1

ZoLa : un langage réflexif pour représenter et opérationaliser des modèles conceptuels

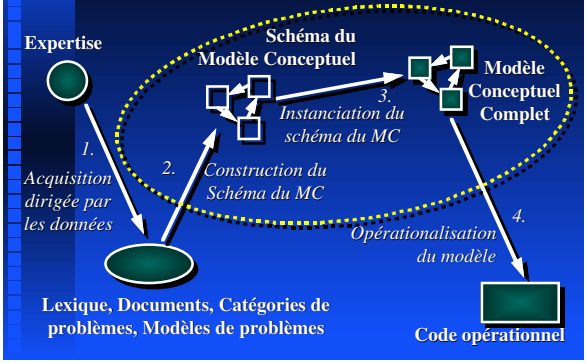
Zoltàn ISTENES
Institut de Recherche en Informatique de Nantes

2

La construction des Systèmes à Base de Connaissances

3

Construction d'un SBC

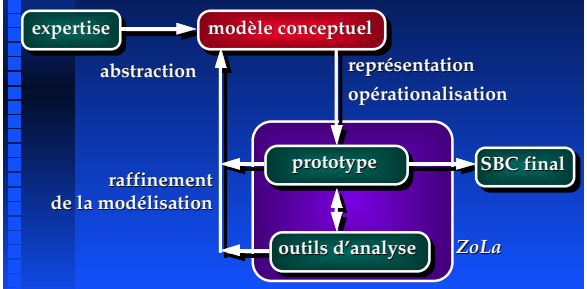


Ma thèse

Le Langage *ZoLa*

"un langage qui permet de construire un prototype par rapport à la modélisation de l'expertise adoptée et des outils d'analyse de ce prototype, pour aider à la modélisation"

Approche MAPCAR (cadre général)



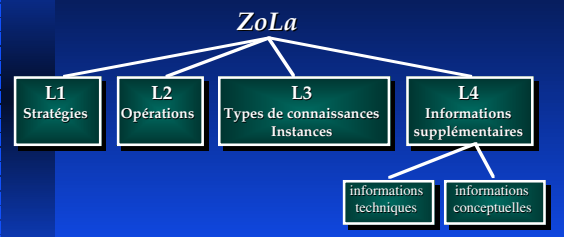
Spécifications et objectifs

Un langage qui permet :

- de construire des structures de représentation qui correspondent aux primitives de modélisation adoptées
- d'implanter les mécanismes inférentiels associés
- de modifier facilement les structures et les mécanismes par rapport à l'évolution de la modélisation
- de construire des outils d'analyse du prototype

Le langage *ZoLa*

Structure du langage



Niveaux d'un MC ≠ Sous-langages de *ZoLa*

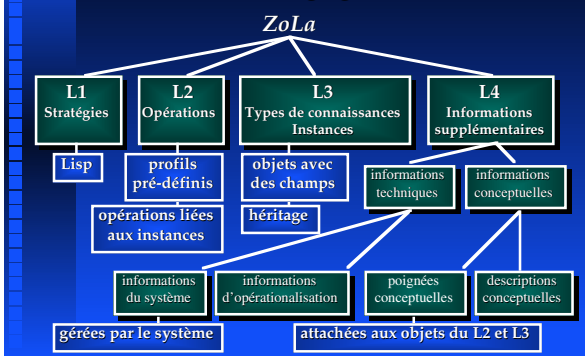
- un sous-langage de *ZoLa* ne correspond pas à un niveau d'un MC
- on utilise (tous) les sous-langages de *ZoLa* pour opérationaliser chaque niveau d'un MC

Modèle d'expertise KADS

- stratégie : raisonnement et ordonnancement des tâches
- tâche : contrôle sur les inférences
- inférence : primitives de raisonnement sur le domaine
- domaine : théorie du domaine

ZoLa en détails

Structure du langage (propriétés)



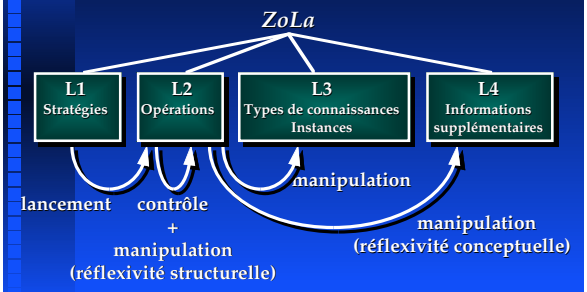
Le profil `l-si-alors-sinon`
(langage L2)

```
(profil `#:z:l2:l-si-alors-sinon
      `(message  "Je réalise Si-Alors-Sinon")
      `(parametres     ())
      `(condition      ())
      `(action         ())
      `(sinon          ())
      `(code   (lambda ()
            << code lisp du profil >> ))
)
```

emplacements pour des sous-opérations

code implicite manipulant des sous-opérations des emplacements

Structure du langage (fonctionnement)



Réflexivité

- Réflexivité structurelle
  - opérations de L2 qui manipulent directement des opérations de L2
  - isomorphisme entre L2 et L3
  - outils d'analyse
- Réflexivité conceptuelle
  - opérations de L2 qui analysent des informations conceptuelles de L4
  - isomorphisme entre L3 et L4
  - contrôle dynamique

Exemple d'application

Langage de contrôle de DSTM

Sélection Dynamique de Tâches et de Méthodes

Modélisation du contrôle :

- primitives de modélisation :
  - tâches
  - méthodes
- mécanismes de manipulation :
  - mécanismes de sélection
  - algorithme de contrôle

Définition des types de connaissances (langage L3)

```
(type 'methode '13
  'preconditions
  'resultats-possibles
  'ressources-necessaires
  'contexte-favorable)

(type 'methode-decomposition 'methode
  'se-decompose-en)

(type 'methode-operation 'methode
  'est-realisee-par)
```

champs

héritage

nom du type

type père

Définition d'une instance
(langage L3)

type de l'instance

```
(instance   `m12  `#:z:l3:methode:methode-operation
 `nom       `method-12
 `preconditions              `(fait-contexte-1)
 `resultats-possibles        `(resultat-t1
                               resultat-t2
                               resultat-t3)
 `ressources-necessaires     `(fait-ressource-1
                               fait-ressource-4)
 `est-realisee-par           `methode-lisp-12
 `contexte-favorable         `(fait-favorable-1)
 )
```

champs *valeurs*

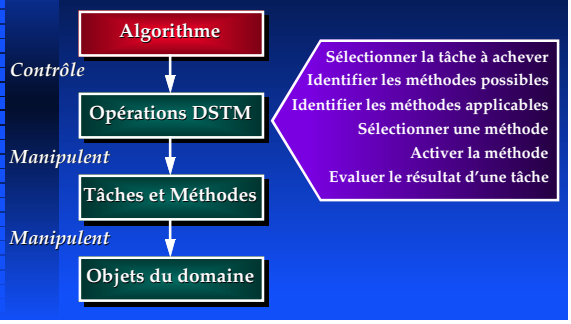
Définition d'une opération
(langage L2)

nom de l'opération

```
(operation
 `nom          `selectionner-une-methode
 `profil       `#:z:l2:1-si-alors-sinon
 `liee         `#:z:l3:t-collection
 `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)
 )
```

nom des sous-opérations *paramètres*

Architecture de DSTM
(avec un contrôle algorithmique)

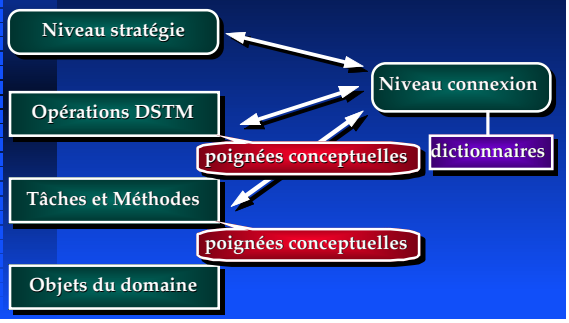


22

Niveau contrôle générique

23

Fonctionnement de DSTM

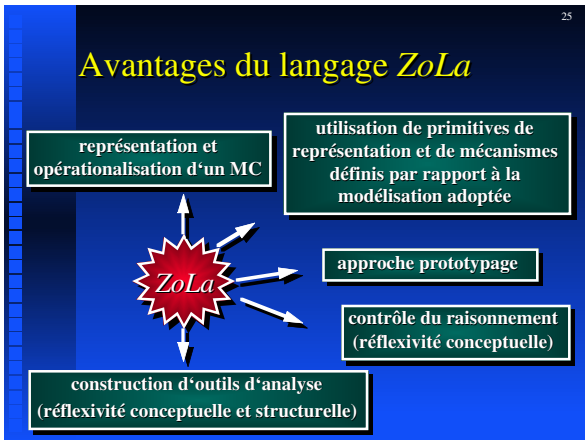


24

Conclusions et perspectives

25

Avantages du langage *ZoLa*



26

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

Langages d'opérationalisation de haut niveau (Aide, Lisa, etc.)
+ utilisables "directement"
- contraignent la modélisation

ZoLa est un générateur de langage

Langages de bas niveau (Lisp, Orienté Objet, Prolog, etc.)
+ opérationalisation d'une modélisation quelconque
- pas étudiés pour (coût en temps)

27

Travaux autour du langage *ZoLa*

- travaux réalisés
  - application Fiona
  - version opérationnelle de DSTM
- travaux en cours
  - noyau DSTM
    - adaptation de DSTM à Mona : ZTM
    - Projet Elmer : application EMMA (EIAO)
  - langages de représentation du domaine
    - entité-association
    - langage relationnel

Perspectives

- **amélioration du langage *ZoLa***
- **construction d'une bibliothèque de profils et d'opérations**
- **utilisation du langage *ZoLa* pour concevoir des noyaux modifiables**

Fin ...