EdiNoS version 3.4.2

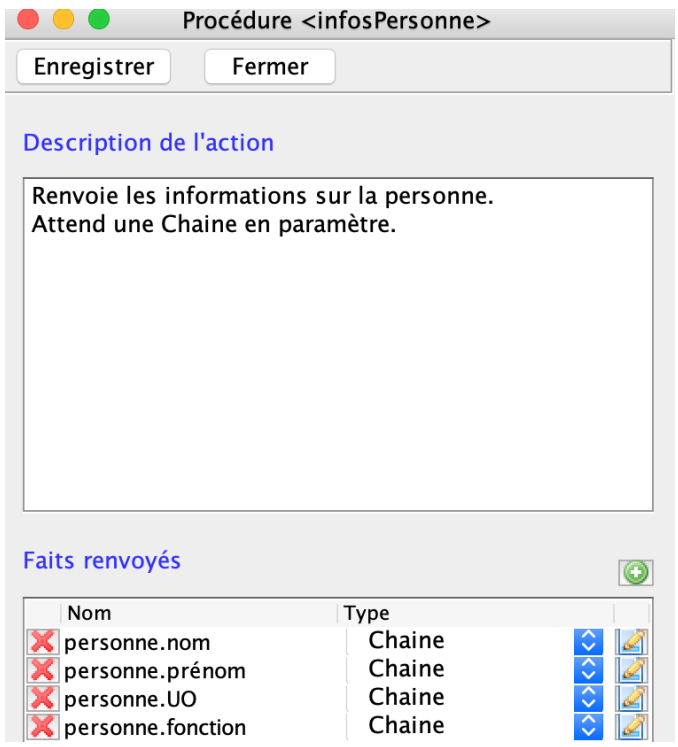
Manuel technique

Auteur : Luc Poittevin
Mise à jour du : 08/05/2019

Le langage de la Mémoire de Travail

1) Déclaration des faits renvoyés par une action :

Depuis la version 3.4.0 d'EdiNoS, les faits renvoyés par une action sont décrits et saisis dans une liste séparée, lors de la saisie d'une action :



4 types de faits sont possibles :

- Proposition
- Chaine
- Entier
- Réel

Une *proposition* est vraie ou fausse.
Lorsque l'action sera exécutée, celle-ci renverra une *valeur* pour les faits de type *Chaine*, *Entier* ou *Réel*. Ces valeurs pourront être comparées dans les règles de choix de nos (voir exemple plus loin).
Se référer également au manuel utilisateur.

2) Langage des faits renvoyés :

A la suite de l'exécution d'une action, EdiNoS reçoit un ensemble de faits dans le format suivant.

<unFaitRenvoyé> :: <uneAffectation> | <unePropriété>
<uneAffectation> :: <unePropriété> <unEspace> <unePropriété>
<unePropriété> :: <uneChaine> | <uneChaine> '.' <unePropriété>

Ces faits sont stockés dans la Mémoire de Travail (MT).

3) Langage des prémisses des règles de choix :

Les faits stockés dans la mémoire de travail sont utilisés lorsqu'un graphe est lancé, au moment du choix du nos suivant.
Les règles de choix contiennent des *prémisses* qui doivent avoir le format suivant :

<unePrémisse> :: <uneFormule> <unConnecteurLogique> <uneFormule>
| <uneFormule>
<uneFormule> :: '(' <uneFormule> <unConnecteurLogique> <uneFormule> ')'
| <uneProposition>
<uneProposition> :: <unePropriété> <unEspace> <unOpérateurDeComparaison> <unEspace>
<unePropriété> | <unePropriété> <unEspace> <motCléInconnu> |
'NON' '(' <unePropriété> ')' | <unePropriété>
<unePropriété> :: <uneChaine> | <uneChaine> '.' <unePropriété>
<unConnecteurLogique> :: 'ET' | 'OU'
<unOpérateurDeComparaison> :: '=' | '<' | '>' | '<=' | '>=' | '<>'
<motCléInconnu> : 'inconnu'

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). Il est possible de tester cela avec le mot-clé <inconnu>. Ex : P1.P2.P3 inconnu.

De même, si une propriété « typée » (c'est-à-dire de type *Chaine*, *Entier*, ou *Réel*) n'est pas comparée à une valeur, la proposition est classée fausse (ex: si <fait1.fait6> était typée, par exemple ayant été déclarée comme étant une chaine, la formule ci-dessus serait fausse également).

4) Exemple :

Déclaration de l'action <infos> :
personne.ville Chaine
personne.âge Entier
personne.sexe.masculin Proposition
personne.sexe.féminin Proposition

Contenu du champ « actions » du nos :
evaluer: infos;

Lors de l'appel à une action, un paramètre (qui peut être un fait ou une valeur) peut être passé à l'action.
Par exemple si l'on veut appeler la procédure <infos> avec le paramètre <identifiantPersonne> :
evaluer: infos identifiantPersonne;
Si l'on veut appeler la procédure <infos> avec la valeur <1234> :
evaluer: infos "1234";

Selon la déclaration de l'action, l'exécution de l'action <infos> pourra renvoyer par exemple les trois faits suivants :
personne.ville "Villeurbanne"
personne.âge 32
personne.sexe.masculin

Exemple de contenu du champ « règles de choix » du nos :
si: personne.ville = "Villeurbanne" alorsAller: n1;
si: (personne.âge > 25) ET personne.sexe.masculin alorsAller: n2;