Représentation des ontologies

Plan :

- Représentation formelle des ontologies
- Représentation de concepts génériques : introduction
- Représentation de concepts génériques : compléments
- Représentation de relations
- Représentation de concepts individuels
- Exercices

Représentation formelle des ontologies

Les ontologies sont représentées au moyen de langages formels dédiés, offrant des structures de données adaptées à la représentation de concepts. Parmi ces langages, on distingue :

- les **langages d'échange** d'ontologies sur le Web, dont la syntaxe est basée sur le langage XML.
- les **langages opérationnels** qui implémentent les ontologies à des fins d'inférences, pour constituer un composant d'un système d'information.

Les langages opérationnels se distinguent, à leur tour, par les services inférentiels qu'ils apportent. De manière générale, ces services permettent de raisonner sur :

- le contenu de l'ontologie elle-même, pour en vérifier la cohérence et aider à sa construction.
- des données exprimées au moyen des notions de l'ontologie, pour déduire de nouvelles connaissances.

Dans ce cours, nous utilisons le langage opérationnel DefOnto, défini et implanté au LaRIA (Laboratoire de Recherche en informatique d'Amiens) de l'Université de Picardie Jules Verne. Plus exactement, nous utilisons une version simplifiée de DefOnto, dans laquelle les aspects réflexifs du langage ont été gommés. Du reste, la complète maîtrise du langage n'est pas recherchée ! L'objectif poursuivi avec la présentation de DefOnto est d'illustrer la puissance d'expression de ces langages opérationnels, de montrer leur apport mais aussi les problèmes liés à leur utilisation.

DefOnto permet de représenter des concepts génériques, des relations et des concepts individuels. À chaque type d'entité conceptuelle correspond une construction du langage particulière, définie au moyen d'un « constructeur » : DefGenConcept, DefRelation ou DefIndConcept.

Représentation de concepts génériques : introduction

Le constructeur **DefGenConcept** est utilisé pour représenter un concept générique, comme dans l'exemple du concept `DOCUMENT ÉLECTRONIQUE` en figure 1 (les mots-clefs du langage apparaissent en caractères gras). Pour vérifier que l'entité de représentation – que nous appelons également « concept générique » - représente bien un concept générique, nous

montrons qu'elle permet de rendre compte des trois composants d'un concept : son, ou ses, terme(s), sa notion et son extension.

Le terme (vedette) exprimant le concept est représenté par le nom donné à l'entité, ici : *DocumentElectronique*. Ce nom joue le rôle d'identificateur pour l'entité, ce qui permet d'y faire référence dans d'autres représentations (*cf.*, dans l'exemple, la référence aux concepts *Format* et *DocumentPapier*.

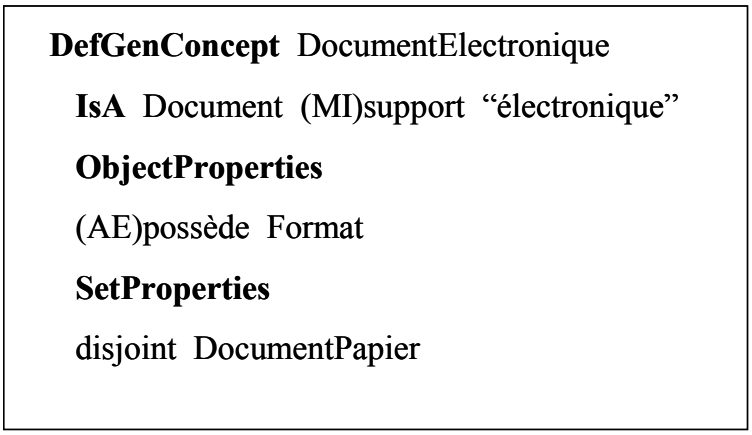

FIG 1 – Définition d'un concept générique

La notion du concept – sa signification – est représentée au moyen de propriétés. Dans notre exemple, la notion du concept `DOCUMENT ÉLECTRONIQUE` est représentée par deux propriétés :

- La première propriété, introduite par le mot-clef « IsA », représente le fait qu'*un document électronique est un document qui a un support électronique*. L'équivalent en logique, de cette propriété est l'axiome suivant :
  $\forall x\ DocumentElectronique(x) \equiv Document(x) \wedge support(x, électronique)$.
  Avec cette propriété, nous disposons d'une Condition Nécessaire et Suffisante d'appartenance pour le concept `DOCUMENT ÉLECTRONIQUE`, qui se traduit par deux conditions nécessaires :
  - Un document électronique est nécessairement un document un document :
    $\forall x\ DocumentElectronique(x) \supset Document(x)$
  - Un document électronique a nécessairement un support électronique :
    $\forall x\ DocumentElectronique(x) \supset support(x, électronique)$.

  et une condition suffisante :
  - Il suffit qu'un document ait un support électronique pour qu'il s'agisse d'un document électronique :
    $\forall x\ Document(x) \wedge support(x, électronique) \supset DocumentElectronique(x)$
- La seconde propriété, figurant après le mot-clef « ObjectsProperties », signifie que : *tout document électronique possède un format*, et a pour équivalent en logique :
  $\forall x\ DocumentElectronique(x) \supset \exists y\ Format(y) \wedge possède(x, y)$.

Note : les lettres 'A' et 'E', précédant le nom de la relation *possède*, tiennent respectivement pour les quantificateurs universel (A = All) et existentiel (E = Exist). La syntaxe de DefOnto est ainsi une variante de la notation de la logique du premier ordre. La lettre 'I' (I = Individu) indique la présence d'une constante. La lettre 'M' représente pour sa part un peudo-quantificateur. Lorsqu'on paraphrase en français, le M introduit une relative et se lit « qui … » ou « dont … » ou encore « ayant … ».

L'entité *DocumentElectronique* représente également l'ensemble, extension du concept DOCUMENT ÉLECTRONIQUE, puisqu'il est possible de lui attribuer des propriétés. Ces propriétés sont séparées du reste de la définition par le mot-clef « SetProperties ». L'ensemble des documents électroniques est déclaré disjoint de l'ensemble des documents papier. La relation pré-définie *disjoint* établit un lien entre deux ensembles, considérés comme des individus, mais le double quantificateur II a été omis, ce qui correspond à une simplification permise par DefOnto. Ce lien a pour équivalent en logique, l'axiome suivant :
$\forall x$ *DocumentElectronique* $\supset \neg$*DocumentPapier*.

On reste bien dans le cadre de la logique du premier ordre !

Représentation de concepts génériques : compléments

Dans cette section, nous exposons plus complètement la puissance d'expression du constructeur DefGenConcept en l'illustrant au moyen de nouveaux exemples. Ces exemples sont rassemblés dans la figure 2.

DefGenConcept Personne
IsA Animé
ObjectsProperties
(AE)nom String
(AE)age Number

DefGenConcept Végétarien
IsA Personne
ObjectsProperties
(VR)mange FruitEtLégume

DefGenConcept Triangle
IsA Polygone (ME(3,3))possède Côté
IsA Polygone (ME(3,3))possède Angle

FIG 2 – Autres exemples de concepts génériques

Dans la définition du concept *Animé*, le mot-clef « IsA » est suivi du super type *Animé*, sans indication d'une différence comme dans l'exemple du concept *DocumentElectronique*. Il est ainsi représenté simplement le fait qu'*une personne est un animé* (*i.e.*, un être doué de vie). L'équivalent en logique fait apparaître une implication, à la place d'une équivalence logique :
$\forall x$ *Personne(x)* $\supset$ *Animé(x)*.

Toujours dans la définition du concept *Animé*, il est également représenté que *toute personne a un nom, codé par une chaîne de caractères, et un age, codé par un nombre*. 'String' et 'Number' correspondent à des types de données – les deux seuls offerts par DefOnto – que l'on peut assimiler à des éléments terminaux pour le langage : il n'est en effet pas possible d'attribuer des propriétés à ces éléments.

Le concept *Végétarien* introduit un nouveau quantificateur VR (signifiant : « Value Restriction ») pour représenter le fait qu'*un végétarien ne mange rien d'autre que des fruits et des légumes*, ce qui se traduit en logique par l'axiome suivant :
$\forall x \forall y$ *Végétarien(x)* $\supset$ *(mange(x, y)* $\supset$ *FruitEtLégume(y))*.

Le concept *Triangle* illustre pour sa part l'utilisation de deux conditions nécessaires et suffisantes, et de restrictions de cardinalités associées aux quantificateurs existentiels pour représenter les propriétés suivantes : *un triangle est un polygone possédant au minimum 3 et au maximum 3 côtés* ; *un triangle est un polygone possédant exactement 3 angles*.

Note : une définition de concept générique peut contenir un nombre quelconque de liens de subsomption introduisant des conditions nécessaires et suffisantes, ou de simples conditions nécessaires, d'appartenance.

Représentation de relations

Les relations, comme nous venons de le voir, interviennent dans la définition des concepts. La notion de « relation » est en fait homogène à celle de « concept », puisqu'une relation est exprimée en langue par un (ou des) terme(s), qu'elle possède une signification, sa notion, et qu'elle admet pour extension un ensemble de n-uplets (*n* étant l'arité de la relation, c'est-à-dire son nombre d'arguments). Traditionnellement, toutefois, la représentation des relations reste plus simple que celle des concepts.

C'est ce que l'on constate en DefOnto. Le constructeur **DefRelation** est dédié à la représentation des relations (*cf.* figure 3). Seules des relations binaires sont considérées. La représentation d'une relation comporte deux parties. Sont ainsi exprimés :

- Un – unique – lien de subsomption, de type IsA, par rapport à une sur-relation. Par exemple (*cf.* figure 3), la relation *mange* spécialise la relation *consomme*, ce qui signifie que tout couple de la relation `MANGE` est également un couple de la relation `CONSOMME`. En logique, cette propriété s'exprime par l'axiome suivant :
  $\forall x \forall y$ *mange(x, y)* $\supset$ *consomme(x, y)*.
- La signature de la relation, qui revient à contraindre le type des arguments des couples de la relation. Pour reprendre l'exemple de la relation *mange* : cette relation a pour domaine le concept générique *Personne*, ce qui signifie que *l'entité qui mange est nécessairement une personne*, et sa portée (en anglais : « range ») est le concept générique *Aliment*, ce qui signifie que *l'entité mangée est nécessairement un aliment*. Il correspond en logique à ces deux propriétés les axiomes suivants :
  $\forall x \forall y$ *mange(x, y)* $\supset$ *personne(x)*
  $\forall x \forall y$ *mange(x, y)* $\supset$ *aliment(y)*

Note : les relations **domain** et **range** sont deux relations prédéfinies du langage.

```
DefRelation mange
  IsA consomme
  RelationProperties
  domain Personne
  range Aliment

DefRelation nom
  IsA attributSocial
  RelationProperties
  domain Personne
  range String
```

FIG 3 – Exemples de définitions de relations

Représentation de concepts individuels

DefOnto permet de représenter des concepts individuels, au moyen du constructeur **DefIndConcept**.

La représentation d'un concept individuel (*cf.* figure 4) consiste en une liste de faits impliquant une relation et, suivant le cas, une donnée (ex : une chaîne de caractères) ou un concept individuel. On notera qu'un concept individuel peut être subsumé par plusieurs concepts génériques.

```
DefIndConcept PierreCurie
  IsA Chercheur
  IsA Père
  ObjectProperties
  nom "Curie"
  mariéA MarieCurie
  domaineDeRecherche "physique nucléaire"
  fille IrèneJoliotCurie
```

FIG 4 – Définition d'un concept individuel

Exercices

Exercice 1

Un concept pour lequel une (ou plusieurs) Condition(s) Nécessaire(s) et Suffisante(s) d'appartenance peuvent être associées est dit concept « défini ». A contrario, si seule une Condition Nécessaire d'appartenance peut être exprimée, le concept est dit concept « primitif ». Trouver des exemples de concepts définis et de concepts primitifs. Dans cet exercice, on se contentera d'exprimer les notions en français.

Solution de l'exercice 1

Les définitions qui suivent sont extraites du dictionnaire : Le Petit Larousse Illustré, 1995.

Exemples de concepts définis :

- Un « moteur » est un appareil qui transforme en énergie mécanique d'autres formes d'énergie
- Une « scie » est un instrument de menuiserie portant sur un côté une suite de dents tranchantes et servant à débiter le bois.
- Une « table » est un meuble composé d'un plateau horizontal posé sur un ou plusieurs pieds.
- Un « homme » est un être humain de sexe masculin.

Exemples de concepts primitifs :

- Une « anémone » est une plantes herbacée dont plusieurs espèces sont cultivées pour leurs fleurs décoratives. *[À noter que cette propriété ne définit pas de façon suffisante une anémone car il existe d'autres plantes herbacées cultivées pour leurs fleurs]*
- Un « gardon » est un poisson d'eau douce. [*même remarque !*].
- Une « ruisseau » est un petit cours d'eau peu profond. [*Toujours même remarque : une rivière ou un torrent peuvent répondre à cette définition.*]

Notes :

- Les concepts dits « naturels », c'est-à-dire faisant référence à des objets non conçus par l'homme, sont généralement primitifs.
- La littérature fait état d'une proportion de 80% de concepts primitifs dans les ontologies déjà conçues. Parmi ces concepts primitifs figurent des « faux primitifs », c'est-à-dire des concepts pour lesquels une CNS peut être exprimée en français mais par dans le langage formel utilisé pour spécifier l'ontologie. Par exemple, en DefOnto, faute de disposer de la négation, il n'est pas possible de représenter le concept `APATRIDE` (une personne sans nationalité) en tant que concept défini.

Exercice 2

Dans cet exercice, on reprend la taxinomie de documents établie dans l'exercice 3 de la session précédente, pour élaborer une ontologie en DefOnto. À cette fin, on considère les notions suivantes pour les concepts :

- Un « document électronique » est un document ayant un support électronique. Tout document électronique a un format.
- Une « page Web » est un document électronique publié sur le Web.

- Une « page personnelle » est une page Web présentant une personne.
- Un « document pdf » est un document au format pdf.
- Un « document pédagogique » est un document exploité pour une activité pédagogique.
- Un « support de cours » est un document pédagogique accompagnant un cours.

Note : en DefOnto, la racine des concepts est l'entité *Object* et la racine des relations, l'entité *objectCouple*.

Conseil : le concept `DOCUMENT` et les autres concepts intervenant dans les différentes notions (ex : `ACTIVITÉ PÉDAGOGIQUE`, `PERSONNE`) pourront spécialiser directement le concept *Object*.

Solution de l'exercice 2

Tout d'abord, les concepts :

DefGenConcept Document
  IsA Object

DefGenConcept DocumentElectronique
  IsA Document (MI)support "électronique"
  ObjectsProperties
  (AE)format String

DefGenConcept PageWeb
  IsA DocumentElectronique (MI)publiéSur Web

DefIndConcept Web
  IsA Document

DefGenConcept Personne
  IsA Object

DefGenConcept PagePersonnelle
  IsA PageWeb (ME)présente Personne

DefGenConcept DocumentPdf
  IsA DocumentElectronique (MI)format "pdf"

DefGenConcept ActivitéPédagogique
  IsA Object

DefGenConcept DocumentPédagogique
  IsA Document (ME)exploitéPour ActivitéPédagogique

DefGenConcept Cours
  IsA ActivitéPédagogique

DefGenConcept SupportDeCours

IsA DocumentPédagogique (ME)accompagne Cours

Ensuite, les relations :

DefRelation support
IsA objectCouple
RelationProperties
domain Document
range String

DefRelation format
IsA objectCouple
RelationProperties
domain Document
range String

DefRelation publiéSur
IsA objectCouple
RelationProperties
domain Document
range Document

DefRelation présente
IsA objectCouple
RelationProperties
domain Document
range Personne

DefRelation exploitéPour
IsA objectCouple
RelationProperties
domain Document
range Object

DefRelation accompagne
IsA objectCouple
RelationProperties
domain Document
range Object