

Éléments sur La conception d'ontologies

Amedeo Napoli

LORIA – UMR 7503 – BP 239, 54506 Vandœuvre-lès-Nancy

Email : `Amedeo.Napoli@loria.fr`

La problématique de la conception d'une ontologie des UCD en OWL

ACI MDA

Strasbourg — 3 février 2005

-
-
-

Une introduction aux problèmes de représentation

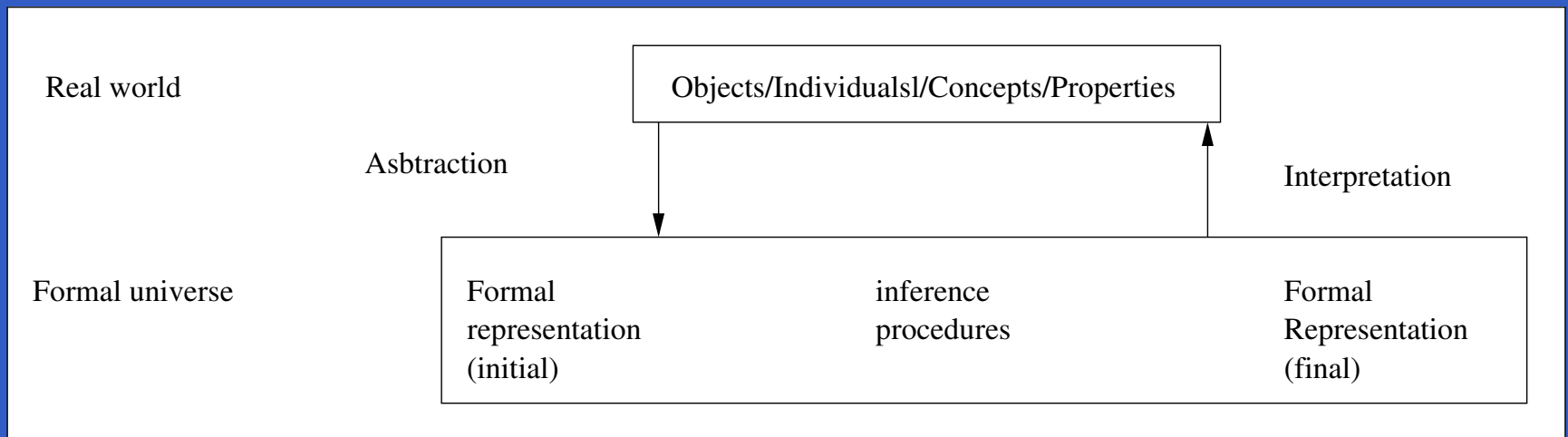
Une introduction aux problèmes de représentation

Données / Informations / Connaissances

- **Données** non interprétées, brutes :
! E ...- - -...
- **Information** = donnée + sens attaché à la donnée :
(!,note), (!,point d'exclamation), (E,mi), (...- - -...,SOS)
- **Connaissance** = information + expertise attaché à l'information pour engendrer des actions :
lire (!,note) et jouer la note indiquée.
lire (!,point d'exclamation) et interpréter la phrase associée.
(E,mi) et jouer la gamme de mi associée.
recevoir (...- - -...,SOS) : si un signal d'alerte est émis, envoyer des secours.

Une introduction aux problèmes de représentation

Architecture d'un système de représentation des connaissances



Une introduction aux problèmes de représentation

Représenter et partager des connaissances

- **Modéliser, représenter et organiser** les connaissances dans des **ontologies**.
- **Établir** une terminologie commune entre les agents — logiciels et humains — pour **partager** des concepts sur un plan syntaxique et sémantique.

-
-
-

La notion de système formel

La notion de système formel

Définition d'un système formel

Un **système formel** peut se voir comme un prototype de système symbolique et se compose de :

- un **vocabulaire** — ou **alphabet** — de symboles,
- un procédé de **construction** de formules,
- un ensemble d'**axiomes**,
- un ensemble de **règles d'inférences**, dites encore règles de **déduction** ou de **dérivation**.

La notion de système formel

La notion de formule bien formée

- Une formule d'un système formel est dite **formule bien formée** (fbf) si c'est un axiome, ou bien si elle est obtenue par application d'une règle d'inférence à partir d'une formule bien formée.
- Un problème général qui se pose dans le cadre d'un système formel est celui de savoir si une formule est une formule bien formée : dans ce cas, il s'agit de rechercher l'enchaînement de formules qui conduit d'un axiome à la formule reconnue.

La notion de système formel

Un exemple de système formel

- Alphabet : $\mathcal{A} = \{p, e, u\}$
- Procédé de construction de formules : concaténation.
- Un axiome : $upueuu$.
- **R1** : si AeB est une fbf alors $uAeBu$ est une fbf.
R2 : si AeB est une fbf alors $AueuB$ est une fbf.
- **Problèmes à résoudre** : $uupuueuuuu$, $uupuueuuuu$, $upupuueuuu$ sont des fbf ?

La notion de système formel

La définition d'une interprétation

- Un **domaine d'interprétation** : $\Delta = \mathbb{N}$.
- Une correspondance entre les symboles (du SF) et Δ :
 $u \longrightarrow 1$: u correspond à l'entier 1.
 $e \longrightarrow =$: e correspond à l'égalité des entiers.
 $p \longrightarrow +$: p correspond à l'addition des entiers.
- $u p u u e u u u u$ s'interprète alors comme l'addition d'entiers
 $1 1 + 1 1 = 1 1 1 1$.
- Associer une interprétation sur la base d'un domaine d'interprétation à une formule c'est lui associer une **sémantique**.

-
-
-

Une introduction aux ontologies

Une introduction aux ontologies

- En représentation des connaissances, le terme **ontologie** fait référence à un module constitué d'un **vocabulaire** spécifique utilisé pour décrire un **modèle** d'un domaine particulier du monde réel, vocabulaire auquel est associé un ensemble d'hypothèses — ou assertions — sur le sens qui doit être attaché aux éléments de l'ontologie.
- Le vocabulaire et les assertions qui composent l'ontologie sont censés apporter une aide aux « agents », agents logiciels ou agents humains, pour **résoudre** des problèmes ou **partager** des connaissances.

Une introduction aux ontologies

- Une ontologie est une spécification explicite d'une **conceptualisation**.
- Dans cette définition, « conceptualisation » correspond à « modèle abstrait » d'une partie du monde réel — sur lequel doit travailler le système considéré — qui se présente comme un ensemble de définitions de concepts munis de propriétés et de relations entre ces concepts.
- L'expression « spécification explicite » signifie que le modèle en question doit être décrit de façon non ambiguë dans un langage (formel) pour pouvoir être manipulé par un agent logiciel aussi bien que par un agent humain.
- L'utilisation effective des ontologies nécessite un langage de représentation des ontologies bien défini et des modules associés de raisonnement qui soient efficaces.

Une introduction aux ontologies

Une ontologie \mathcal{O} est un système formel qui est constitué de :

- Un ensemble S_C de concepts organisés en une hiérarchie \mathcal{H} où les concepts sont reliés par une relation de spécialisation (subsomption) \sqsubseteq (un ordre partiel), où $C_1 \sqsubseteq C_2$ signifie que C_1 est un sous-concept de C_2 .
- Un ensemble S_R de relations binaires qui spécifient des couples (D, R) de **domaines** et **co-domaines** dans S_C .
- Un ensemble A d'axiomes de l'ontologie.

Une introduction aux ontologies

Une **base de connaissances** peut être considérée comme un couple constitué d'une ontologie et d'une base d'**assertions** (**faits**) qui décrivent les individus et les assertions dans lesquelles ils sont engagés.

Éléments	Ontologie	Assertions
Expressions logiques	oui	oui
Théorie	générale	faits particuliers
Types des expressions	intensionnel	extensionnel
Construction	unique	multiple
Logiques CP0	proposition	instanciation
Logiques CP1	prédicat	instanciation
Logiques de descriptions	TBox	ABox

Éléments sur l'ingénierie des ontologies

- **Initialisation et modélisation** : spécification des éléments et des besoins de l'ontologie, mise en œuvre d'un modèle du domaine.
- **Représentation et raffinement** : implantation du modèle (choix d'un langage de représentation) et raffinement du modèle en fonction de l'évolution de l'implantation.
- **Évaluation** : utilisation de l'ontologie dans des applications (mise en place d'environnements logiciels adaptés) et preuve du bien-fondé pratique de l'ontologie.
- **Maintenance** : les éléments constituant l'ontologie évoluent ce qui fait évoluer d'autant la spécification de l'ontologie ; ces modifications doivent être répercutées dans l'ontologie qui doit être mise à jour (garantir la cohérence et la compatibilité).

-
-
-

Raisonnement et ontologies

Raisonnement et ontologies

- Le **raisonnement** est important pour assurer la qualité d'une ontologie et pouvoir l'exploiter convenablement.
- Il peut être employé à différents stades du développement de l'ontologie (cycle de vie).

Raisonnement et ontologies

- **Conception** : tester si les concepts sont non contradictoires et faire émerger les implications implicites.
- Par exemple, faire émerger une **hiérarchie des concepts** (ordre partiel des concepts nommées) sur la base d'une relation de subsomption.
- Retrouver qu'un concept est une **spécialisation** d'un autre concept, ou les concepts qui sont **synonymes**, peut être utilisé lors de la phase de conception, pour tester la cohérence des définitions de concepts.

Raisonnement et ontologies

- Le raisonnement peut aussi être utilisé lors du déploiement de l'ontologie, pour déterminer la **satisfiabilité** (cohérence) des faits utilisés ou inférer des relations entre instances ou concepts.
- Par exemple, lors d'une recherche d'information, comme des éléments contenus dans une page Web annotée avec les concepts d'une ontologie, il peut être intéressant de considérer les concepts de l'ontologie qui s'apparient avec les concepts de la page, ou encore des concepts plus généraux ou plus spécifiques, selon le type de résultats que l'on veut obtenir.

-
-
-

Les logiques de descriptions : une brève introduction

Les logiques de descriptions : une brève introduction

- Les éléments de connaissance du monde réel sont représentés par des **descriptions** qui peuvent être des **concepts**, des **rôles** et des **individus**.
- Les **logiques de descriptions** sont des logiques : les descriptions ont une syntaxe et une sémantique : elles sont construites selon un procédé de construction de FBF et elles sont manipulées en accord avec la sémantique associée au FBF.

Les logiques de descriptions : une brève introduction

- Les éléments et les classes d'éléments du domaine sont décrits par des **descriptions de concepts** et qui se construisent à partir d'**atomes** en utilisant des **constructeurs**.
- Les atomes peuvent être des **concepts atomiques** — prédicats unaires — et des **rôles atomiques** — prédicats binaires.
- Les logiques de descriptions sont équipées d'une sémantique formelle propre à l'image de celle du CP1.

Les logiques de descriptions : une brève introduction

- Deux niveaux de connaissances peuvent être pris en compte : un niveau générique **conceptuel** ou **terminologique (Tbox)** et un niveau **factuel** ou niveau des **assertions (Abox)**.
- Le niveau conceptuel est celui de l'introduction des différents types de concepts et de rôles (« mémoire à long terme »).
- Le niveau des assertions est celui de l'introduction des individus et des faits dans lesquels ils sont impliqués (« mémoire à court terme »).
- Les concepts et les rôles peuvent être ordonnés (ordre partiel) par la relation de **subsumption** dans une hiérarchie des concepts et une hiérarchie des rôles respectivement.

Les logiques de descriptions : une brève introduction

- Aux DL est associé un ensemble de **procédures d'inférence** qui permettent d'inférer des connaissances implicites à partir des connaissances explicitement représentées.
- La **subsumption** permet de tester qu'un concept D est plus général qu'un concept C (relation entre sous et super concepts).
- La **classification** permet de retrouver la position d'un concept C dans la hiérarchie des concepts.
- La **satisfiabilité** des concepts permet de tester qu'un concept C est satisfiable, c'est-à-dire qu'il — son **interprétation** — ne dénote pas toujours l'ensemble vide.

Les logiques de descriptions : une brève introduction

- La **satisfiabilité** d'une base de connaissances permet de tester que la connaissance représentée dans une base de concepts et d'assertions est satisfiable.
- La **dérivation** ou **déduction** permet de retrouver les faits qui peuvent être déduits des connaissances représentées.
- L'**instanciation** permet de retrouver les concepts dont un individu peut être une instance ou, de façon duale, les instances d'un concept donné.

Les logiques de descriptions : une brève introduction

$$\mathcal{AL} = \{\top, \perp, \neg A, C \sqcap D, \forall r.C, \exists r\}$$

- Négation de concepts définis: $\mathcal{ALC} = \mathcal{AL} \cup \{\neg C\}$
- Disjonction de concepts: $\mathcal{ALU} = \mathcal{AL} \cup \{C \sqcup D\}$
- Quantification existentielle typée : $\mathcal{ALE} = \mathcal{AL} \cup \{\exists r.C\}$
- Restrictions de cardinalités : $\mathcal{ALN} = \mathcal{AL} \cup \{(\geq n r), (\leq n r)\}$
 $\mathcal{ALQ} = \mathcal{AL} \cup \{(\geq n r.C), (\leq n r.C)\}$
- $\mathcal{ALO} = \mathcal{AL} \cup \{\text{types énumérés (listes d'individus)}\}$
- Domaines et co-domaines de rôles, rôle inverse, hiérarchie de rôles, composition de rôles, ...

-
-
-

Vers une ontologie des UCD

Vers une ontologie des UCD

- Le stage de Sébastien Derivaux et de Rim Al Hulou (juillet et août 2004) : proposer les premiers éléments de ce que peut être une ontologie des UCD.
- Construction de l'ontologie avec l'éditeur d'ontologies Protégé-2000 et OWL, le langage de représentation d'ontologies du Web sémantique (logique de descriptions *SHIQ*).

Vers une ontologie des UCD

Les problèmes abordés.

- Construction d'une ontologie explicitant les règles syntaxiques de fabrication d'un UCD : un UCD est une formule bien formée dans un système formel.
- Construction d'une ontologie pour la classification des UCD « en fonction de leur sémantique » à l'aide d'un formulaire : les UCD comme des concepts définis autorisant la classification de nouveaux concepts (décrivant de nouveaux UCD).

Vers une ontologie des UCD

- Expliciter les règles de construction syntaxique des UCD et les représenter dans l'ontologie : un concept de l'ontologie est dénoté par une formule bien formée.
- Les concepts de l'ontologie : concepts primitifs, par exemple `primary`, `secondary` selon la position du concept dans une formule, ...
- Deux propriétés (rôles) primitifs : `allow` et `need`, qui ont pour domaine et co-domaine un UCD et qui doivent s'interpréter comme un mot « peut » ou « doit » être suivi d'un autre mot d'un certain type.
- Exemple : `pos.eq.ra` ; `stat.max` ; `meta.main` est un mot composé valide.

Vers une ontologie des UCD

Travaux initiés.

- Importation des listes d'UCD en format CDS pour transformation en format OWL.
- Problème : mettre en place des **concepts définis** qui sont les seuls à autoriser une classification dans l'ontologie (quels sont ces concepts et que doivent-ils représenter ?).
- Une classification par formulaire : classer des concepts à partir d'une description ; c'est plus la le rôle attendu d'une ontologie.

Vers une ontologie des UCD

Quelques problèmes.

- La prise en main des outils du Web sémantique : les logiques de descriptions, Protégé-2000, OWL, et les contraintes liées à ces systèmes.
- Quelques limitation sensibles : la gestion des listes en OWL, les traitements numériques de tous ordres (conversion d'unités ...), ...
- ...

Vers une ontologie des UCD

Conclusion provisoire.

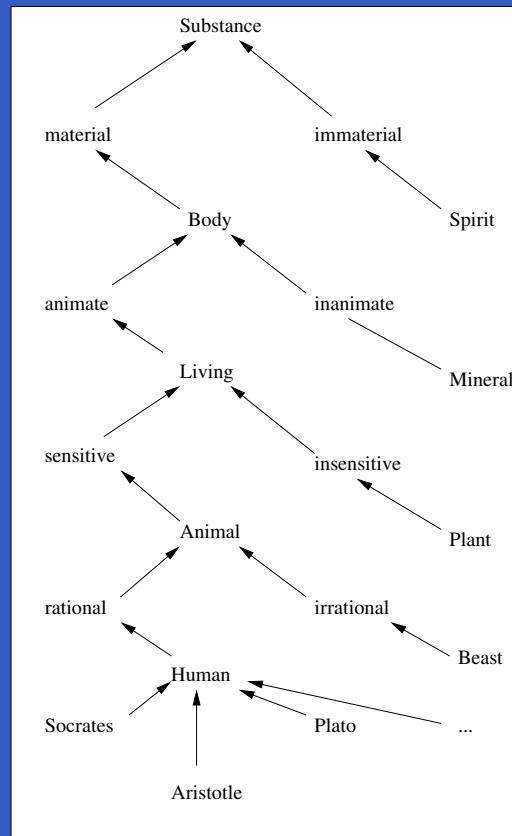
- Une première expérience importante pour la suite et des éléments sur lesquels s'appuyer, mais tout est à refaire (ou presque) : toutefois, cette fois on ne part plus de rien, que ce soit au CDS ou au LORIA.
- Un stage de DEA va débiter sur le sujet à partir de février 2005 : l'accent va être mis sur une ontologie des UCD comme un ensemble de concepts définis décrivant les UCD.

-
-
-

Annexes

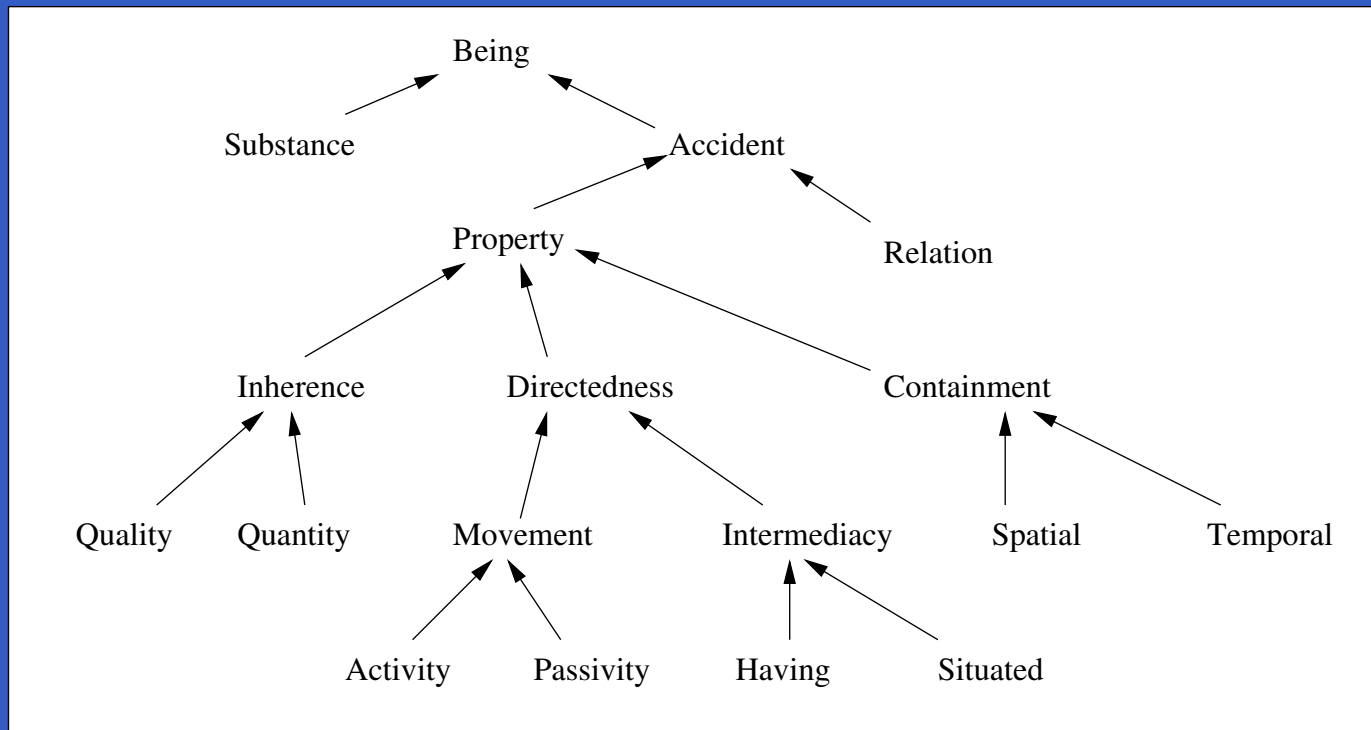
Annexes

L'ontologie de Porphyre



Annexes

L'ontologie d'Aristote



Annexes

- “A happy man is a man that is married to a doctor and has at least five children, all of whom are professors”
 $\text{Human} \sqcap \neg \text{Female} \sqcap \exists \text{married.Doctor} \sqcap \geq 5 \text{ hasChild} \sqcap \forall \text{hasChild.Professor}$
- $\text{HappyMan} \sqsubseteq \exists \text{hasChild.Professor}$
- A terminological axiom :
 $\exists \text{hasChild.Human} \sqsubseteq \text{Human}$, mentioning that only humans can have human children.
- Instantiation: $\text{HappyMan}(\text{Paolo})$ states that Paolo is an instance of the concept HappyMan .
- $\text{hasChild}(\text{Paolo}, \text{Maria})$ states that the pair $(\text{Paolo}, \text{Maria})$ is an instance of the role hasChild , i.e. Maria is one of the children of Paolo.

Annexes

- Since instances of `HappyMan` have at least five children, all of whom are professors, the *instance* algorithm determines instance relationships, i.e. the individual j is an instance of the concept description C iff j is always interpreted as an element of C : `Maria` is an instance of `Professor`.
- The *consistency* algorithm determines whether a knowledge base (assertions plus terminological axioms) is non-contradictory.
- For example, if we add the assertion $\neg\text{Professor}(\text{Maria})$ to the two above assertions, the resulting knowledge base becomes inconsistent.