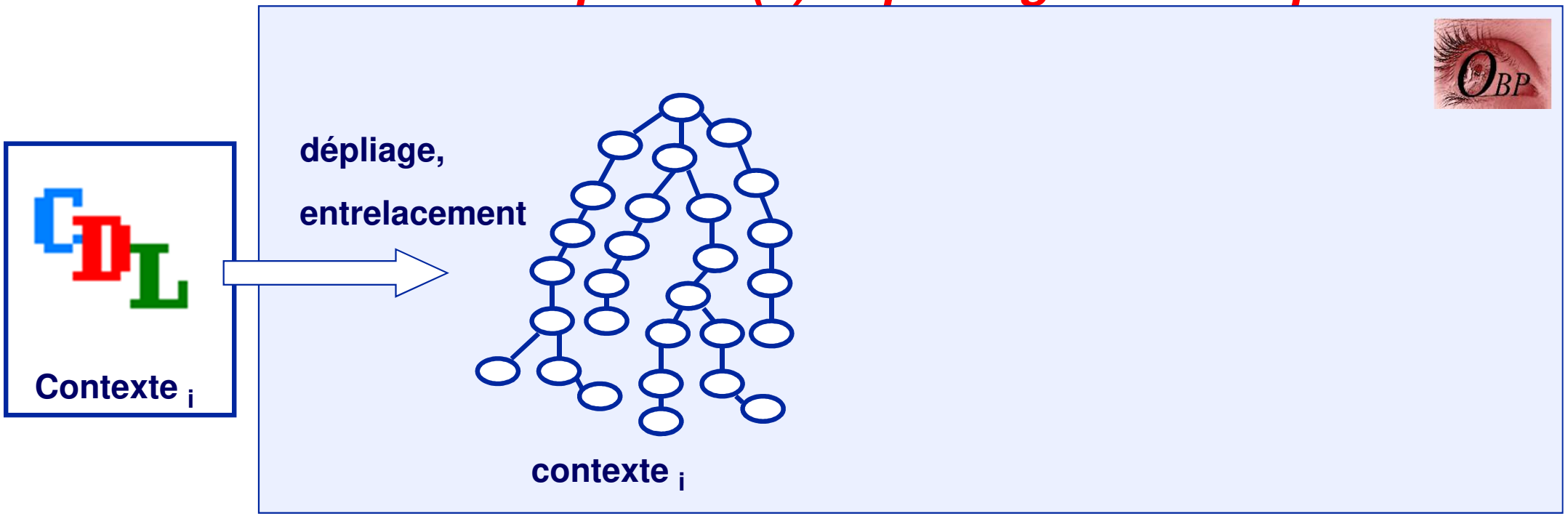
Réduction de la complexité (1) : Identification des contextes



Réduction de la complexité (1) : Identification des contextes

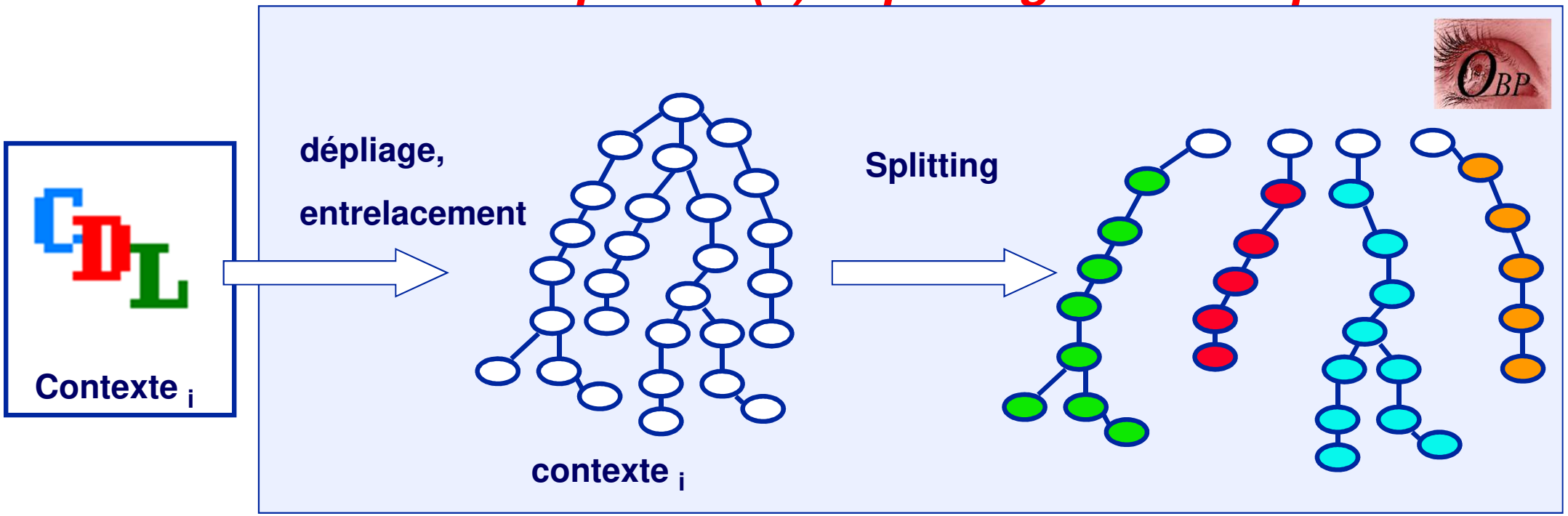


Réduction de la complexité (2) : splitting automatique



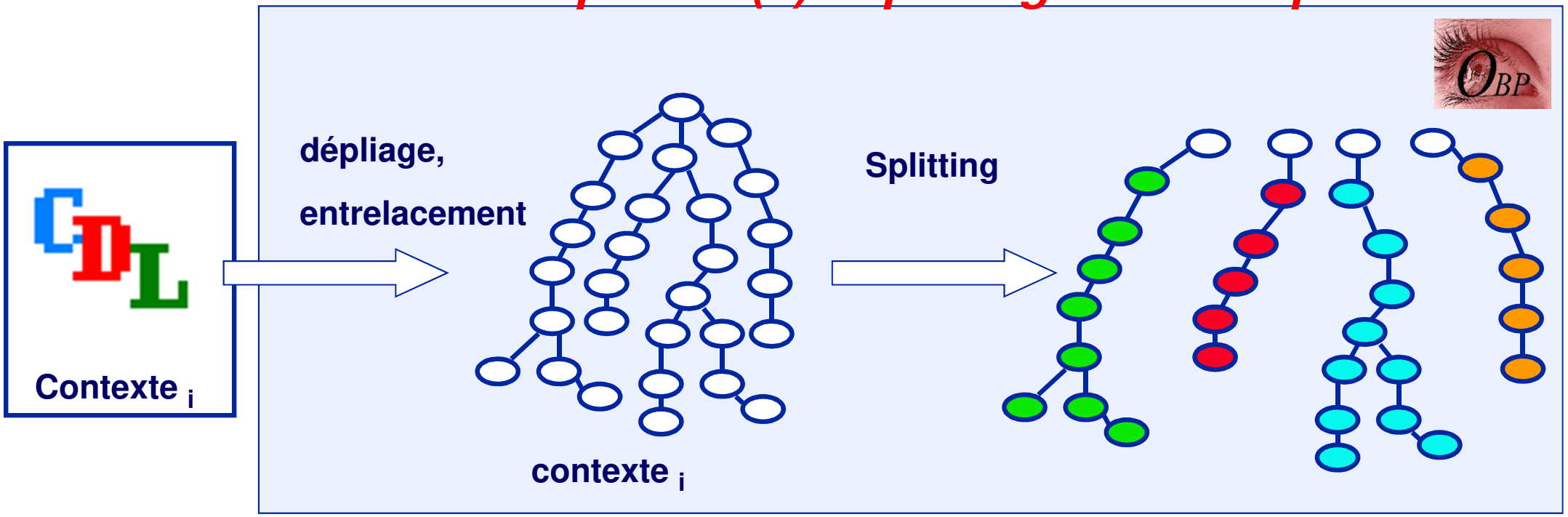
Contextes : Finis et acycliques

Réduction de la complexité (2) : splitting automatique



Contextes : Finis et acycliques

Réduction de la complexité (2) : splitting automatique



Ensemble de K_i sous-contextes

Modèle à valider (Fiacre)

K_i explorations et vérifications

Explorateur

AFADL 2011

HASE 2011,
Adv. In SE 2012
InTech 2012
TSI 2012

Exemple de résultats (sans CDL)

N (Number of devices)	Exploration & analyze time (sec)	N.of LTS configurations	N.of LTS transitions
1	1	43 828	321 002
2	4	350 256	2 475 392
3	19	1 466 934	6 430 265
4	H	–	–

Configuration mémoire 3 G.O.

Exemple de résultats (avec CDL)

N. of devices	Exploration time (sec)	N. of sub-contexts	N. of LTS config.	N. of LTS trans.
4	954	22	16 450 288	75 362 832
5	1 256	28	33 568 422	156 743 290
6	3 442	242	68 880 326	368 452 864
7	6 480	344	126 450 324	634 382 590
...	...	...	...	...
...	...	...	...	...

Splitting

Langage 

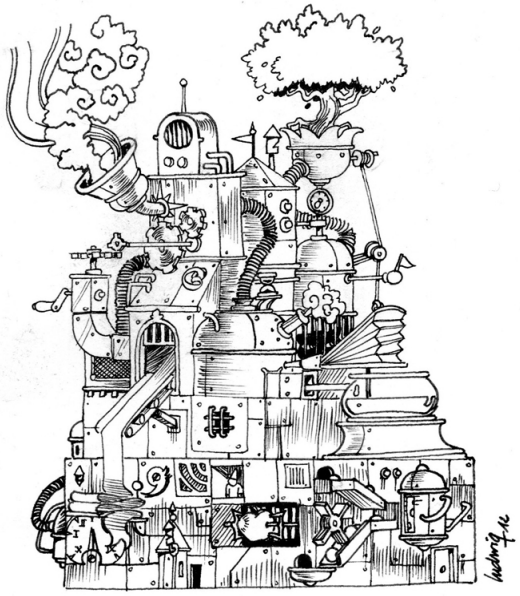
- ***Gestion des contextes***
- ***Gestion des propriétés***

Type et expression des propriétés

Disposer de primitives élémentaires d'observation, à grain fin

- **Invariants :** expression basée sur des prédicats

  Valeur d'une variable, Etat d'un processus, Etat d'une fifo



Type et expression des propriétés

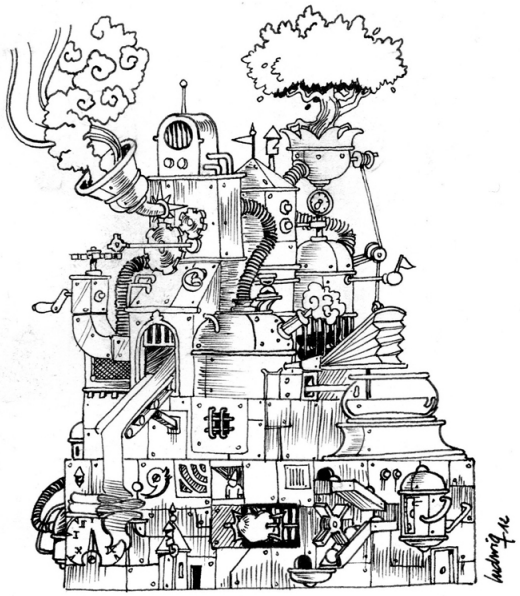
Disposer de primitives élémentaires d'observation, à grain fin

- **Invariants :** expression basée sur des prédicats

  Valeur d'une variable, Etat d'un processus, Etat d'une fifo

- **Observateur :** expression basée sur des patrons de définition de propriétés :

  Réponse, Précédence, Absence, Existence



Extension des patrons de Dwyer…

Type et expression des propriétés

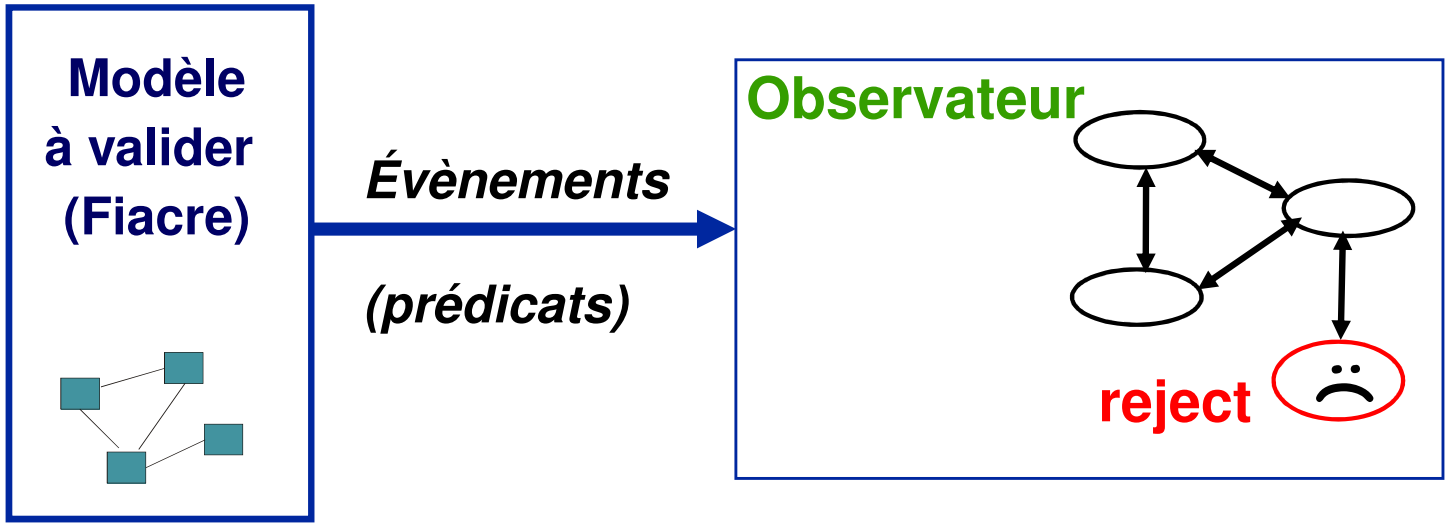
Disposer de primitives élémentaires d'observation, à grain fin

- **Invariants :** expression basée sur des prédicats

  Valeur d'une variable, Etat d'un processus, Etat d'une fifo

- **Observateur :** expression basée sur des patrons de définition de propriétés :

  Réponse, Précédence, Absence, Existence



Exemple de patron de définition de propriété (Réponse)

Property P;

ALL Ordered
exactly one **occurrence of** S_CP_hasReachState_Init
exactly one **occurrence of** login1
end
eventually **leads-to** [0..maxD_log]
AN
one or more **occurrence of** ackLog (id)
end

S_CP_hasReachState_Init may never **occurs**
login1 may never **occurs**
one of ackLog (id) cannot **occur before** login1
repeatability: true

Forme compréhensible par les ingénieurs

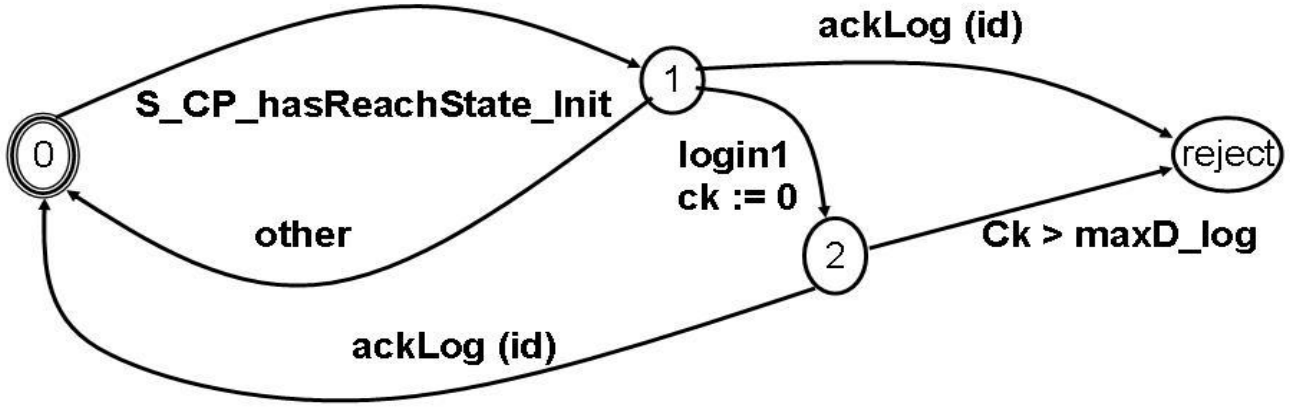
Exemple de patron de définition de propriété (Réponse)

Property P;

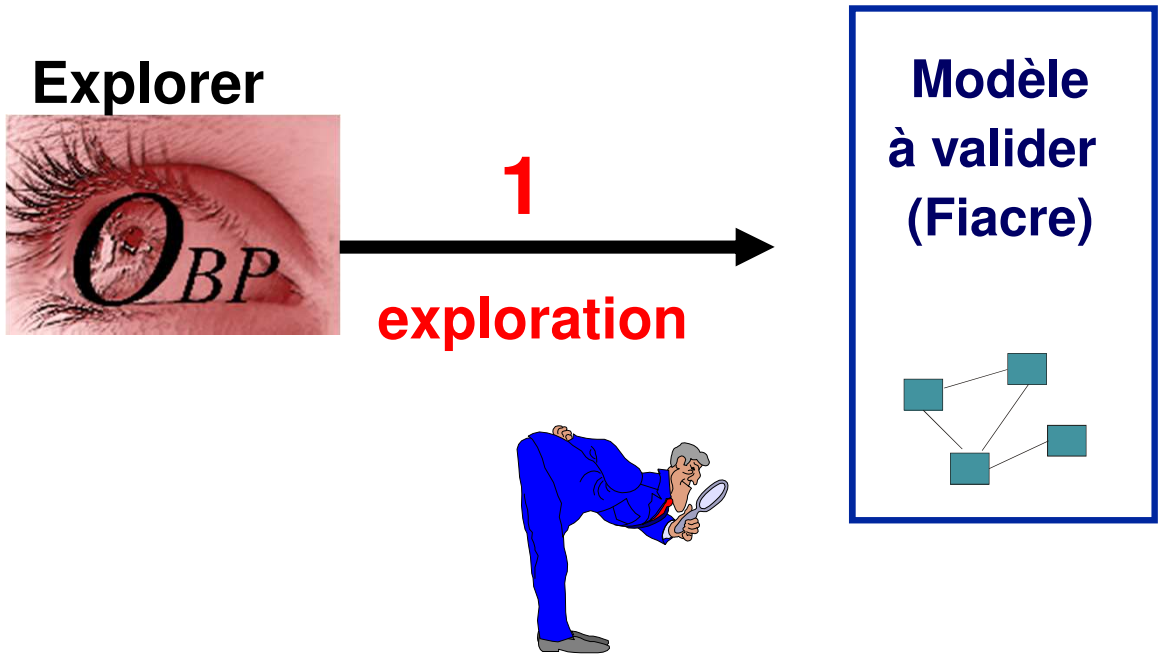
ALL Ordered
exactly one **occurrence of** S_CP_hasReachState_Init
exactly one **occurrence of** login1
end
eventually **leads-to** [0..maxD_log]
AN
one or more **occurrence of** ackLog (id)
end

S_CP_hasReachState_Init may never **occurs**
login1 may never **occurs**
one of ackLog (id) cannot **occur before** login1
repeatability: true

Transformation automatique



OBP : noyau d'observation



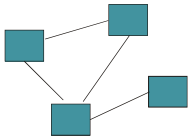
OBP : noyau d'observation

Explorer

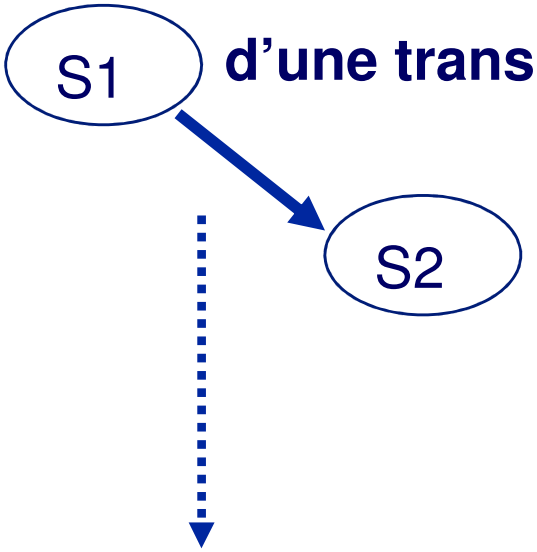
1

exploration

Modèle à valider (Fiacre)



Exécution d'une transition



Liste :

- **Valeurs de prédicats**
- **Évènements**

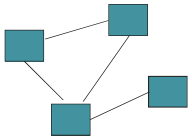
OBP : noyau d'observation

Explorer

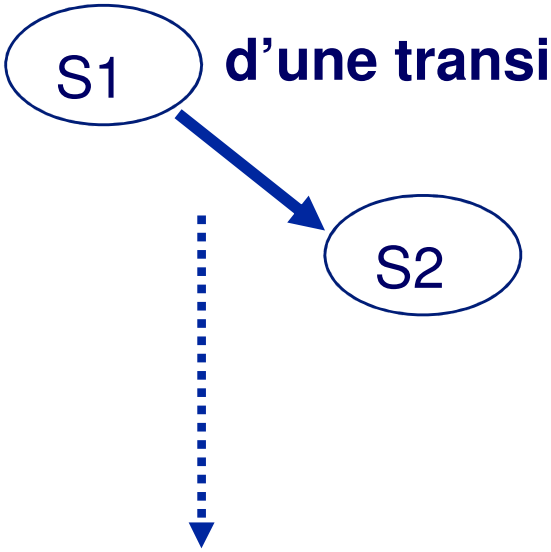
1

exploration

Modèle à valider (Fiacre)



Exécution d'une transition



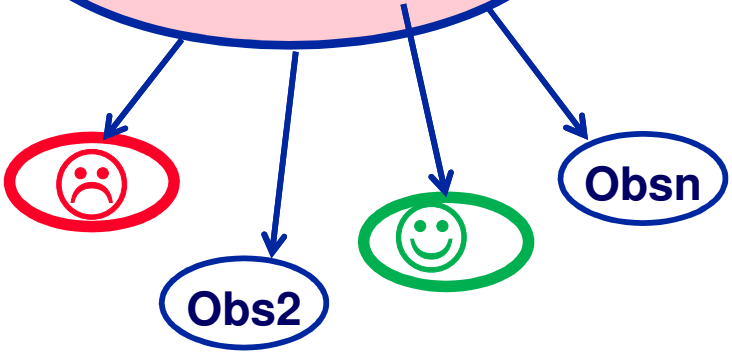
2

observation

Liste :

- Valeurs de prédicats
- Évènements

Machine d'observation



Ensemble d'observateurs

Quelques expérimentations

ATOS **Commandes de vol A380** (SysML)

Protocoles ATC - A340, A380 (SDL)

Composants logiciels embarqués(UML)

Spécifications fonctionnelles SLAM-F (NAF)

Logiciels de maintenance satellites (SysML)

Logiciels de bord satellites (SysML / AADL-Like)

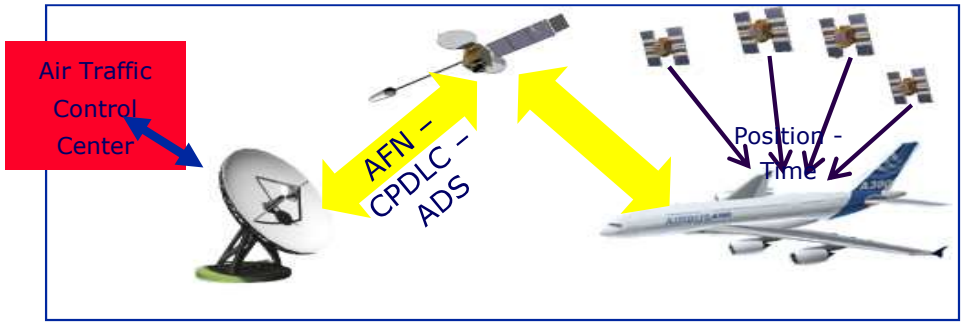
Spécifications fonctionnelles système-Logiciel (SysML)

Pacemaker [Hermes'13, Willey'13], Chalenge **Landing Gear System** [ABZ'14]

Quelques résultats

Nombres de propriétés formalisées avec les patrons de définition

Nombre de propriétés étudiées	Cas 1 (49)	Cas 2 (94)	Cas 3 (136)	Cas 4 (85)	Cas 5 (188)	Cas 6 (151)	Total (703)
Propriétés prouvables							
Non-prouvables							



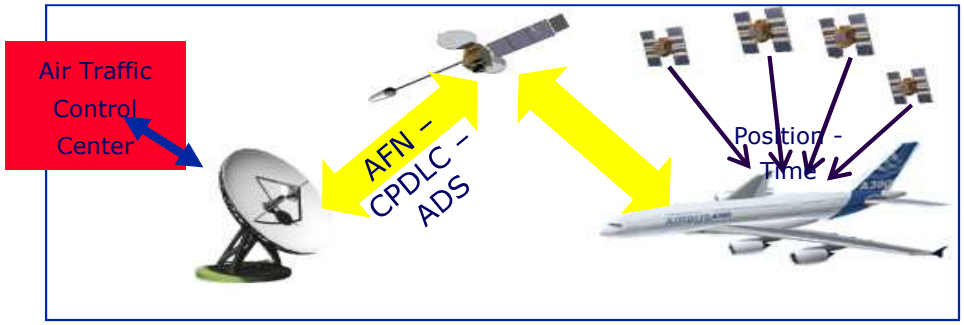
AIRBUS

CS COMMUNICATION & SYSTEMES

Quelques résultats

Nombres de propriétés formalisées avec les patrons de définition

Nombre de propriétés étudiées	Cas 1 (49)	Cas 2 (94)	Cas 3 (136)	Cas 4 (85)	Cas 5 (188)	Cas 6 (151)	Total (703)
Propriétés prouvables	38 (78%)						
Non-prouvables	11 (22%)						



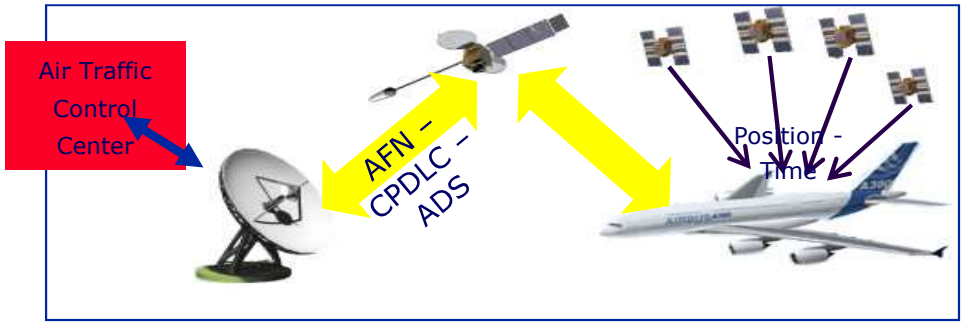
AIRBUS

CS COMMUNICATION & SYSTEMES

Quelques résultats

Nombres de propriétés formalisées avec les patrons de définition

Nombre de propriétés étudiées	Cas 1 (49)	Cas 2 (94)	Cas 3 (136)	Cas 4 (85)	Cas 5 (188)	Cas 6 (151)	Total (703)
Propriétés prouvables	38 (78%)	73 (78%)	72 (53%)	49 (58%)	155 (82%)	41 (27%)	428 (61%)
Non-prouvables	11 (22%)	21 (22%)	64 (47%)	36 (42%)	33 (18%)	110 (73%)	275 (39%)



AIRBUS

CS COMMUNICATION & SYSTEMES

Quelques résultas

Nombres de propriétés formalisées avec les patrons de définition

Nombre de propriétés étudiées	Cas 1 (49)	Cas 2 (94)	Cas 3 (136)	Cas 4 (85)	Cas 5 (188)	Cas 6 (151)	Total (703)
Propriétés prouvables	38 (78%)	73 (78%)	72 (53%)	49 (58%)	155 (82%)	41 (27%)	428 (61%)
Non-prouvables	11 (22%)	21 (22%)	64 (47%)	36 (42%)	33 (18%)	110 (73%)	275 (39%)

Rédaction des propriétés en amont des spécifications

Travaux de recherche

Observer-Based Prover

Context Description Language

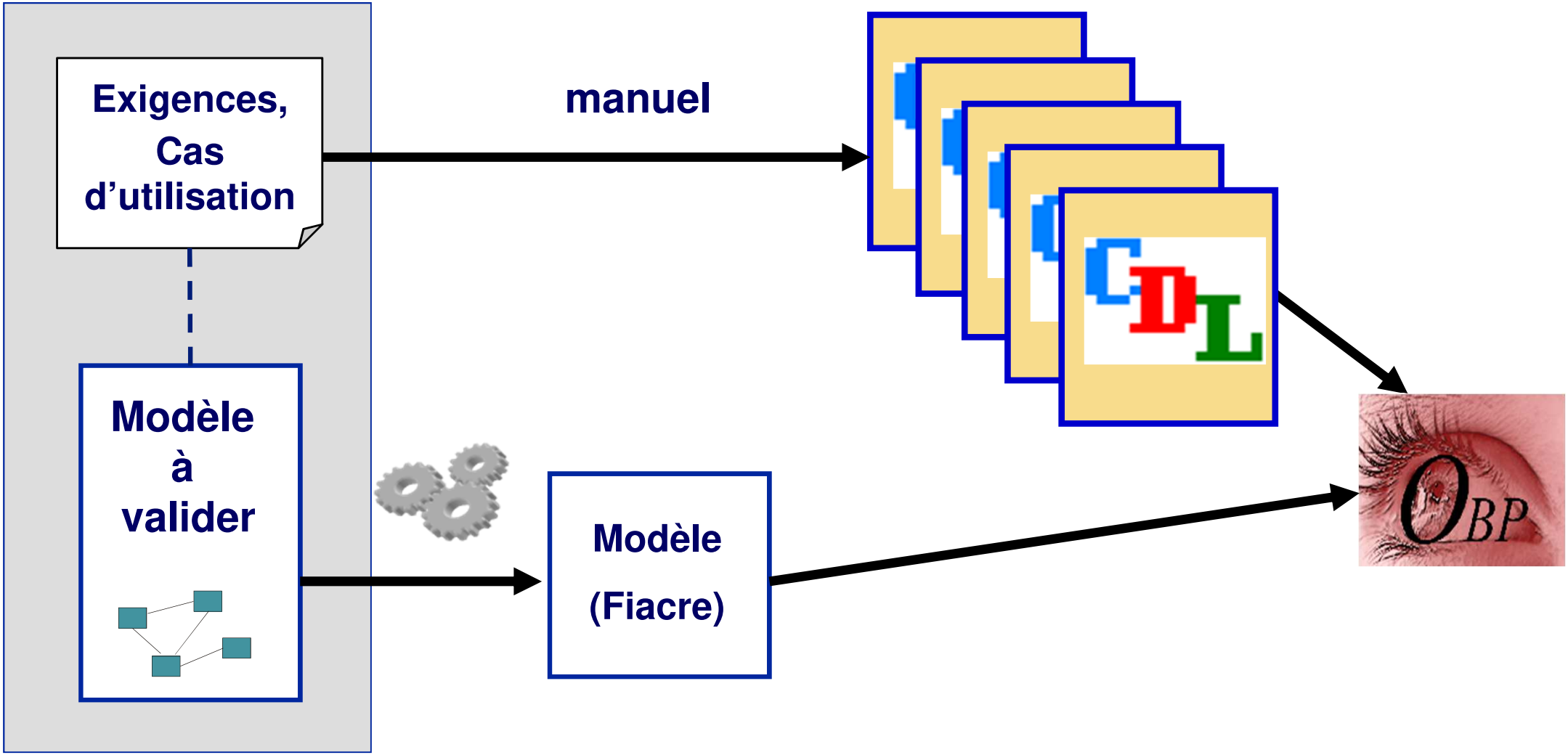
Expérimentations industrielles

UCM & Méthodologie

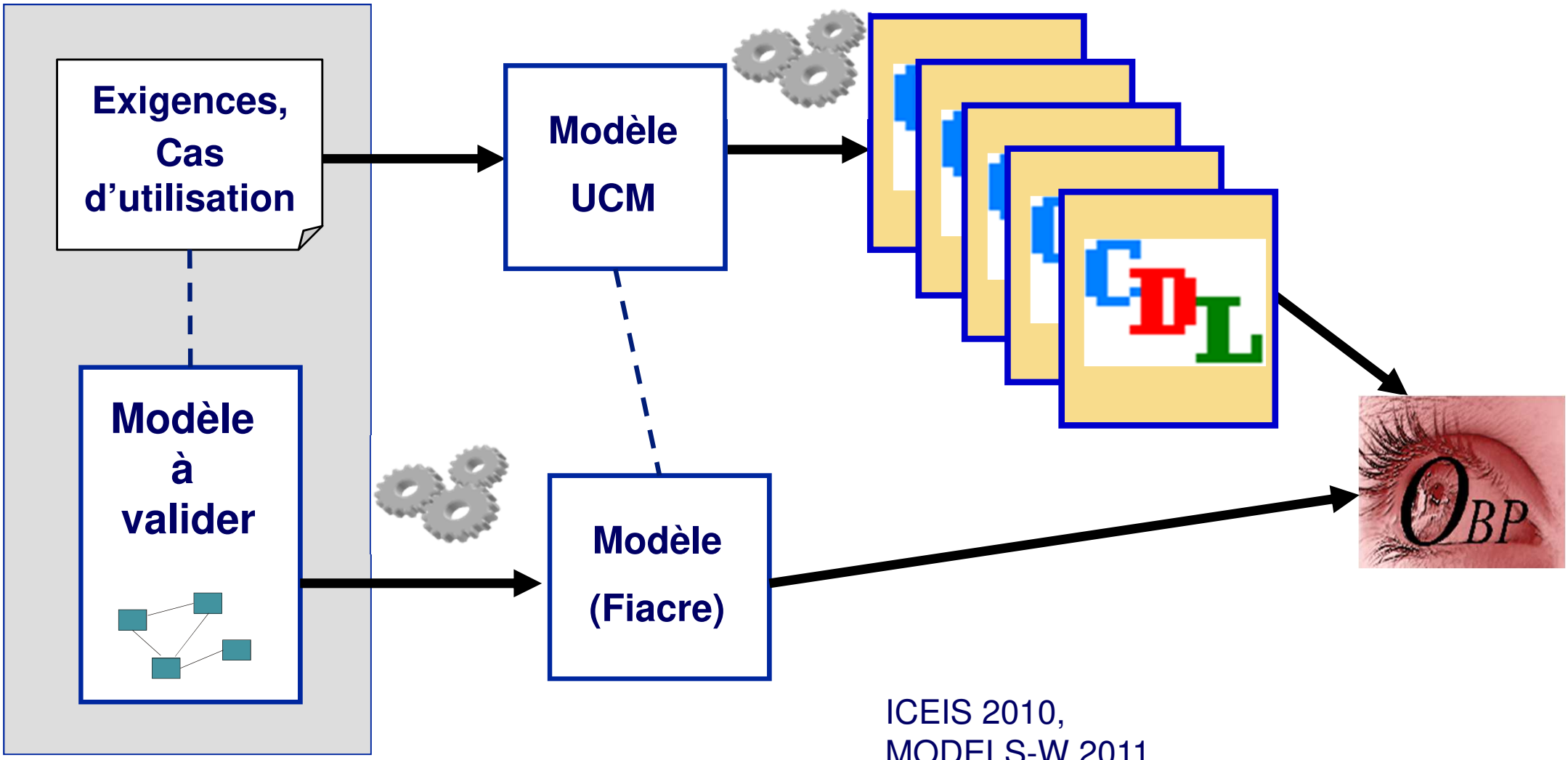
Optimisation, parallélisation OBP

Chaîne de transformation

Génération de codes CDL : User Context Model (UCM)



Génération de codes CDL: User Context Model (UCM)



ICEIS 2010,
MODELS-W 2011,
MODEVVA 2010, 2011,
APSEC 2011

User Context Model (UCM)

XUC

Use case 1

Déclarations
Body
Exigences

Use case 2

⋮

Use case n-1

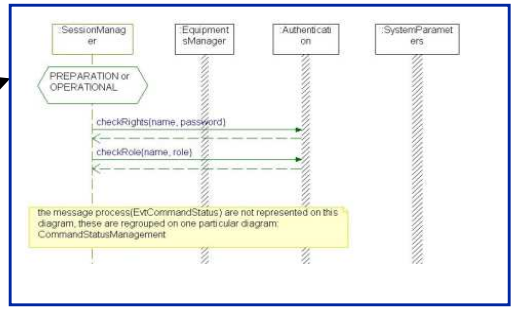
Use case n

référence

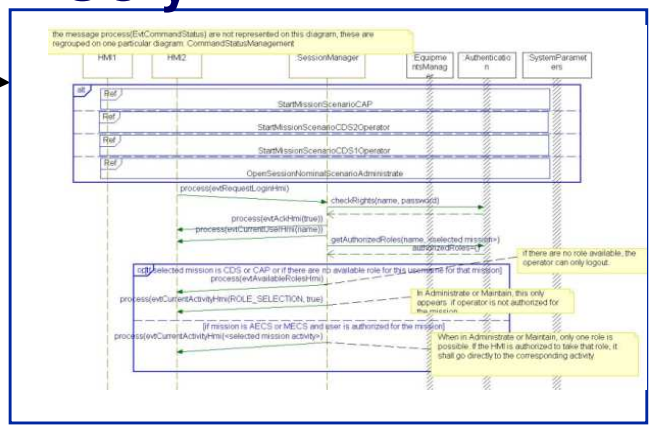
référence

référence

MSC x



MSC y



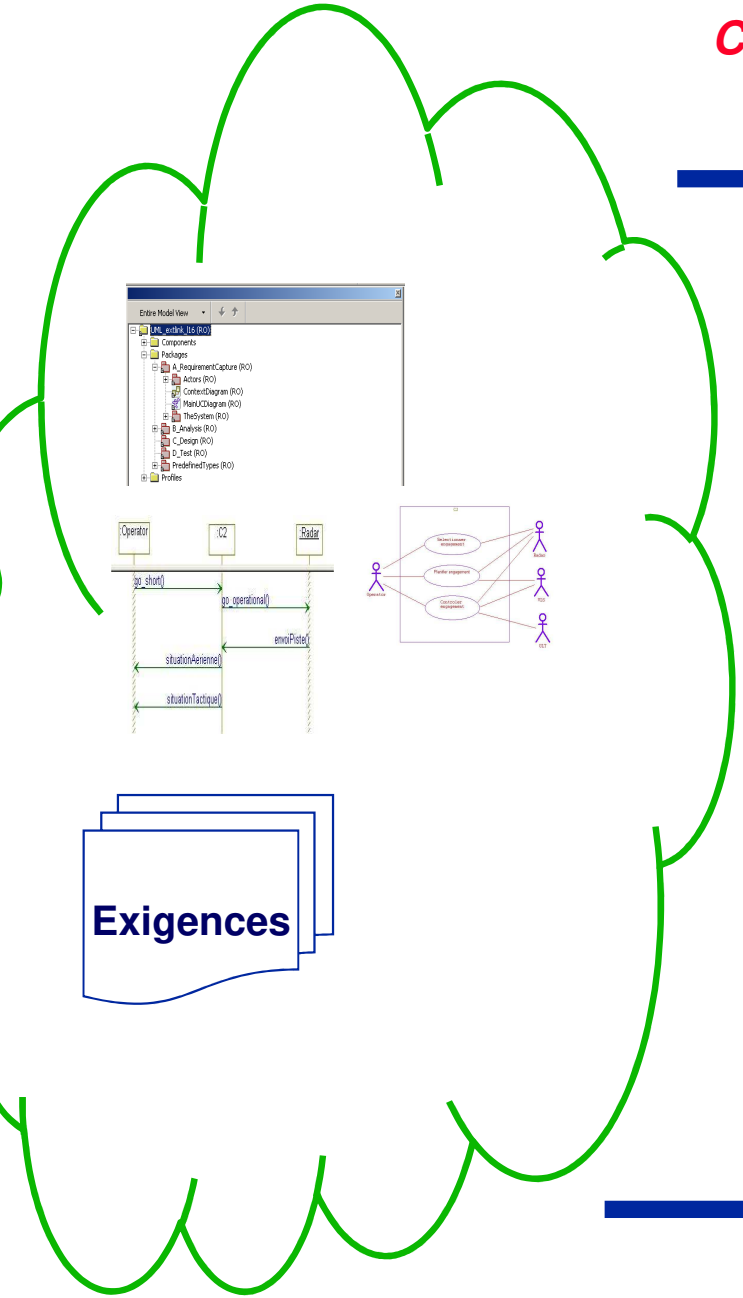
URM

Exigence 1

Exigence 2

Exigence n

Validation de modèles : Méthodologie



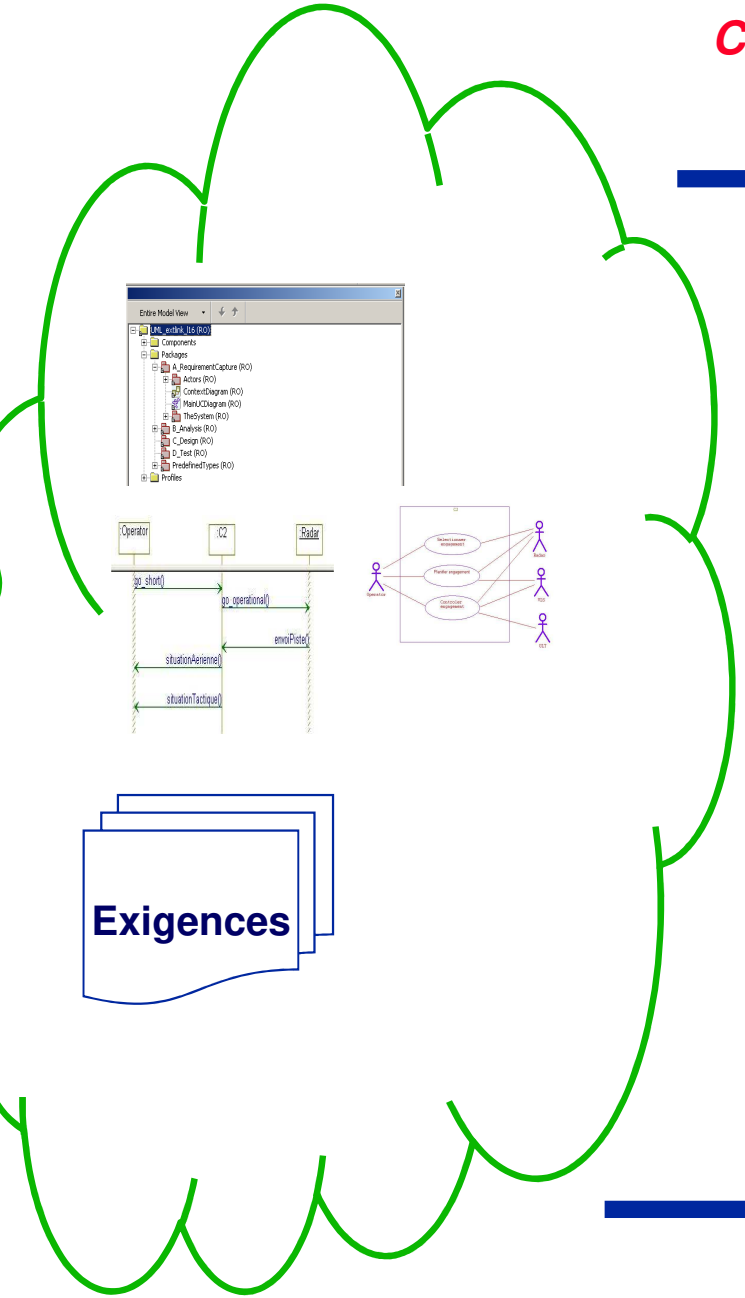
Construction d'un UCM

UCM

Construction du modèle

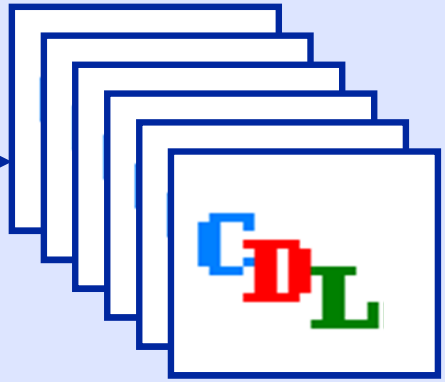
Modèle à valider

Validation de modèles : Méthodologie



Construction d'un UCM

UCM

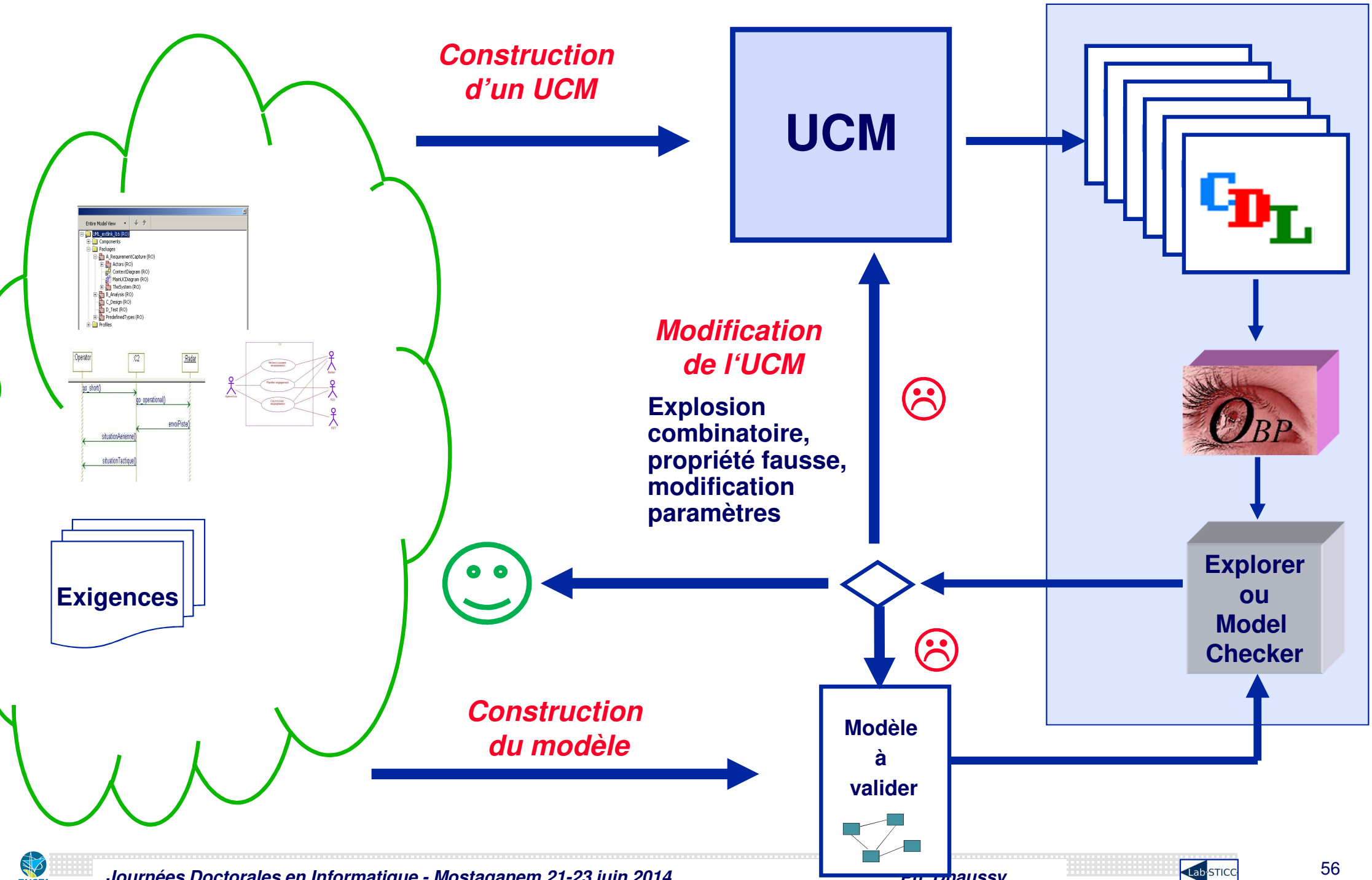


Explorer ou Model Checker

Construction du modèle

Modèle à valider

Validation de modèles : Méthodologie



Travaux de recherche

Observer-Based Prover

Context Description Language

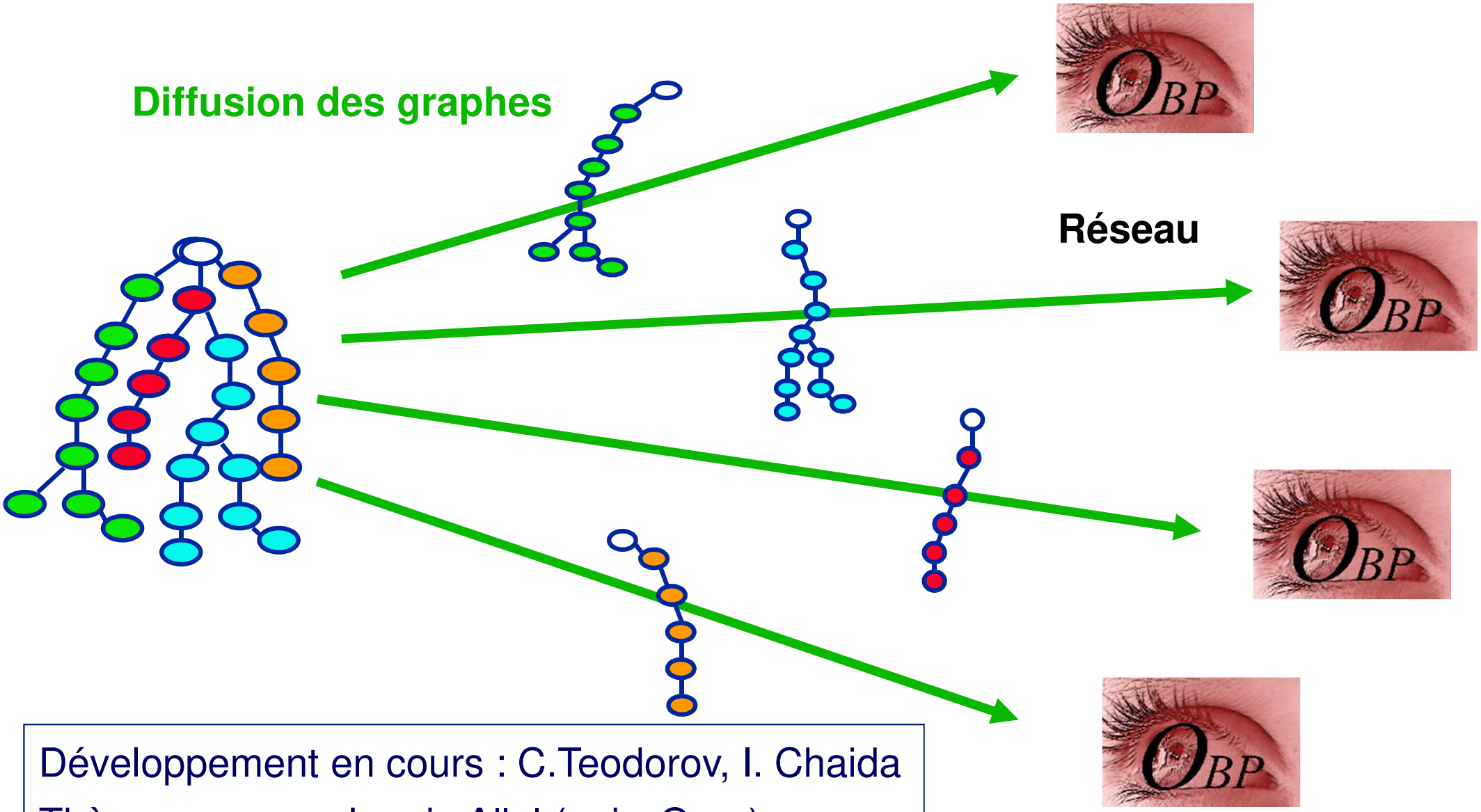
Expérimentations industrielles

UCM & Méthodologie

Optimisation, parallélisation OBP

Chaîne de transformation

Explorations distribuées sur un réseau de machine

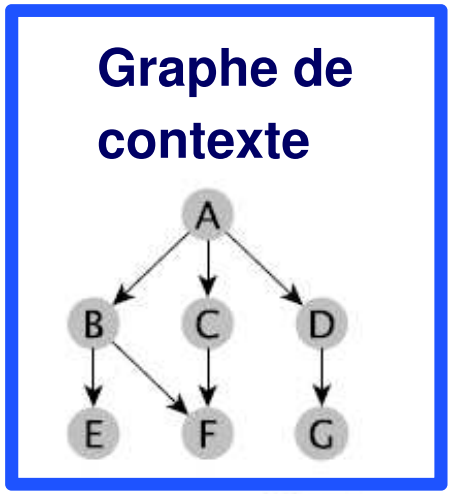


Développement en cours : C.Teodorov, I. Chaida

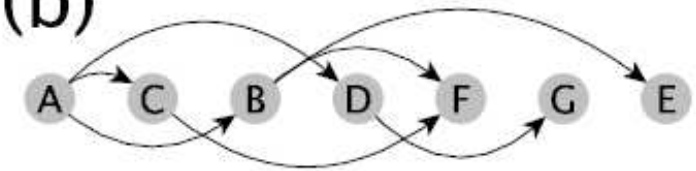
Thèse en cours : Lamia Allal (univ. Oran)

Optimisation des explorations
[PastFree Reachability Algorithm]

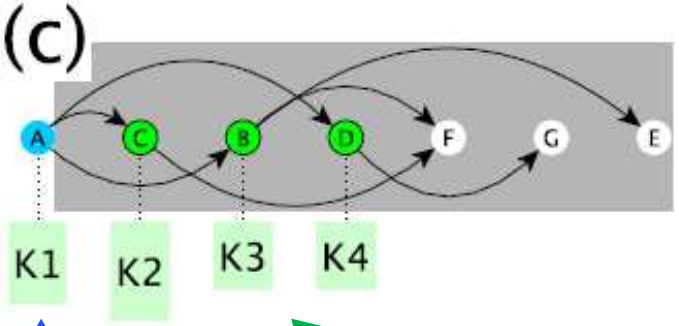
(a)



(b)



(c)

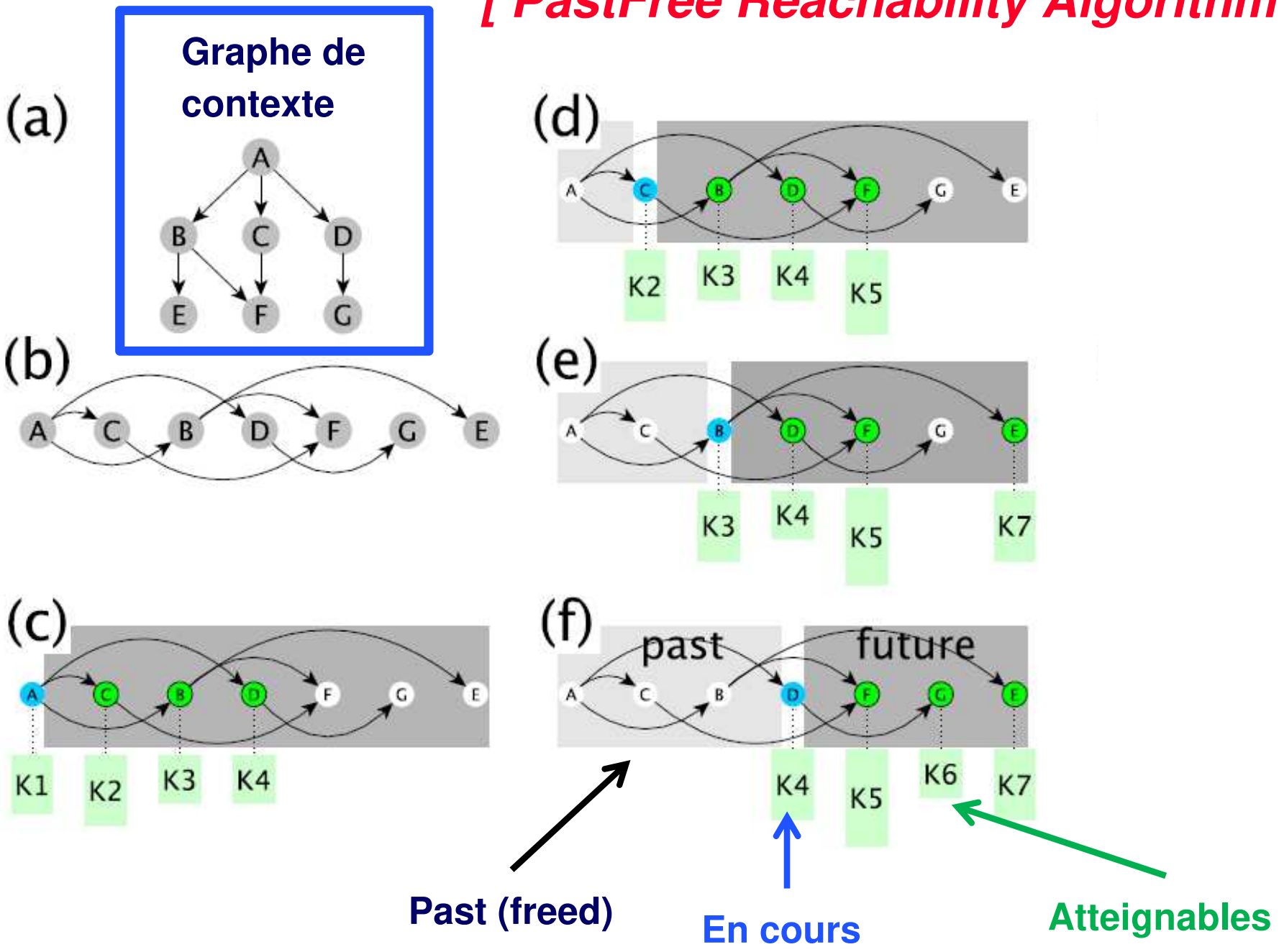


En cours

Atteignables

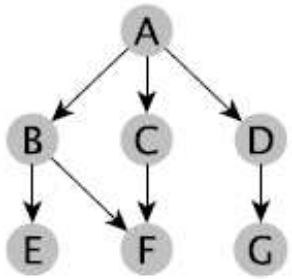
[soumission ASE 2014]
Avec C.Teodorov, L.Leroux

Optimisation des explorations
[PastFree Reachability Algorithm]

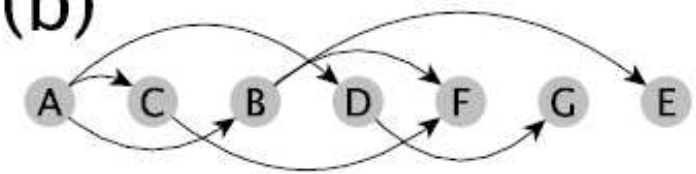


Optimisation des explorations
[PastFree Reachability Algorithm]

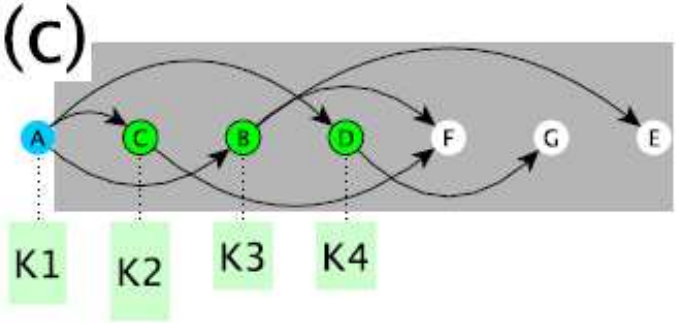
Graphe de contexte



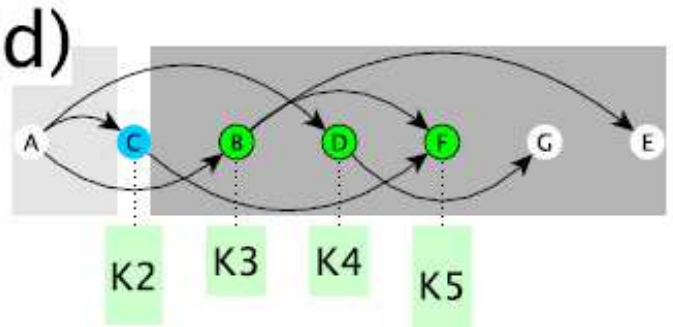
(a)



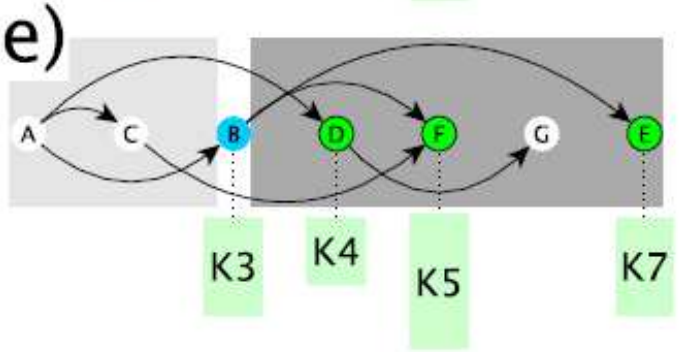
(b)



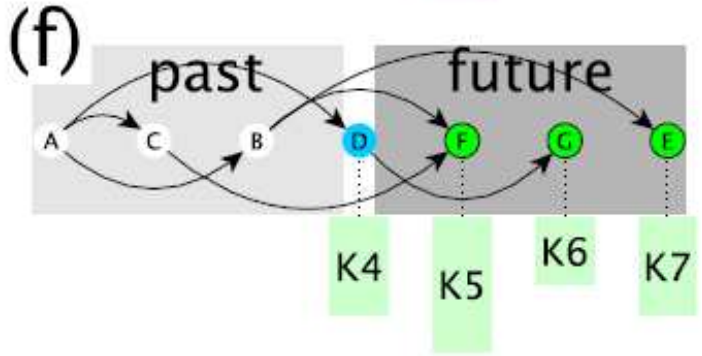
(c)



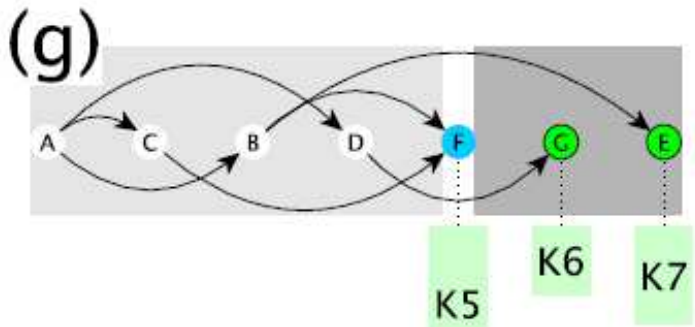
(d)



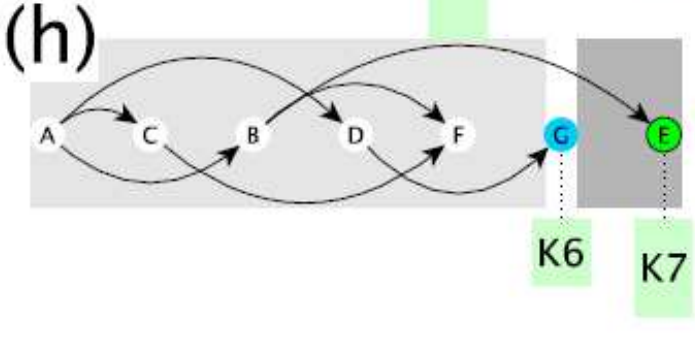
(e)



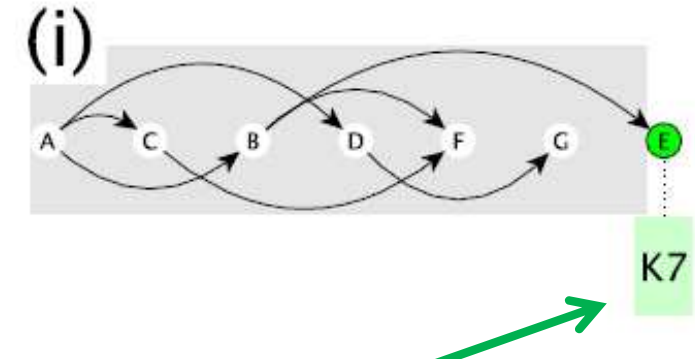
(f)



(g)



(h)

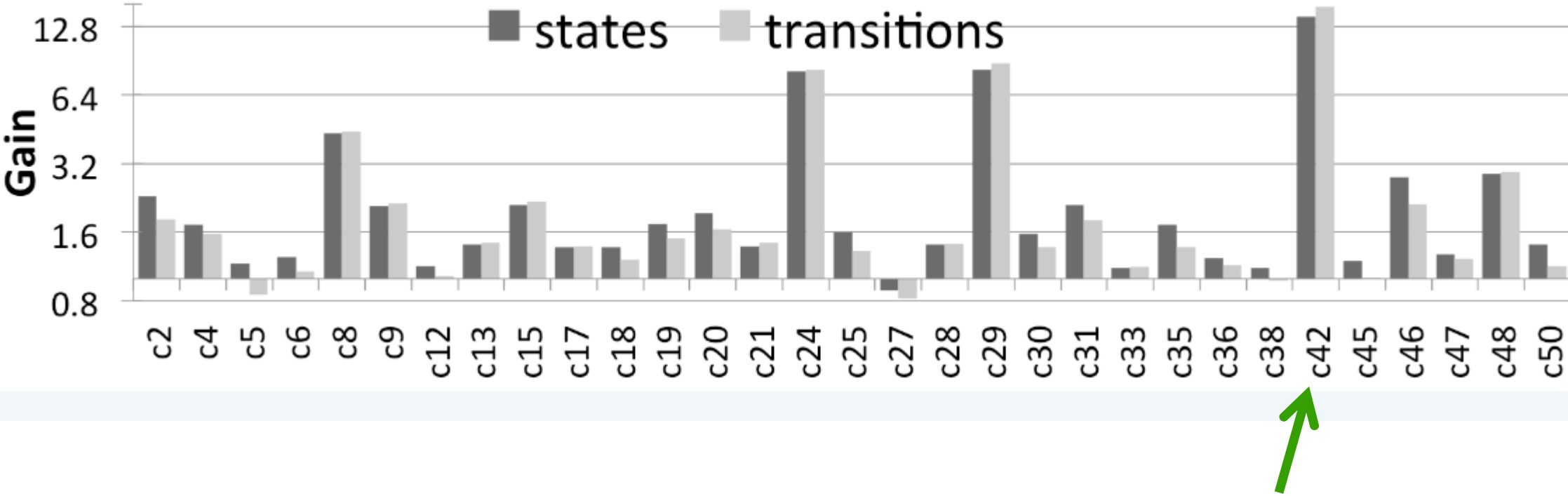


(i)

Atteignables

Optimisation des explorations
[PastFree Reachability Algorithm]

Gain obtenu avec l'algorithme



14 fois plus de configurations parcourues

Travaux de recherche

Observer-Based Prover

Context Description Language

Expérimentations industrielles

UCM & Méthodologie

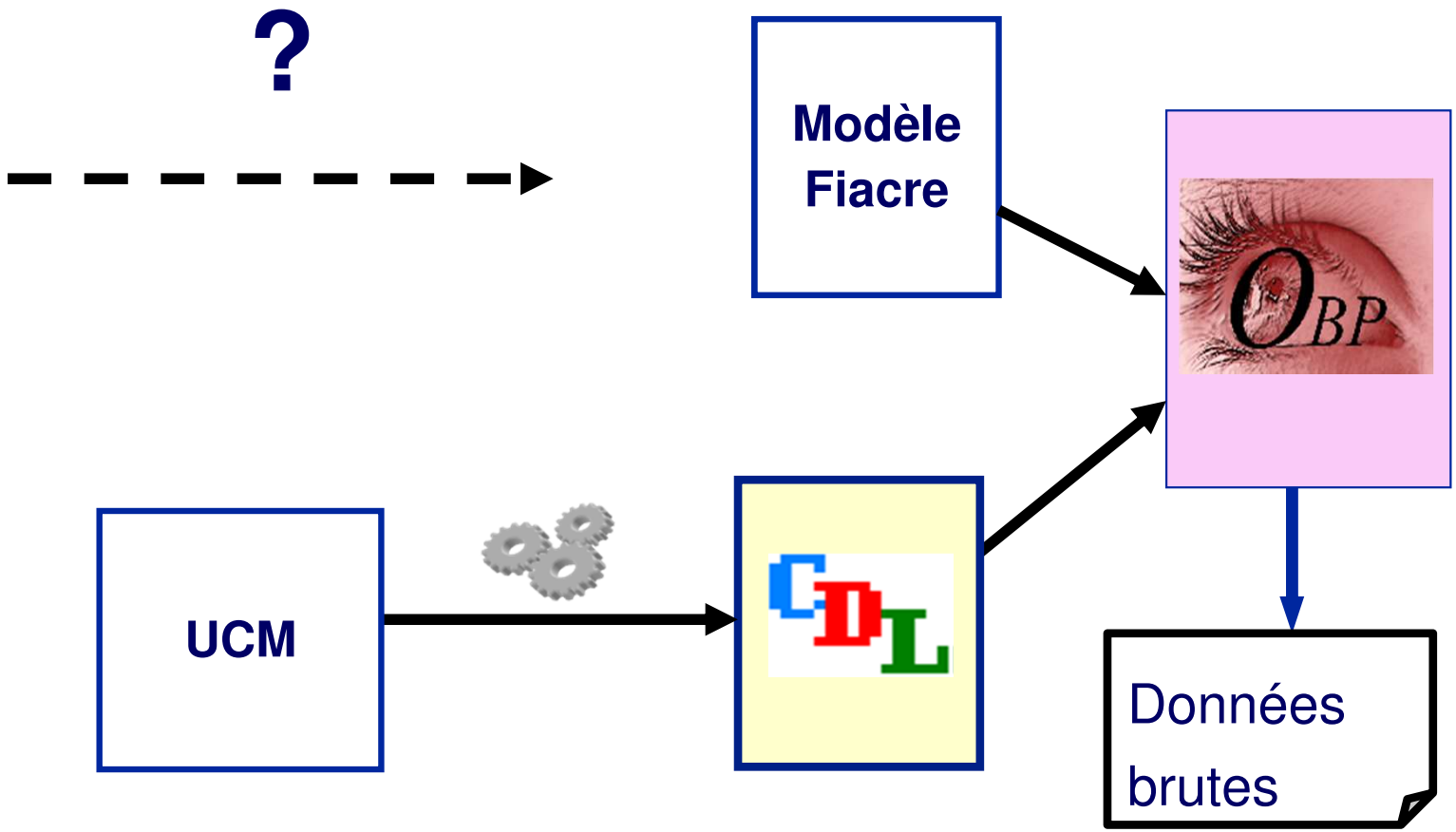
Optimisation, parallélisation OBP

Chaîne de transformation

Chaîne de transformation

Editeurs

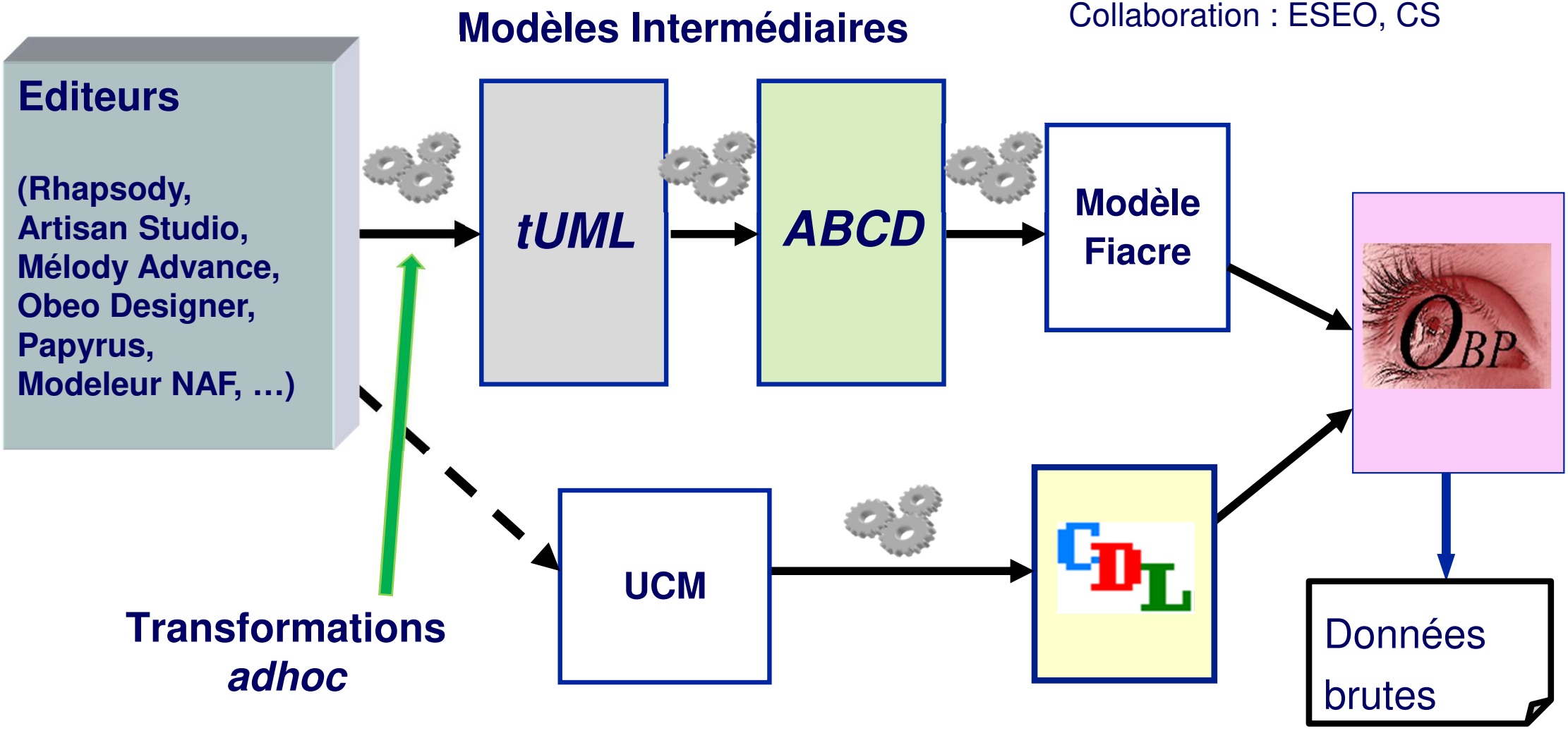
(Rhapsody, Artisan Studio, Mélody Advance, Obeo Designer, Papyrus, Modeleur NAF, …)



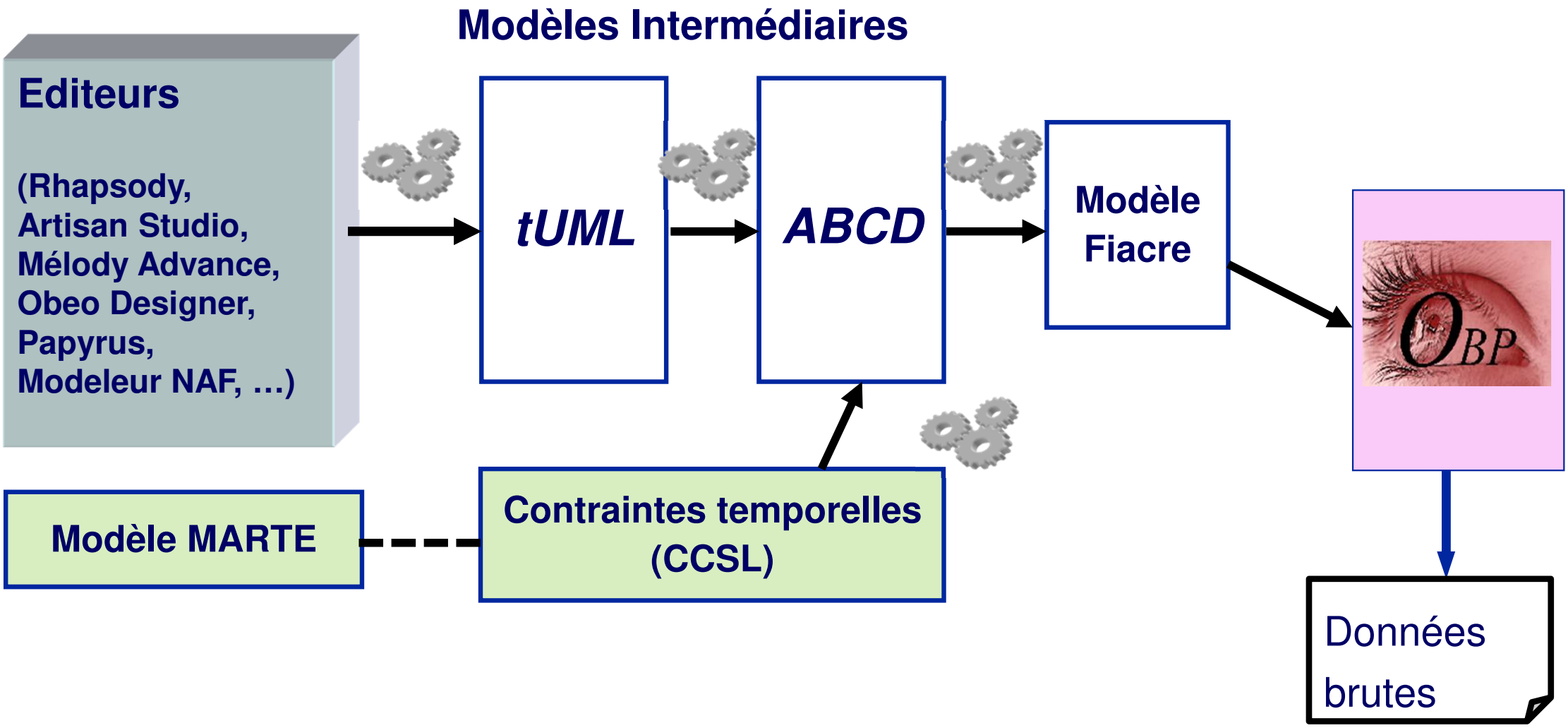
Chaîne de transformation

Projet DEPARTS

[Neptune 2014]
Collaboration : ESEO, CS



Chaîne de transformation



[SEFM'13]
(projet ANR GEMOC)

Retour d'expérience : l'apport

- CDL, UCM : **permet une formalisation**

  Cas d'utilisation (contextes) et des propriétés

- Contribution à la **gestion de la complexité**
- Motivation des partenaires pour une **approche plus formelle** de leurs exigences
- Aide à la **structuration** de leurs spécifications (UCM)
- **Mise en pratique** en contexe industriel

→ **Meilleure appropriation des processus de validation formelle**

Retour d'expérience : difficultés et limites

- Spécifications industrielles :
  - Problèmes de **complétude et de cohérence**
  - Non pensées pour la validation formelle

- Effort important nécessaire pour la **compréhension**
  ~ plusieurs semaines pour une application

- Dialogue incontournable avec les experts métier
  Les bons outils, les bonnes méthodes, la conviction ne suffisent pas !

- Difficulté d'ordre **méthodologique** (peu abordé en enseignement)

Efforts à poursuivre

Minimiser les ruptures dans les pratiques industrielles

Méthodologies *ad-hoc* à imaginer, à concevoir, à développer

→ Domaines et types d'application spécifiques

Diffusion outils - méthode

www.obpcdl.org

Groupes d'échange avec la communauté industrielle

Laboratoire Communs

CALIPSO (Thales S.A)
VAS (soumission ANR LabCom)

Plateforme de distribution d'outils Open Source : PolarSys

- Partage des moyens entre industriels (*end user*)
- Services et écosystèmes autour des composants open source
- Gestion de la qualité et de la maturité des outils et composants
- Préparation de la certification

Plan

- Motivations
- Travaux, résultats, retour d'expérience
- Perspectives

Perspectives

Poursuite des expérimentations industrielles

Aide au diagnostic

UCM & Méthodologie

Optimisation OBP

Correction des transformations

شكرا على حسن متابعتكم

www.obpcdl.org