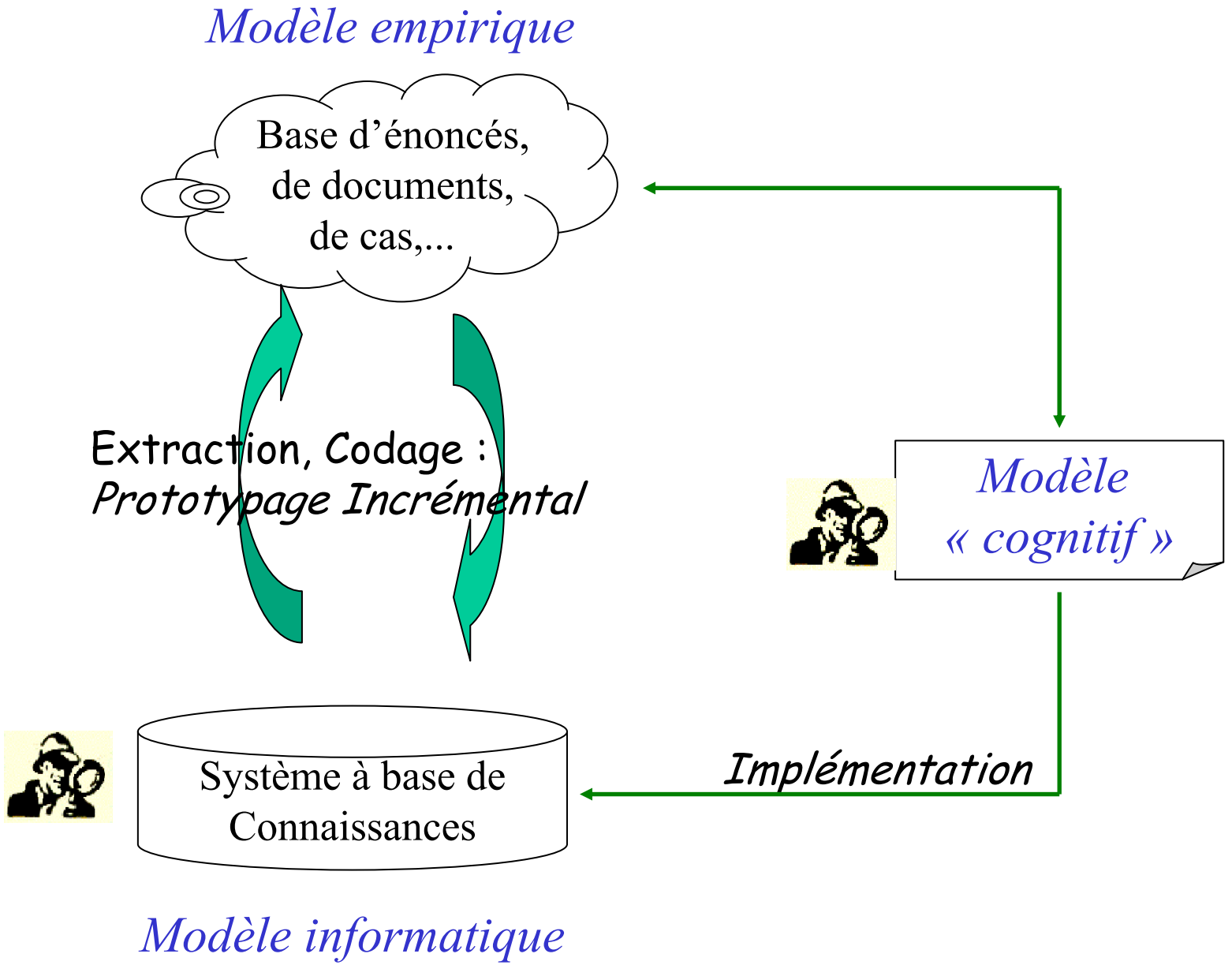
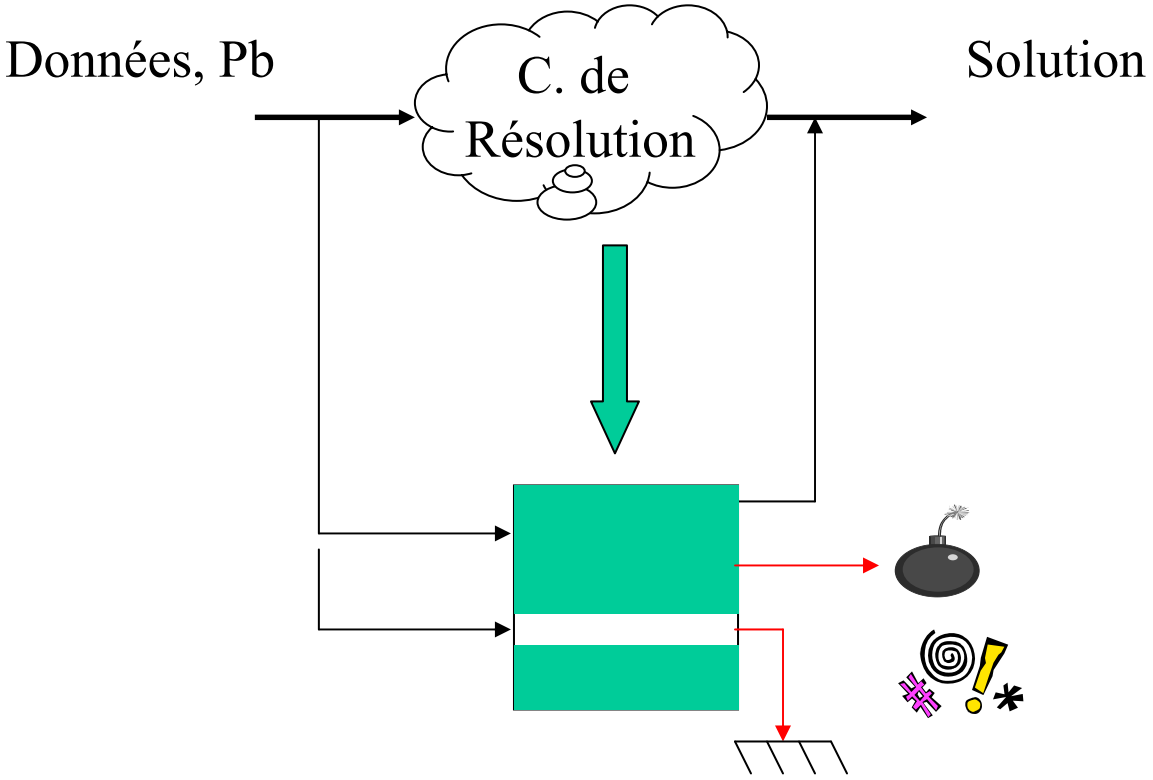

Modèle empirique
Base d'énoncés,
de documents,
de cas,...
Extraction, Codage :
Prototypage Incrémental
Modèle
« cognitif »
Implémentation
Système à base de
Connaissances
Modèle informatique

Approche directe

Dès que le développeur a acquis un élément de connaissance il l'implémente.

Avantages :
Montrer rapidement quelque chose,
En phase d'étude de faisabilité, permet de cerner des points tels que la communicabilité de la connaissance, les besoins des utilisateurs, …

Inconvénients :
Manque de vision d'ensemble, d'abstraction, de fiabilité
(tests a posteriori)



Prototypage Incrémental

Approche indirecte

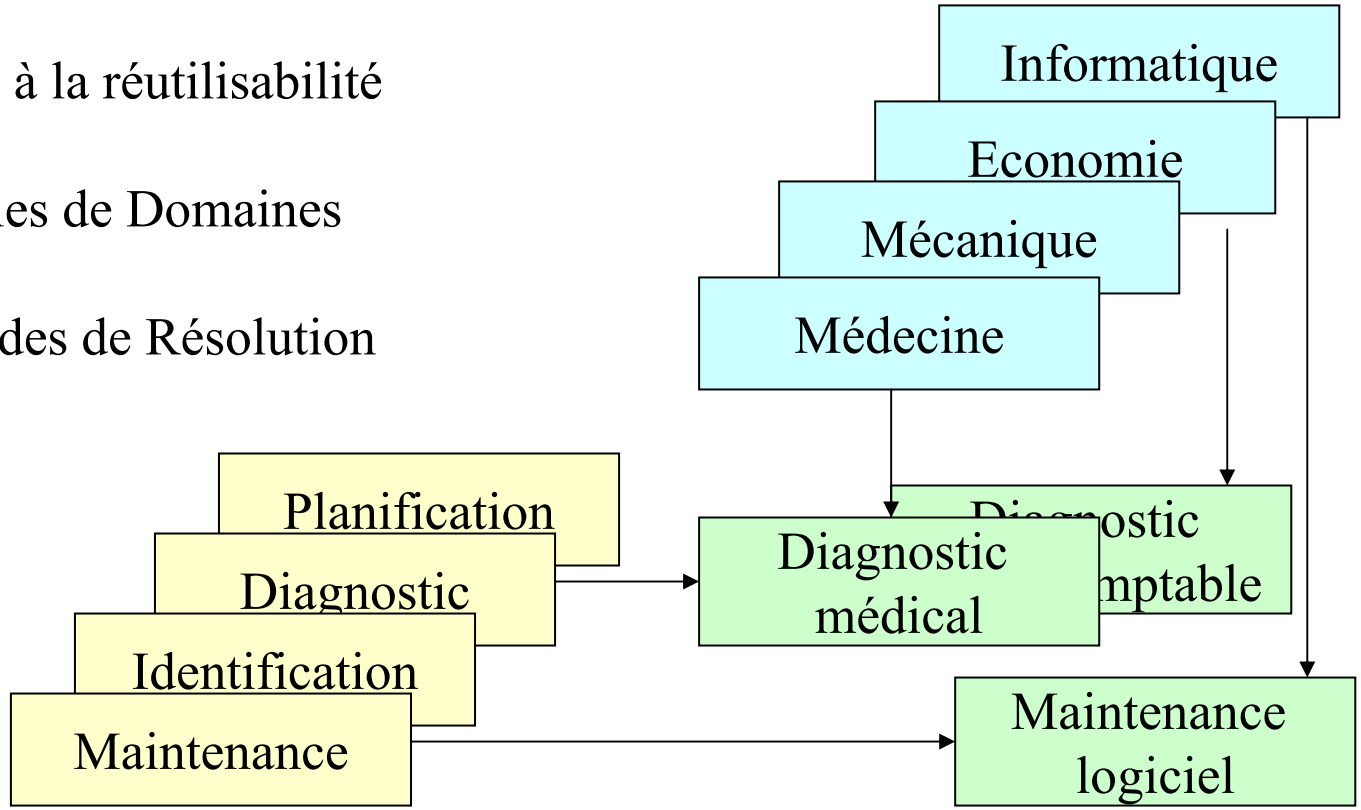
Elève le niveau d'abstraction;

Permet un contrôle a priori;

Ouvre la voie à la réutilisabilité

- de modèles de Domaines

- de méthodes de Résolution



1) Approche 'ontologique' (analytique) par sélection

Q : Peut-on bâtir un modèle d'un domaine susceptible de couvrir tous les usages
i.e. à la fois neutre vis à vis de toute tâche et réutilisable ?
Où s'arrête un Domaine ?

« is a »

C. de Domaine : Ontologie de Domaine = Une vue consensuelle sur un Domaine indépendante de la Tâche

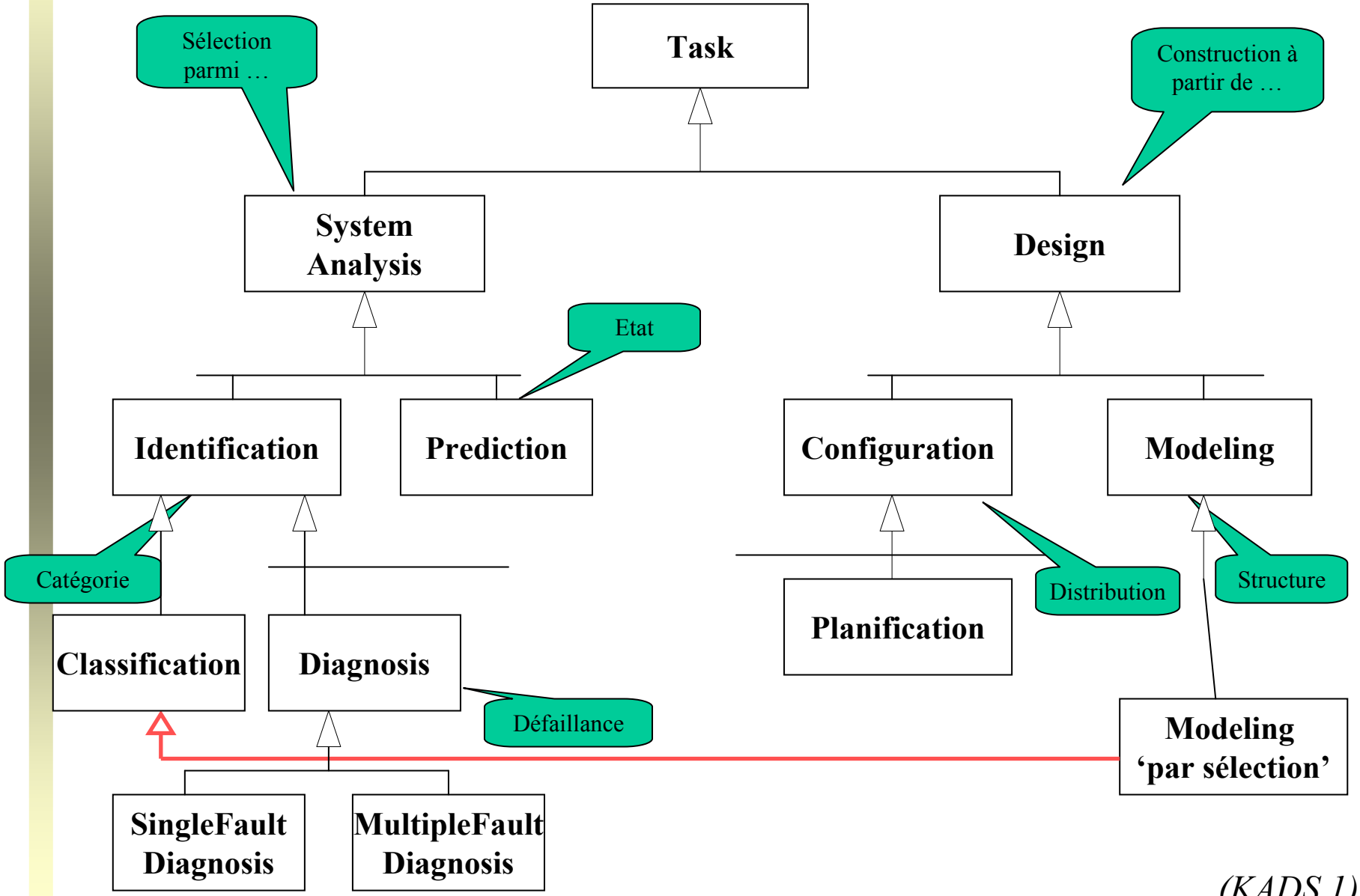
Q : Fusion ?

C. de Résolution : Schéma de Résolution + Objets de Raisonnement = Tâche Générique indépendante du Domaine

« is a »

Q : Dispose t-on de méthodes génériques pour tout type de tâche ?
Granularité optimale des méthodes génériques ?
(généricité versus réutilisabilité)
Sait-on discriminer (pour sélectionner) les tâches génériques ?

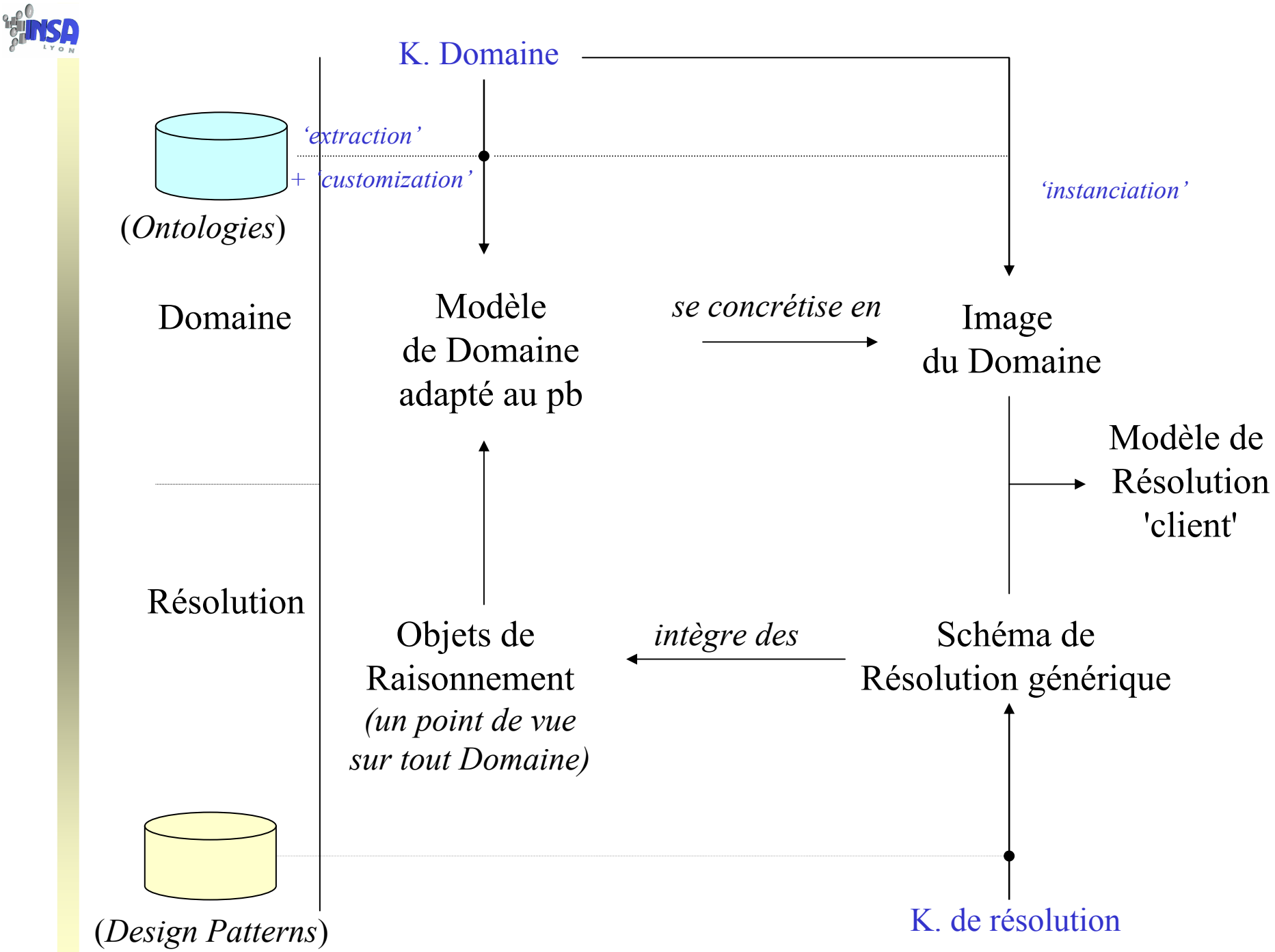
1) Approche ‘ontologique’ (analytique) par sélection



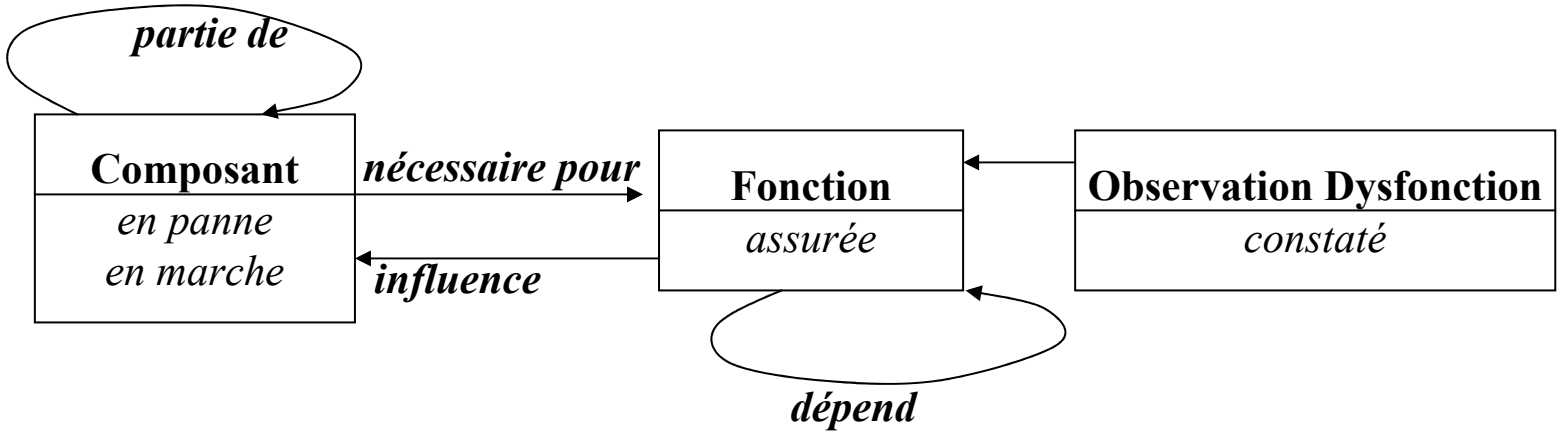
(KADS 1)

2) Approche constructiviste

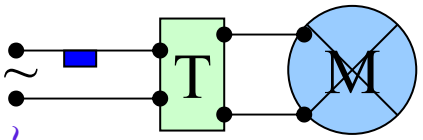
Construction et/ou
Adaptation, Raffinement, …
& Combinaison de « *Design Pattern* »


K. Domaine
'extraction'
+ 'customization'
'instanciation'
(Ontologies)
Domaine
Modèle
de Domaine
adapté au pb
se concrétise en
Image
du Domaine
Modèle de
Résolution
'client'
Résolution
Objets de
Raisonnement
(un point de vue
sur tout Domaine)
intègre des
Schéma de
Résolution générique
(Design Patterns)
K. de résolution

Exemple de Pattern : (structure, schémas causaux, méthode(s))



nécessaire pour(C, F) ∧ C. en marche = faux => F. assurée=faux
dépend (F1, F2) ∧ F2 . assurée = faux => F1. assurée = faux
F. assurée = faux ∧ influence (F, C) => C . en marche = faux
C . en panne = vrai => C . en marche = faux

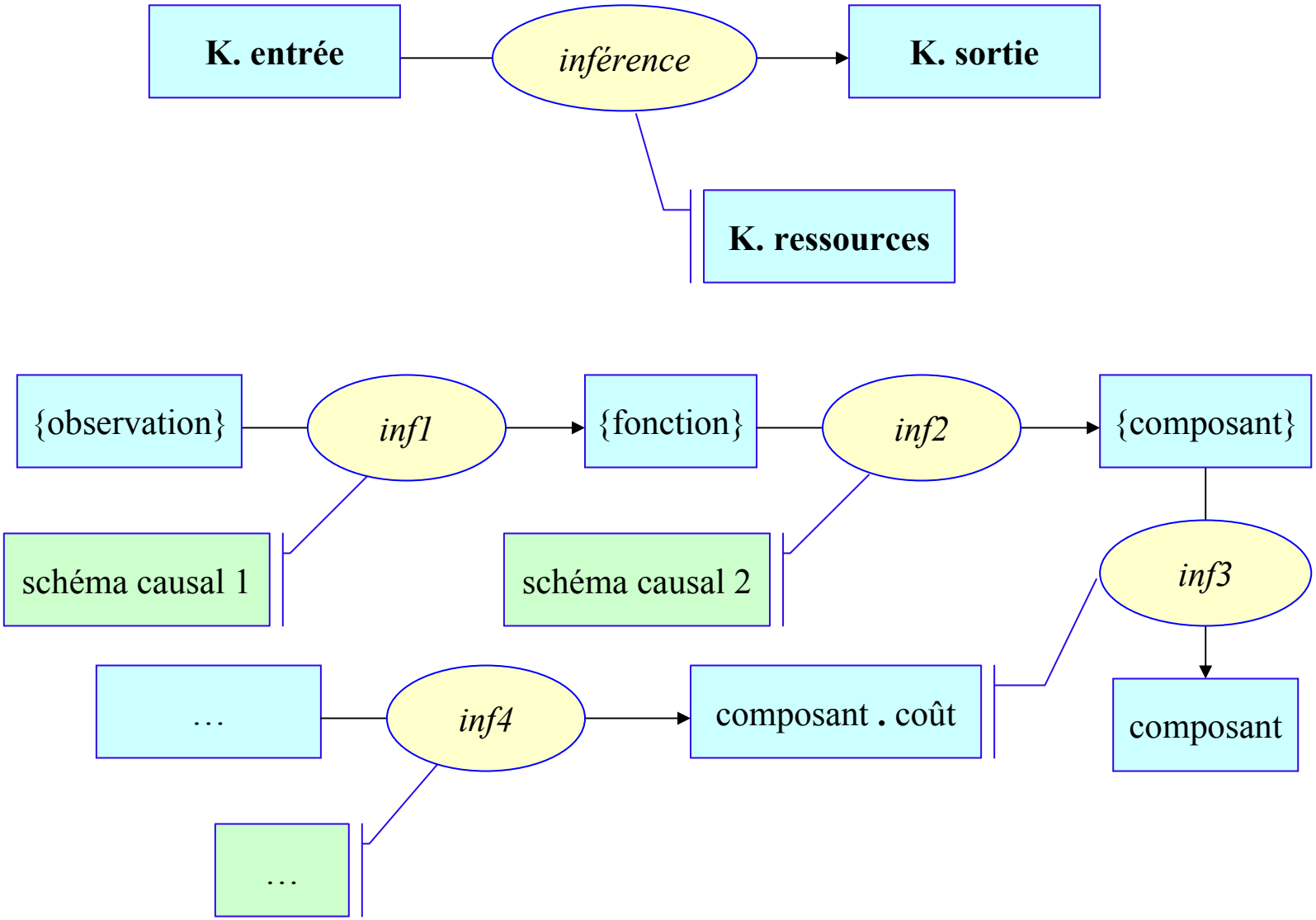


Méthodes :

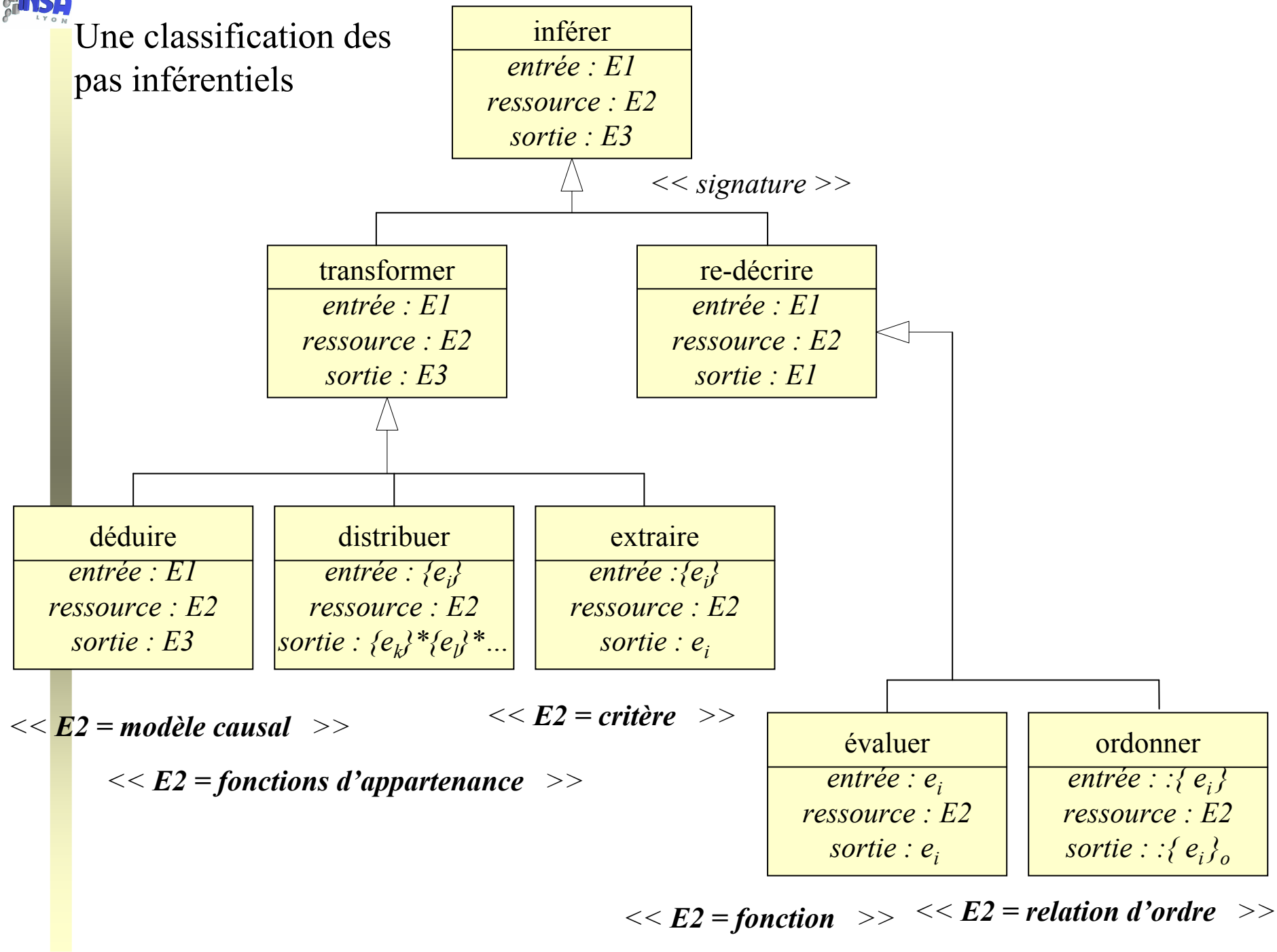
m1 : { A partir de dysfonctions,déterminer les composants défectueux }

m2 : { Evaluer l'influence d'un défaut sur composant }

Description de(s) la méthode(s) (flux des connaissances, enchaînement de pas inférentiels) associée(s) au Design Pattern

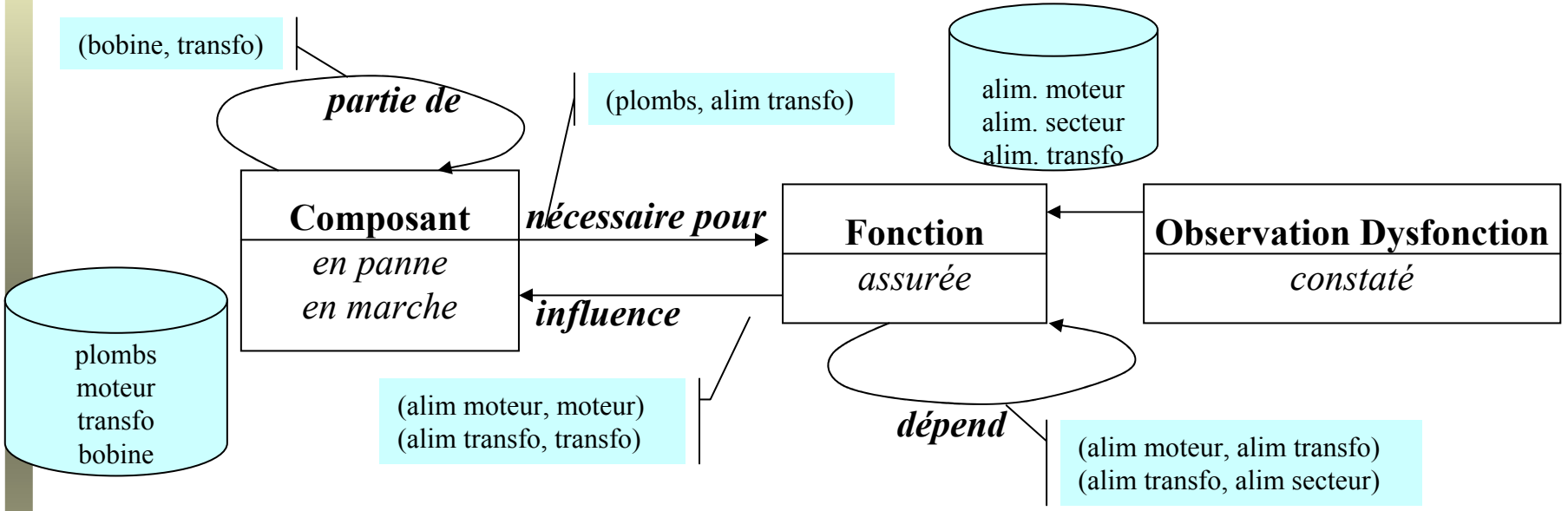


Une classification des pas inférentiels

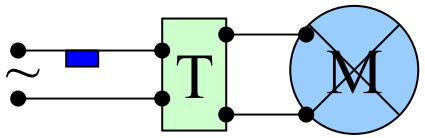


1) Sélection de Pattern, (structure, schémas causaux, méthode(s))

2) Instanciation



nécessaire pour(C, F) ∧ C. en marche = faux => F. assurée=faux
dépend (F1, F2) ∧ F2 . assurée = faux => F1. assurée = faux
F. assurée = faux ∧ influence (F, C) => C . en marche = faux
C . en panne = vrai => C . en marche = faux

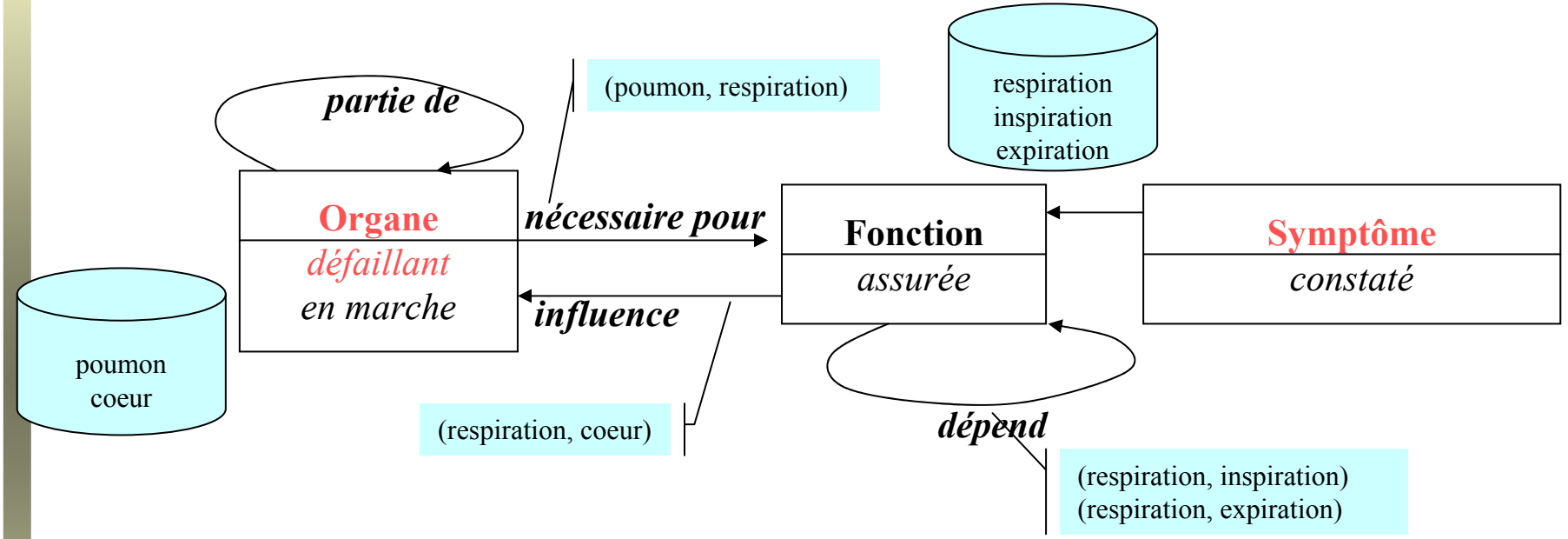


Méthodes :

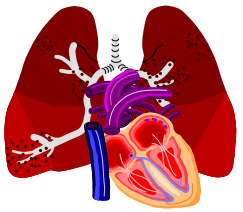
m1 : un Flux Inférentiel

1) Customisation de Pattern

2) Instanciation



nécessaire pour(C, F) ∧ C. en marche = faux => F. assurée=faux
dépend (F1, F2) ∧ F2 . assurée = faux => F1. assurée = faux
F. assurée = faux ∧ influence (F, C) => C . en marche = faux
C . défaillant = vrai => C . en marche = faux

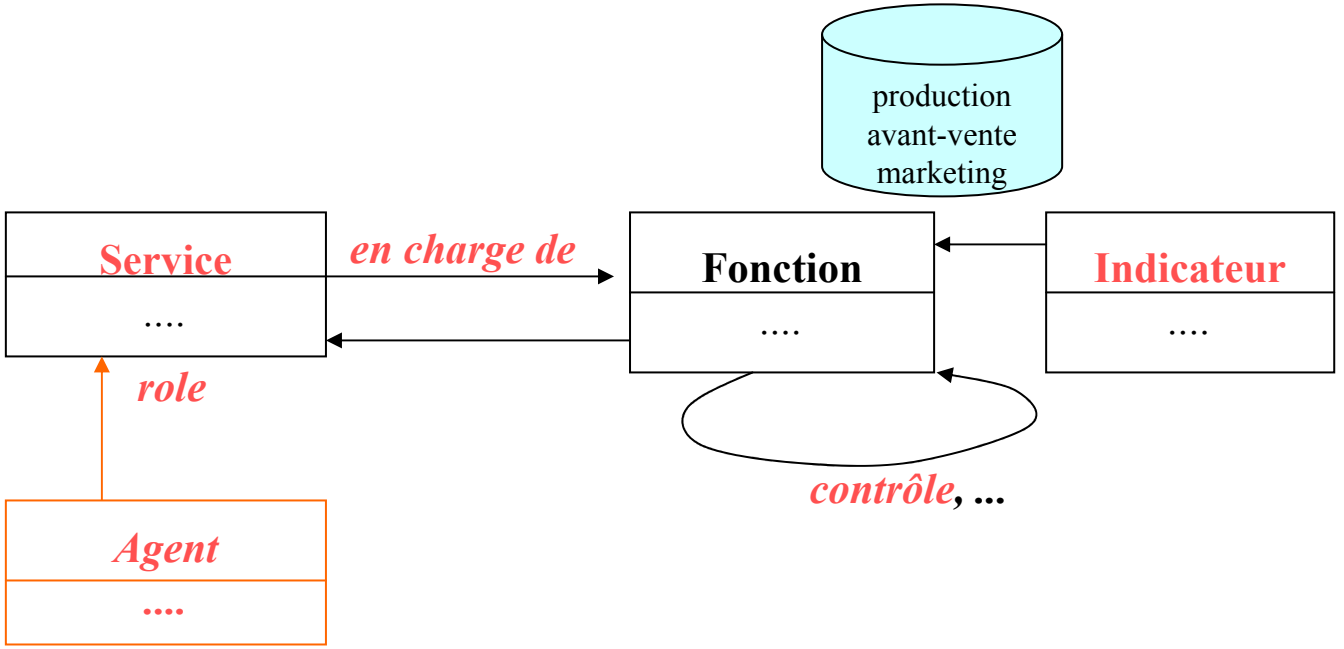


Méthode :

...

1) Modification, Fusion, … de Pattern

2) Instanciation



…

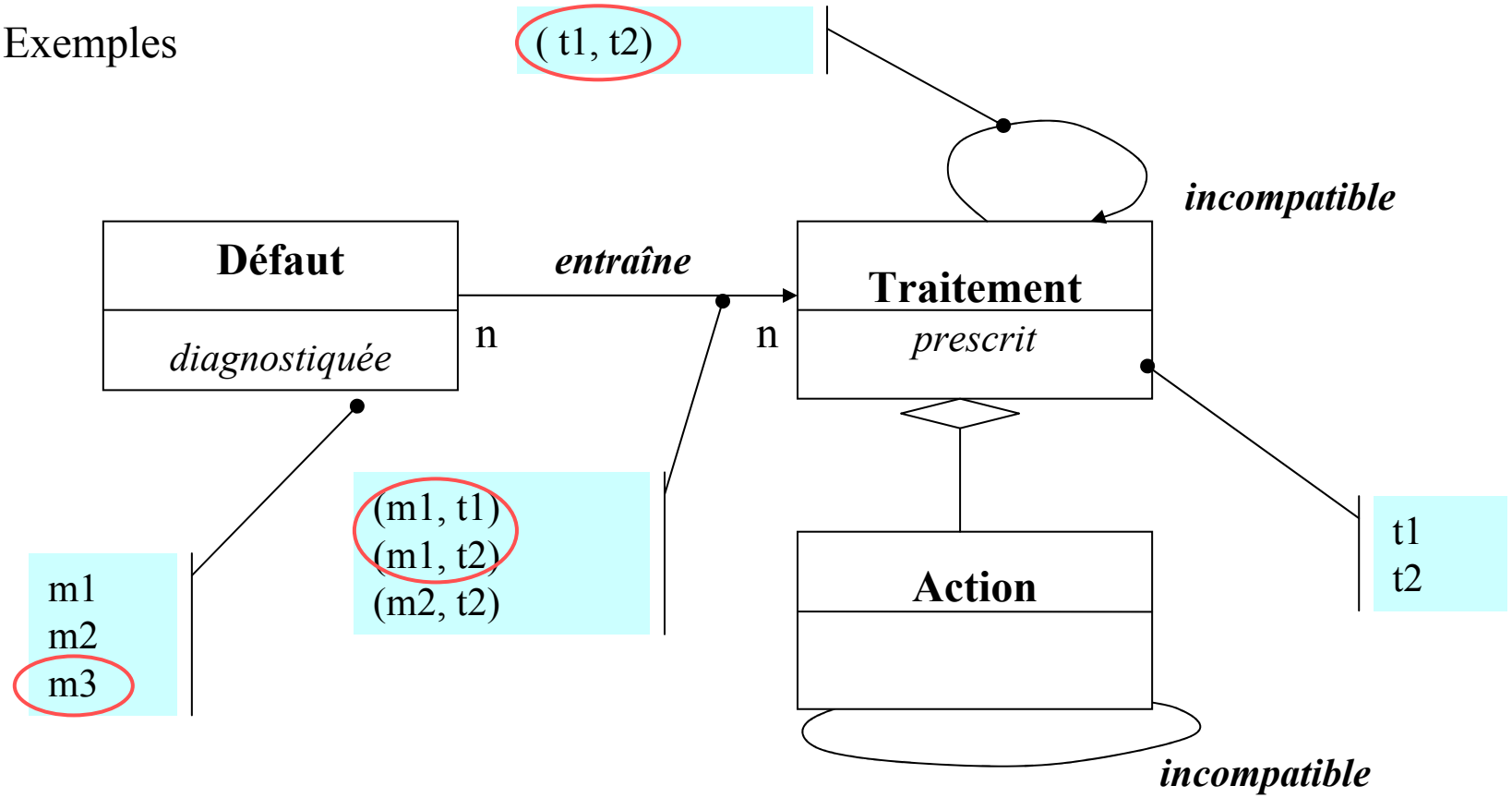
Méthode :

…

1) Construction du Modèle de Résolution = set of Patterns
2) Instanciation (→ Modèle ‘client’)

3) Test du Modèle ‘client’ (bruit, silence)

Exemples



Règle InstanciationIncomplète 1 (silence)

$\forall$ m $\in$ Défaut : $\neg \exists$ entraîne(m, t) $\Rightarrow$ Pb d'Instanciation

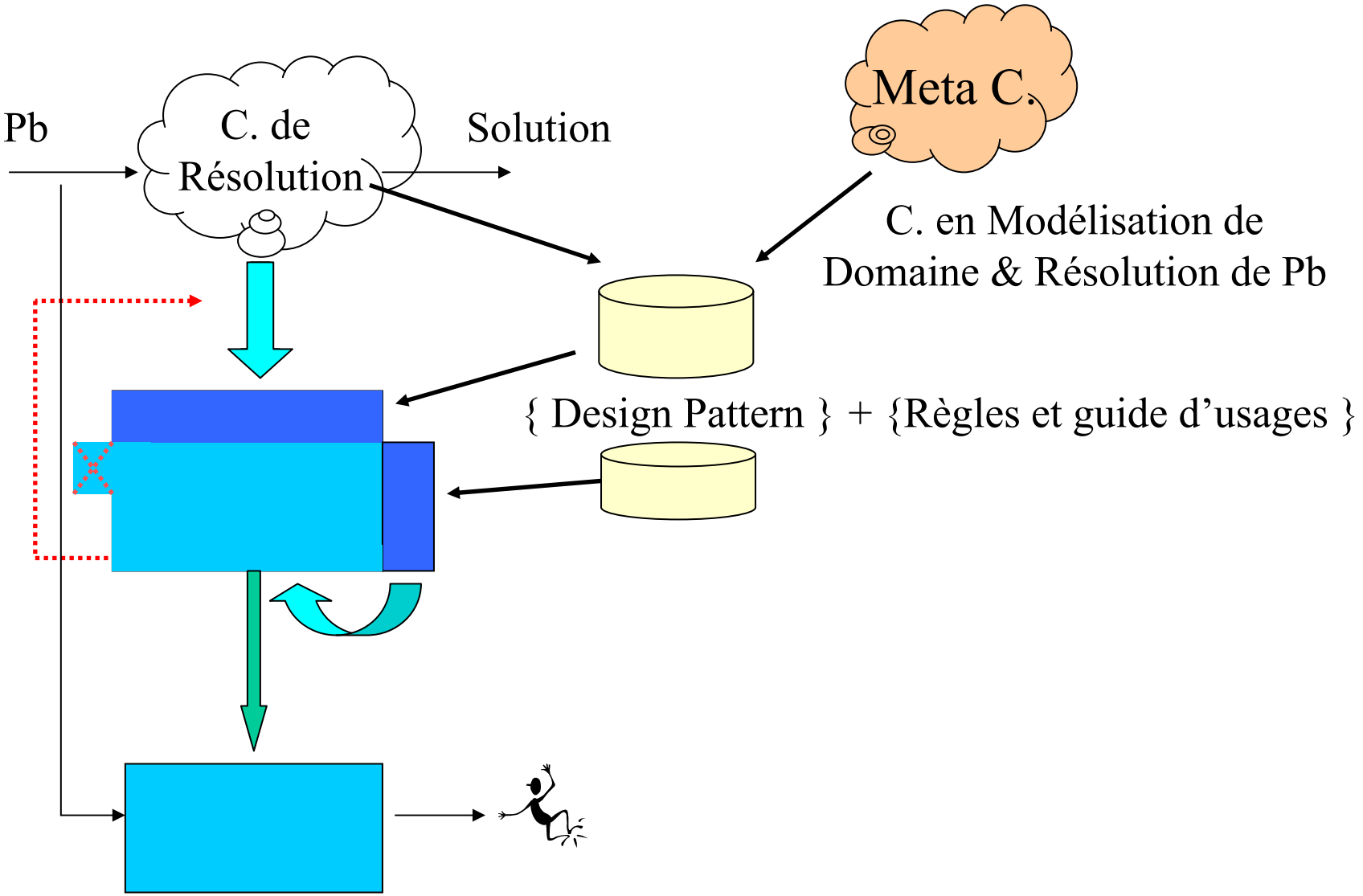
Règle InstanciationIncohérente 1 (bruit)

$\forall$ t1, t2 $\in$ Traitement : t1 . prescrit $\wedge$ t2 . prescrit $\wedge$ incompatible(t1, t2) $\Rightarrow$ Pb d'Instanciation

Ces règles découlent de la sémantique des objets de raisonnement et de leurs relations ($\in$ **pattern**) et sont à appliquer aux K. de domaine

1) Construction du Modèle de Résolution = set of Patterns
2) Instanciation (→ Modèle ‘client’)
3) Test du Modèle ‘client’ (bruit, silence)

Dès la conception du Modèle de Résolution,
on élabore le protocole de test du Modèle ‘client’



Approche par modèle

1) Construction du Schéma de Résolution,
2) Instanciation (→ Modèle 'client')
3) Test du modèle 'client'

Un modèle est une réalisation d'un méta-modèle.
(une instanciation)

L'étape 1 est une étape ***2*** !

qui s'appuie sur une étape ***1*** :

Construction du Schéma de Résolution de
« Conception de Modèles »

« Conception de Modèles » contient, par définition,
une structure (les notions de méta-modélisation),
des schémas causaux, une méthode (méthodologie de modélisation)
+
les règles permettant de tester la validité de ses instanciations
(et donc de réaliser l'étape ***3***)