

EdiNoS version 3.2.4

Manuel technique

Auteur : Luc Poittevin
Date : 02/11/2011

Le langage de la Mémoire de Travail

1) Langage des faits déclarés (description des actions):

Dans le champ « description d'une action », tout ce qui est entre guillemets est un commentaire.

```
<déclaration> :: [<uneLigneDéclaration> ';' ]
<uneLigneDéclaration> :: <uneDéclarationDeTypeDePropriété>
                        | <uneDéclarationDeDomaineDePropriété>
                        | <uneDéclarationDeRelation>
<uneDéclarationDeTypeDePropriété> :: <unePropriété> <unEspace> <unType>
<uneDéclarationDeDomaineDePropriété> :: <uneConstante> | '(' <uneListeDeValeurs> ')'
                        | <unePropriété> '|' '(' <uneListeDeValeurs> ')'
<uneDéclarationDeRelation> :: <unePropriété> <unEspace> <unePropriété>
<unePropriété> :: <uneConstante> | <unObjet> '|' <unAtome>
<unAtome> :: <uneChaîne> | <uneChaîne> '|' <unAtome>
<unType> :: 'ENTIER' | 'CHAINE' | 'REEL'
<uneListeDeValeurs> :: <uneValeur> | <uneListeDeValeurs> <unEspace> <uneValeur>
<uneConstante> :: <uneChaîne>
<uneValeur> :: <uneChaîne>
<unObjet> :: 'o' <unChiffre>
```

Remarque: les propriétés sont monovaluées.

2) Langage des faits renvoyés :

A la suite de l'exécution d'une action, EDINOS reçoit un ensemble de faits dans le format suivant. Ces faits sont stockés dans la MT.

```
<unFaitRenvoyé> :: <uneAffectation> | <unePropriété>
<uneAffectation> :: <unePropriété> <unEspace> <unePropriété>
<unePropriété> :: <uneChaîne> | <uneChaîne> '|' <unePropriété>
```

3) Langage des prémisses des règles de choix :

```
<unePrémisse> :: <uneFormule> <unConnecteurLogique> <uneFormule>
                | <uneFormule>
<uneFormule> :: '(' <uneFormule> <unConnecteurLogique> <uneFormule> ')'
                | <uneProposition>
<uneProposition> :: <unePropriété> <unOpérateurDeComparaison> <unePropriété>
                | 'NON' '(' <unePropriété> ')' | <unePropriété>
<unePropriété> :: <uneChaîne> | <uneChaîne> '|' <unePropriété>
<unConnecteurLogique> :: 'ET' | 'OU'
<unOpérateurDeComparaison> :: '=' | '<' | '>' | '<=' | '>=' | '<>'
```

Exemple de prémisse :

(P1.P2.P3.P4 > Q1.Q2.Q3) ET (fait2.fait5 = 4 ET fait4 = 'V') ET fait1.fait6

Les propriétés non comparées (<fait1.fait6> dans l'exemple ci-dessus), sont interprétées comme des valeurs booléennes (dans l'exemple, est-ce que <fait1> possède la propriété <fait6> ?). Quand la propriété ou la relation n'existe pas, la proposition est classée fausse (ex: si <P3> n'est pas trouvée dans <P1.P2.P3.P4>, la prémisse de l'exemple ci-dessus est fausse). De même, si une propriété "typée" n'est pas comparée, la proposition est classée fausse (ex: si <fait1.fait6> était typée, par exemple ayant été déclarée comme étant une chaîne, la formule ci-dessus serait fausse également).

4) Exemples :

Exemple 1:

Déclaration de l'action <infos>:
o1.ville CHAINE;
o1.âge ENTIER;
o1.sexe.(masculin féminin);

Contenu du champ « actions » :
evaluer: infos sur: enfant1

L'exécution de cette action pourra alors renvoyer par exemple les trois faits suivants:
enfant1.ville Villeurbanne
enfant1.âge 32
enfant1.sexe.masculin

Contenu du champ « règles de choix » :
si: enfant1.ville = 'Villeurbanne' alorsAller: n1;
si: (enfant1.âge > 25) ET enfant1.sexe.masculin alorsAller: n2;

Exemple 2:

Déclaration de l'action <caisseRattachement>:
o1.departement.CRAM CHAINE;
o2.departement o1.departement;

Contenu du champ « actions » :
evaluer: caisseRattachement sur: affilié et: conjoint

Faits renvoyés:
affilié.departement.CRAM Marseille
conjoint.departement affilié.departement

Contenu du champ « règles de choix » :
si: conjoint.departement.CRAM = 'Lyon' alorsAller: n1;

La règle ci-dessus ne sera pas activé pour cet exemple, puisque <conjoint.departement.CRAM> vaut <Marseille>.