Plan de l'exposé

- La construction des Systèmes à Base de Connaissances
- Le langage *ZoLa*
- *ZoLa* en détails
- Exemple d'application
- Conclusions et perspectives

Ma thèse

Le Langage *ZoLa*

- modélisation de l'expertise à l'aide de primitives de modélisation et de mécanismes de manipulation
- réflexivité structurelle et conceptuelle
- construction d'outils d'analyse du modèle

Le Modèle Conceptuel

- “Le rôle du modèle conceptuel est de décrire la méthode de résolution d’un système à base de connaissances en utilisant des termes pertinents pour les experts humaines.”
- “Le modèle conceptuel caractérise la méthode de résolution mise en oeuvre par un système ; sa fonction principale est de permettre ... comprendre cette méthode”
- “Le modèle conceptuel est un modèle de la méthode de résolution qui exploite les connaissances recueillies sur le domaine pour résoudre les problèmes”
- “Le modèle conceptuel est vu comme une construction abstraite finalisée qui permet de réduire la complexité de l’expertise en se focalisant sur certains aspects des connaissances”
- “Le modèle conceptuel constitue à la fois un cadre pour comprendre et interpréter les informations provenant de l’expert et un langage pour les formaliser en vue de construire un système”
- “Un modèle conceptuel est un modèle d’expertise non exécutable, autrement dit un modèle décrivant un processus de résolution de problèmes par l'intermédiaire d’un langage non informatique”

Spécificités du langage, objectifs

- Opérationalisation d'un MC
  défini à l'aide de primitives de modélisation
  non pré-définies
- Prototypage du MC
- Construction d'outils d'analyse du MC

Opérationalisation d'un MC
(défini à l'aide de primitives de modélisation non pré-définies)

- définition des primitives de modélisation
- définition des mécanismes de manipulations
- définition d'un contrôle
  - fixe
  - dynamique

Prototypage

- respect du principe de correspondance structurelle
- opérationalisation facile et rapide, réalisée en parallèle avec la représentation

  (sur un modèle incomplet)
- réutilisation de parties de code

  (éviter la re-programmation ex nihilo lors des modifications du modèle)

Construction d'outils d'analyse

- représentation explicite des connaissances
  (l'analyse de la base de connaissances)
- représentation explicite des opérations
  (la réflexivité structurelle)
- représentation explicite du modèle
  (la réflexivité conceptuelle)

Modèle conceptuel

Représentation et opérationalisation :

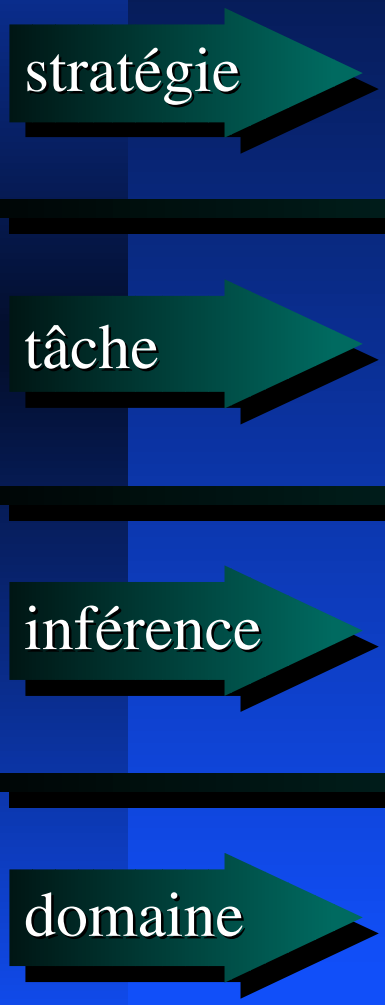
- primitives de modélisation
- mécanismes de manipulation

Primitives de modélisation
pour le contrôle et pour le domaine !

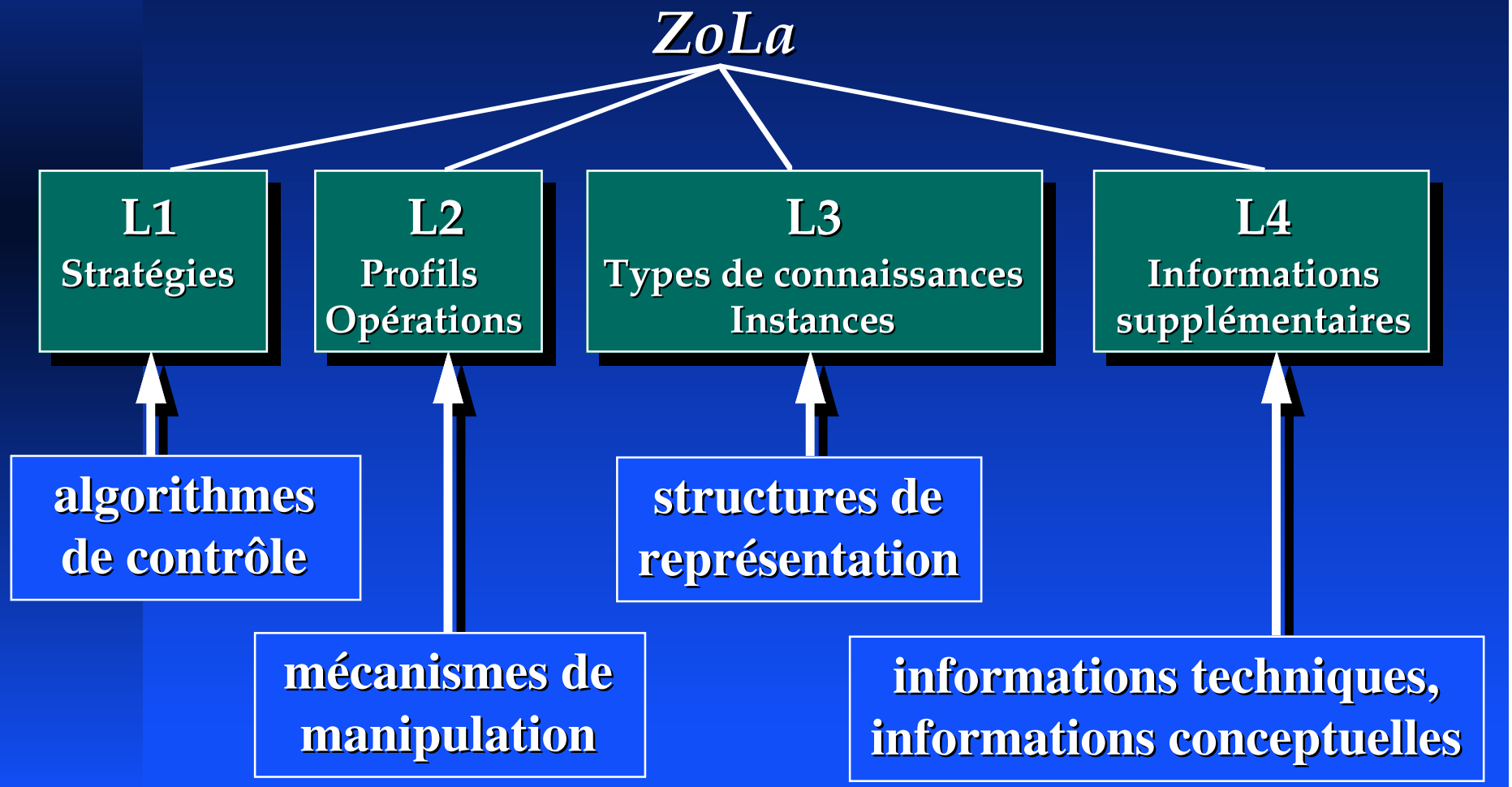
Types de connaissances à opérationaliser

- primitives de modélisation
  (structure de représentation)
- mécanismes de manipulation
  (par rapport aux types de connaissances)
- contrôle sur les mécanismes
  simple (algorithmique) / dynamique (réflexif)
- informations supplémentaires
  (pour l'analyse)

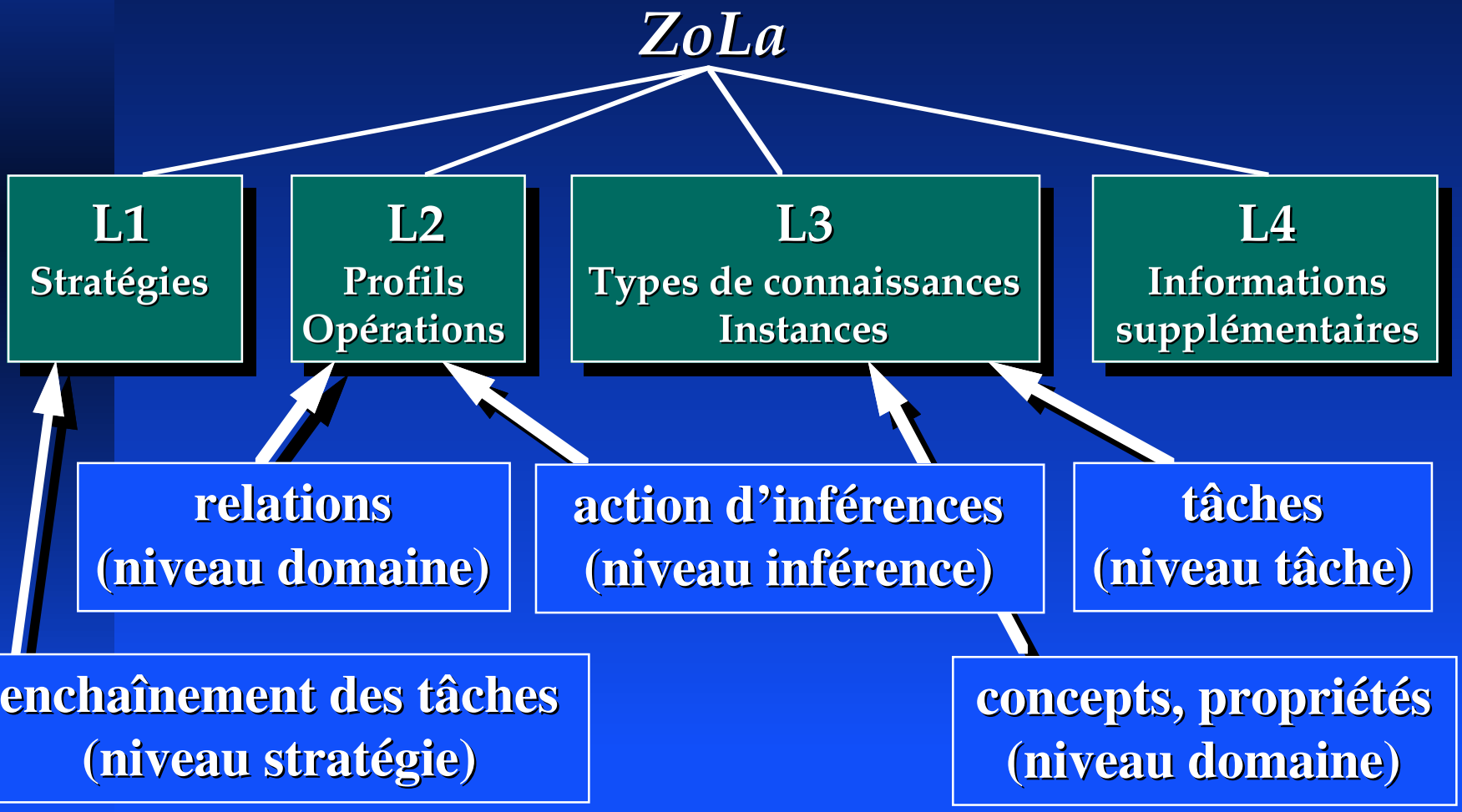
Modèle d'expertise KADS



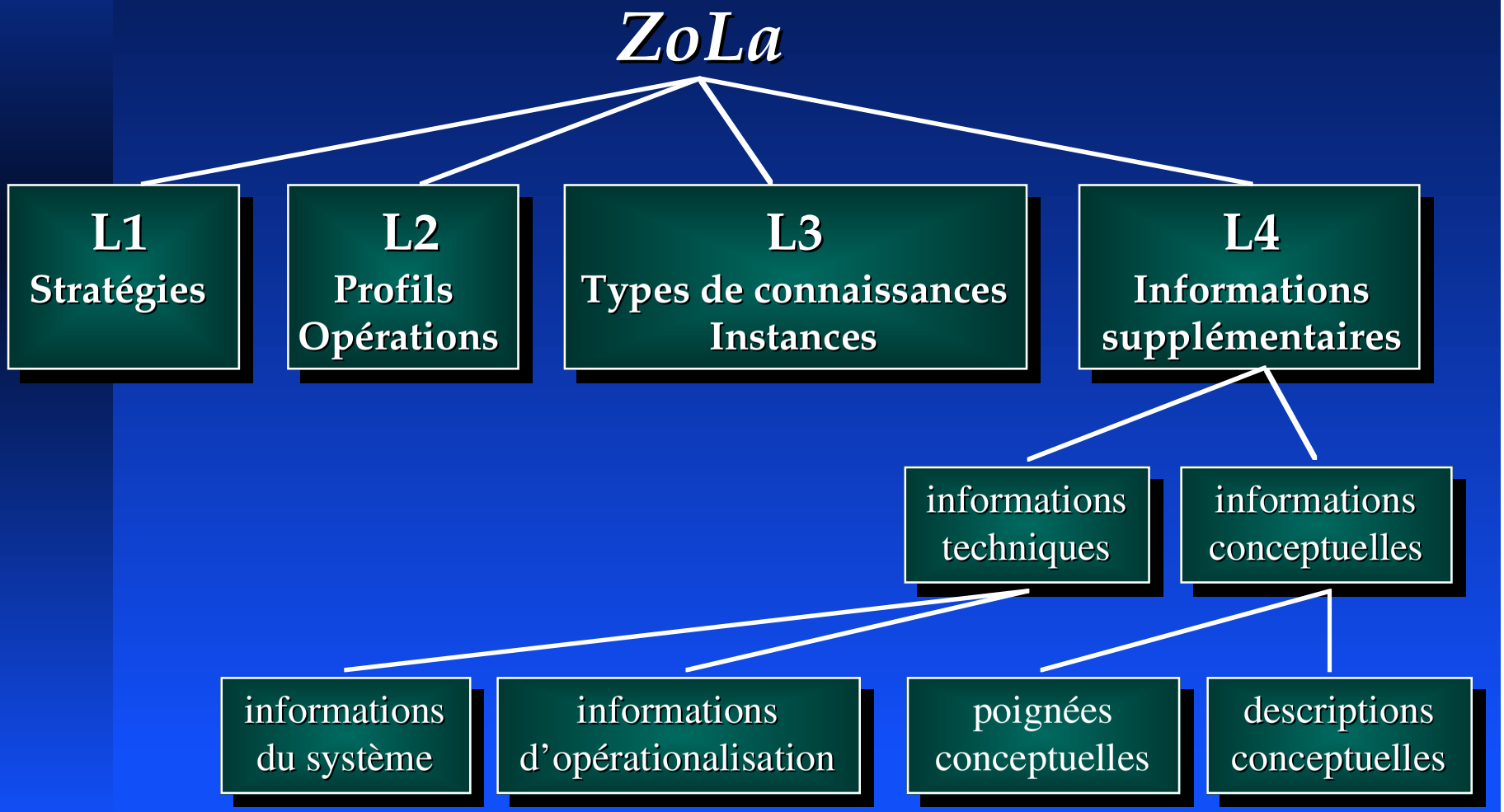
Le langage *ZoLa*



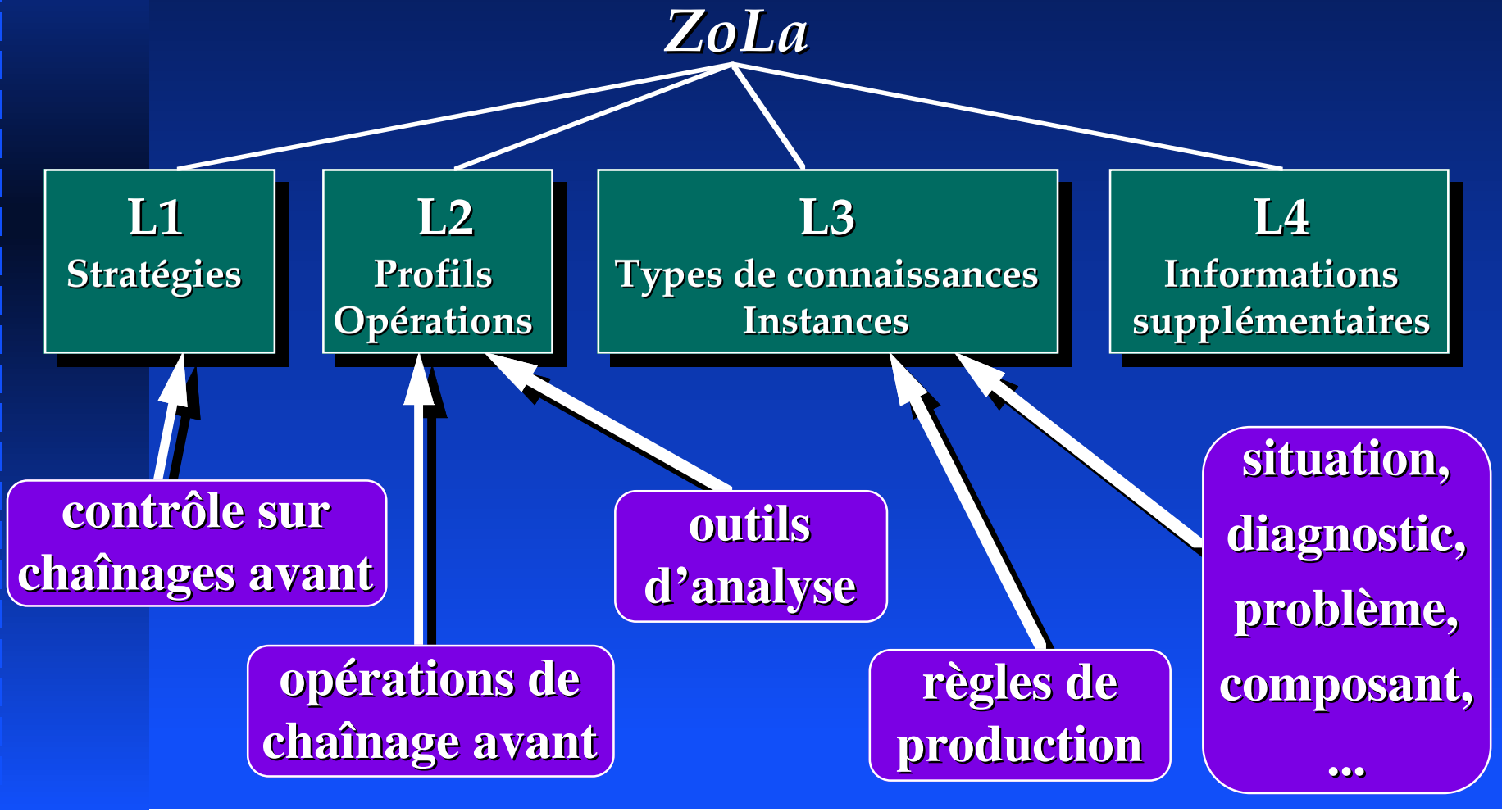
Utilisation de *ZoLa*
(méthodologie KADS)



Structure du langage



Utilisation des sous-langages de *ZoLa* dans l'application Fiona



Une règle stratégique pour DSTM

Si

Contexte de résolution de problème

Un objectif est défini
Un ensemble de moyens est défini
Un des moyens a été essayé
Le moyen a échoué

Stratégie

La stratégie pour l'objectif est saturer

Alors

Action Générique

Choisir un nouveau moyen pour l'objectif

Apport du langage *ZoLa* à DSTM

- primitives de modélisation par rapport au modèle
- opérations de manipulation des primitives de modélisation
- poignées conceptuelles, réflexivité conceptuelle, contrôle explicite, dynamique

noyau adaptable

Spécificités du langage

- représenter et opérationaliser des MC
- prototyper des MC
- proposer des outils d‘analyse

Réalisation du langage

- MC par rapport l'expertise
- pas de primitive de modélisation, pas de structure de contrôle pré-définie
- réflexivité

Comparaison des approches

- primitives de modélisations ?
- modèle d'expertise ?
- utilisation du prototypage ?
- fixes ?
- pré-définies ?
- adaptation ?
- construction ? (ascendante)
- sélection d'une bibliothèque ? (descendante)

Cadre général : le Projet MAPCAR

- construction d'un Modèle Conceptuel
- aide et guide pour l'élaboration du MC
- opérationalisation d'un MC
- prototypage, outils d'analyse
- la modélisation comme un processus constructif

ZoLa

Principe d'utilisation de *ZoLa*
(exemple de l'application Fiona)

règles de production, notions → types de L3

opérations de chaînage avant → opérations de L2

algorithme de contrôle → boucle de contrôle en L1

analyse des opérations → opérations réflexives en L2

Application Fiona

- diagnostic des pannes des robots
- aide au diagnostic
- formation des techniciens du site client

MC :

- primitives de modélisation :

  règles de production + différent notions
- mécanismes de manipulation :

  chaînage avant, enchaînement des actions

Types de connaissances de Fiona

- concept : contexte, situation, diagnostic, procedure, tache, probleme, composant, regle
- règle : regle-de-contexte, regle-de-diag, regle-post-diag, regle-proc-pour-tache, regle-definissant-une-tache, regle-de-pre-diag
- collection : liste-de-concept, base-de-concept, base-de-regle

Une règle de diagnostic de Fiona

Nom	R62
Composant	frein
Contexte-d-examen	pas-d-historique ecart-faible relance
Expliquer	oui
Diagnostic-actuel	(defaut codeur ecarte)
Situation	(ecart faible confirme) (arret avant cote prog) (sens en montee)
Nouveau-diagnostic	(defaut frein moteur avere)

Une type de règle de diagnostic de Fiona en ZoLa

```
(type `regle-de-diag `#:z:l3:t-concept:t-regle
    `composant
    `contexte-d-examen
    `situation
    `diagnostic-actuel
    `expliquer
    `nouveau-diagnostic)
```

Une instance de règle de diagnostic de Fiona en ZoLa

```
(instance `r62 `#:z:l3:t-concept:t-regle:regle-de-diag
    `nom    `r62
    `composant              `(t-composant237)
    `contexte-d-examen      `(t-contexte207
                              t-contexte215
                              t-contexte223)
    `situation              `(t-situation241
                              t-situation228
                              t-situation248)
    `diagnostic-actuel      `(t-diagnostic225)
    `expliquer              `(oui)
    `nouveau-diagnostic     `(t-diagnostic240))
```

Instances engendrées à partir d'une règle de diagnostic de Fiona

```
(instance `t-composant237 `#:z:13:t-concept:t-composant
    `nom    `t-composant237
    `termes    `frein)

(instance `t-contexte207 `#:z:13:t-concept:t-contexte
    `nom    `t-contexte207
    `termes    `pas-d-historique)

(instance `t-situation241 `#:z:13:t-concept:t-situation
    `nom    `t-situation241
    `termes    `(ecart faible confirme))

(instance `t-diagnostic225 `#:z:13:t-concept:t-diagnosti
    `nom    `t-diagnostic225
    `termes    `(defaut codeur ecarte)
    `statut    `())
```

Opérations `test-lance-tout-regle` et `test-lance-regle`

```
(operation
   'nom          'test-lance-tout-regle
   'profil       '#:z:l2:1-pourtout
   'liee         '#:z:l3:t-collection:t-base-de-regle
   'parametres   '(une-base-de-regle une-regle)
   'action       '(test-lance-regle une-regle))

(operation
   'nom          'test-lance-regle
   'profil       '#:z:l2:1-si-alors
   'liee         '#:z:l3:t-concept:t-regle
   'parametres   '(une-regle)
   'condition    '(test-regle une-regle)
   'action       '(lance-regle une-regle))
```

Opération `test-regle`
(pour les règles du type `regle-de-contexte`)

```
(operation
 'nom      'test-regle
 'profil   '#:z:l2:l-actions-liste-et
 'liee     '#:z:l3:t-concept:t-regle:regle-de-contexte
 'parametres    '(regle)
 'liste    '(

(test-aucune-condition regle 'contexte-a-ajouter
                       base-de-contextes-courantes)

(test-tout-condition regle 'type-de-probleme
                       base-de-types-de-probleme-consideres)

(test-tout-condition regle 'contexte-d-examen
                       base-de-contextes-courantes)

(test-tout-condition regle 'situation
                       base-de-situations-courantes)

(test-tout-condition regle 'diagnostic-actuel
                       base-de-diagnostic-actuel)))
```

Opération `lance-regle`
(pour les règles du type `regle-de-contexte`)

```
(operation
  'nom        'lance-regle
  'profil     '#:z:l2:l-suite
  'liee     '#:z:l3:t-concept:t-regle:regle-de-contexte
  'message  "Je lance une règle de contexte"
  'info       't
  'parametres      '(regle)
  'liste      '(

(ajoute-tout-action regle 'contexte-a-ajouter
                          base-de-contextes-courantes)

(enleve-tout-action regle 'contexte-a-enlever
                          base-de-contextes-courantes)))
```

Profil et opération
(de type primitive de manipulation)

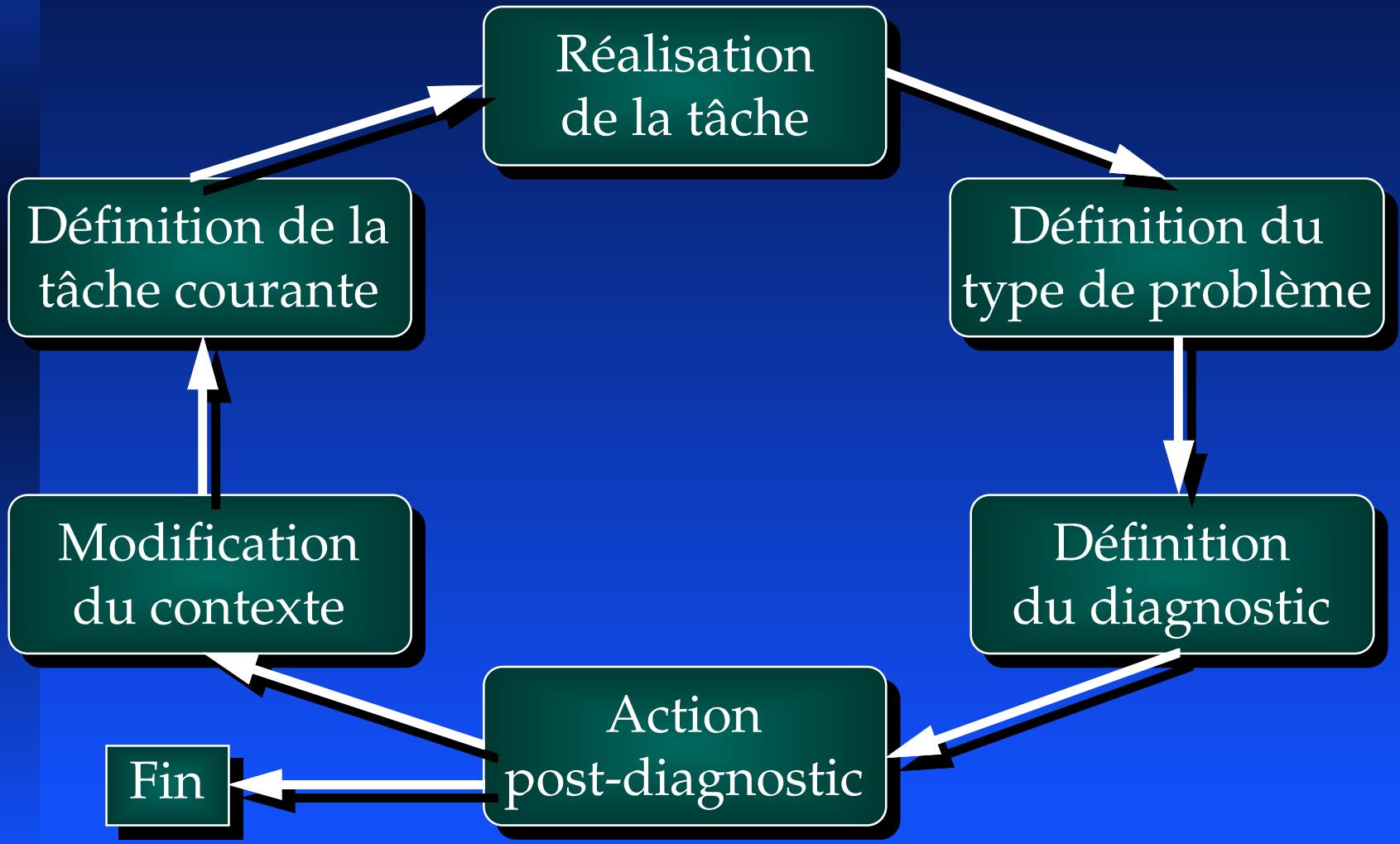
```
(profil   `#:z:12:l-ajout
 `(message
      "ajout d'un élément à une liste d`éléments")
 `(code
      (lambda (une-liste un-element)
            << code lisp du profil >>)))

 (operation
     `nom      `ajouter-element
     `profil   `#:z:12:l-ajout
     `liee     `#:z:13:t-collection)
```

Opération `classer-verifier` (du profil `1-si-alors-sinon`)

```
(operation
  `nom       `classer-verifier
  `profil    `#:z:l2:1-si-alors-sinon
  `liee      `#:z:l3
  `parametres  `(element list-A list-B
                       oper-de-selection critere)
  `condition  `(faire-operation oper-de-selection
                                critere element)
  `action      `(ajouter-element list-A element)
  `sinon       `(ajouter-element list-B element)
)
```

L'enchaînement des tâches de Fiona



Outils réflexifs pour Fiona

- Besoin : outils
  - rôle joué par une connaissance dans le diagnostic
  - analyse des liens entre des actions
- Solution : Réflexivité structurelle

  (opérations analysent la structure d'autres opérations)
- Moyen : isomorphisme entre L2 et L3
- Résultat : affinage du MC

Apport du langage ZoLa au projet Fiona

- Représentation et opérationalisation du MC : représentation explicite des différents notions du MC. (analyse)
- Prototypage : opérationaliser les différentes versions, de les tester, de détecter des améliorations possibles grâce aux outils réflexifs.
- Mise en évidence de différents aspects du modèle : outils dédiés à l'étude du MC à des fins d'affinage, différents outils ont été construits à des fins d'EIAO. Construction d'outils proposant différents points de vue sur la base de connaissances du système, outils qui

Contrôle réflexif pour DSTM

Besoin :

- ajout d'informations supplémentaires aux primitives de modélisation (L3)
- ajout d'informations supplémentaires aux opérations

Moyen :

- poignées conceptuelles
- analyse réflexive sur les informations supplémentaires

Résultat : contrôle réflexif

Réflexivité

idée :

- raisonner sur lui-même

nécessite :

- connaissances sur soi-même
- mécanismes capables de raisonner sur ces connaissances

Les définitions de la réflexivité

- Réflexivité / Introspection
- Réflexivité dynamique / statique
- Réflexivité structurelle / conceptuelle

intérêts de la réflexivité :

- Contrôle dynamique
- Outils d‘analyse

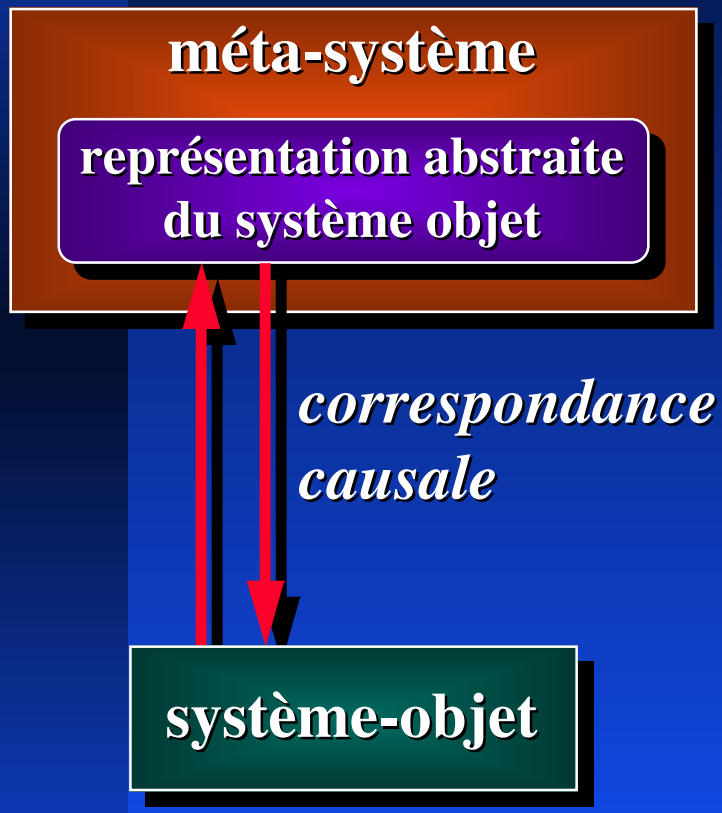
Réflexivité structurelle

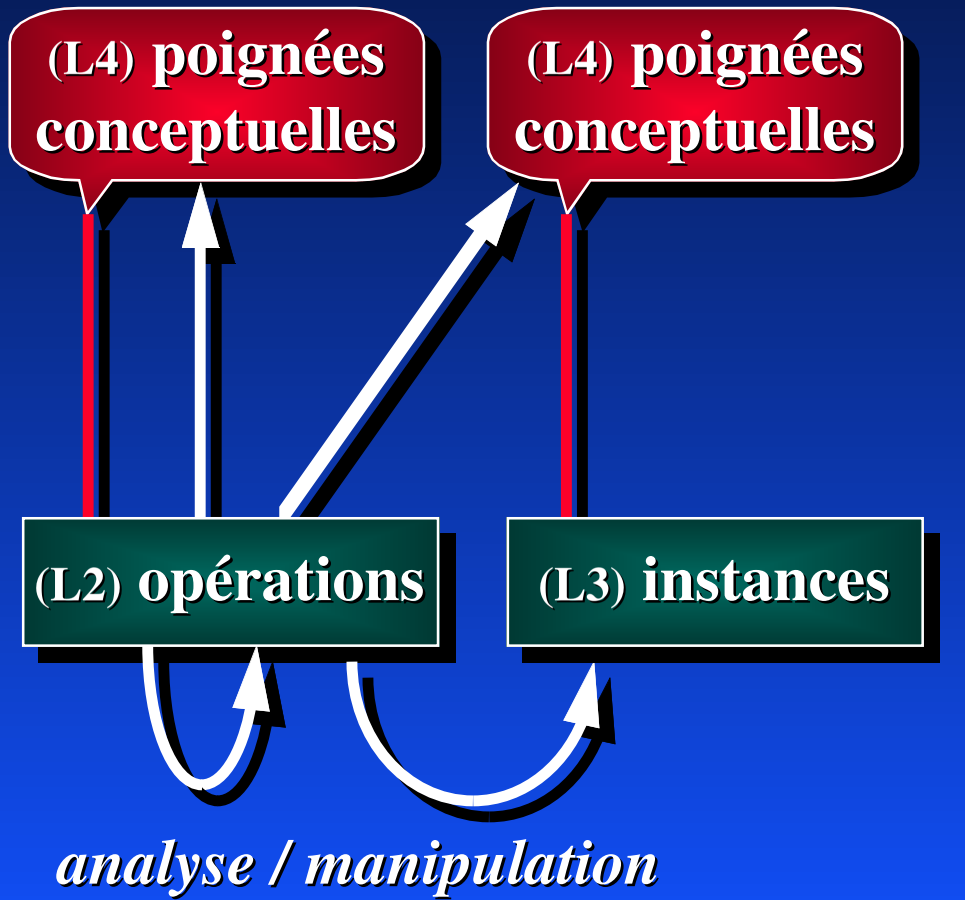
```
(operation
  'nom 'analyse-lancement
  'liee '#:z:12
  'action (
   (analyse-operation operation 'liste ...)
...)

(operation
  'nom 'lance-regle
  'liee '#:z:13:t-concept:t-regle:regle-de-diag
  'liste (
   (ajoute-action regle 'nouveau-diagnostic bd-diagnostic)
...)

(instance 'r4ode9 '#:z:13:t-concept:t-regle:regle-de-diag
  'composant              'connectique-capteur-weid
  'contexte-d-examen      'panne-intermitente
  'situation              ((nouveau defaut)
                           (shunt sur WEID))
  'nouveau-diagnostic (pb connectique capteur WEID ecarte)
...)
```

Réflexivité conceptuelle





Comparaison réflexivité

- Mapcar <-> pas seulement l'introspection
- SADE <-> modélisation plus complète (pas des règles)
- Lisa <-> réflexivité pas seulement sur les notions but/méthode
- Reflect <-> pas de connexion causale pas de niveaux de KADS

Situation du *ZoLa* par rapport à d'autres langages

ZoLa est un générateur de langage ;

langage d'opérationalisation ne reposant pas sur des primitives de modélisation pré-définies, mais permettant de définir ces primitives, adaptées à la spécificité des expertises à modéliser

entre :

- haut <-> pas de structures de modélisation pré-définies
- bas <-> plus de facilité de construire des structures de modélisation nécessaires

Opération avec des poignées conceptuelles

```
(operation
  `nom          `selectionner-une-methode-version-classique
  `profil       `#:z:l2:l-si-alors-sinon
  `liee         `#:z:l3:t-collection
  `message      "Sélectionne une méthode pour la tâche"
  `info         `t
  `pc   `(
        pc-op-selectionner-une-methode-version-classique
        pc-evaluation-echec-selectionner-une-methode
        pc-evaluation-reussite-selectionner-une-methode
        )
  `parametres   `(ensemble-de-methodes methode tache)
  `condition    `(tache-courante-associee-methodes-connues
                                                  tache)
  `action       `(selectionner-une-methode-connue
                        ensemble-de-methodes methode tache)
  `sinon        `(selectionner-une-methode-associe
                        ensemble-de-methodes methode tache)
)
```