

# Des réseaux sémantiques au Web sémantique

Nhan LE THANH - Equipe KEWI  
(Knowledge Engineering and Web Intelligent)  
Pôle GLC - Laboratoire de I3S

## Plan du séminaire

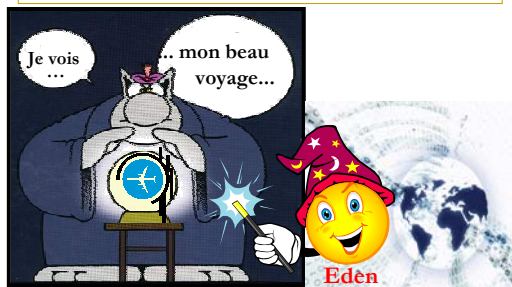
- Web sémantique : rêve et défis
- Aspects théoriques
  - Langages du web (balisage)
  - Graphes conceptuels
  - Logiques de descriptions
- Aspects pratiques
  - Annotation : Langages
  - Ontologies : la modélisation
  - Web services : la mise en oeuvre
- Tendances et thèmes de recherche
  - Distribution et fédération
  - Web social et centre d'usage
  - Thèmes de recherche
- Et demain ...  
... c'est un autre jour ...

# Rêves et défis

## Web sémantique : rêve et défi

« ... Un jour de l'automne 2020 :

- Adam et Ève décident d'organiser leur voyage de noce en l'Amérique latin.
- Adam est gourmet et Ève est une fan de la nature. Ils ont donc décidé que leur voyage sera un séjour géographique et gastronomique ...



- Adam interroge Eden, son moteur de recherche personnel et dans quelques minutes Eden leur propose quelques propositions illustrées par une simulation multimédia de leur voyage.
- Après le choix du couple, l'Eden s'occupe de la mise en place du voyage le plus adapté ... »

## Web sémantique : rêve et défi

### ■ Défis à relever pour Eden



## Web sémantique : rêve et défi

**Aujourd'hui** : le Web est exploité par des *personnes* qui recherchent des informations via un moteur de recherche et qui exploitent elles-mêmes le(s) résultat(s).



**Demain** : le Web sera exploité en priorité par des *machines* qui traiteront elles-mêmes les questions posées par des personnes, et qui délivreront les résultats à ces personnes.

## Web sémantique : rêve et défi

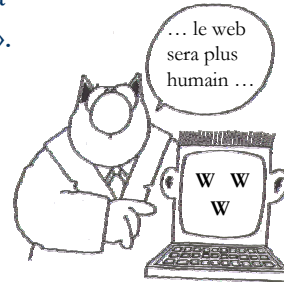
- Vers le Web sémantique
  - Le Web va devenir un *espace d'échanges d'informations entre machines*, en permettant l'accès à un très grand volume d'informations, et en fournissant les moyens de gérer ces informations.
  
  - Une machine peut appréhender le volume des informations disponibles sur le Web, et donc fournir une aide plus conséquente aux personnes, à condition qu'on la dote d'une certaine « intelligence ».

## Web sémantique : rêve et défi

- **Aujourd'hui dans le Web il existe ...**
  - des moteurs de recherche et une indexation des pages,
  - des procédures d'extraction d'information,
  - des procédures d'analyse de l'information,
- mais**
  - il n' y a pas encore (de manière massive) la possibilité d'exploiter des *connaissances* pour une gestion plus intelligente des informations : recherches par le contenu des documents, raisonnements, résolution de problèmes, etc.

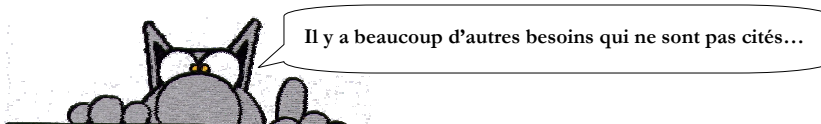
## Web sémantique : rêve et défi

- Demain dans le Web, on attend :
  - Un Web *qualitatif* et *quantitatif* : « il y a tout ce qu'on cherche sur le Web ... ».
  - Cet espace d'échange peut être *personnalisé*, avec des espaces privés, publics, sécurisés, des procédures d'authentification, une signature digitale, ...
  - La *preuve du bien fondé* d'un résultat doit pouvoir être donnée par la machine.



## Web sémantique : rêve et défi

- **Quels sont les besoins ?**
  - Une *identification* et un *accès* aux ressources du Web de façon simple, robuste, et efficace (URI).
  - Des langages pour décrire le *contenu* des documents et ressources, pour associer une *sémantique* à ces descriptions, et des moteurs d'inférences pour les manipuler ...
  - Des *ressources* de plusieurs types : *ontologies* (bases de connaissances), bases de données, éléments de code, etc.



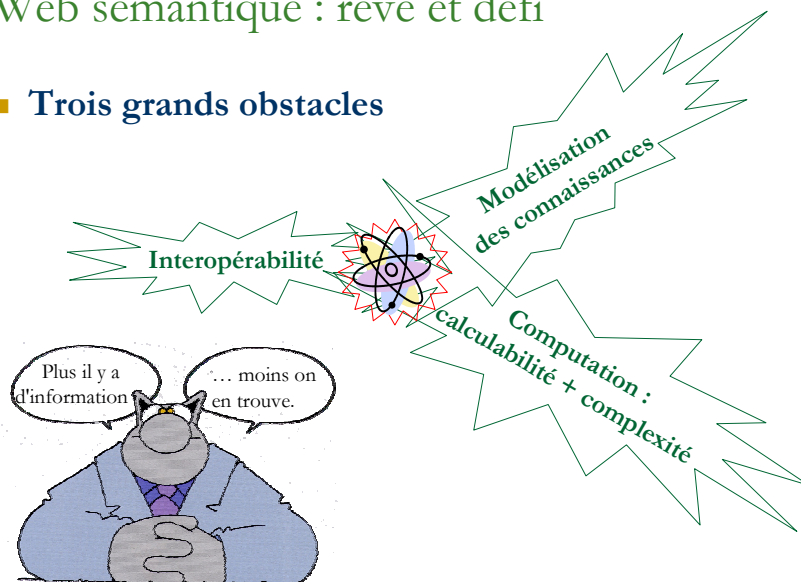
## Web sémantique : rêve et défi

### ■ Et les technologies en jeu

- ❑ La technologie du Web : URI, XML, RDF(S), ...
- ❑ La représentation des connaissances : ontologies, logique, preuve de théorèmes, ...
- ❑ La gestion de bases de données : requêtes, stockage, données semi-structurées, ...
- ❑ La recherche et extraction d'informations, fouille de données.
- ❑ Le traitement et analyse de l'information.
- ❑ L'interopérabilité et coopération.

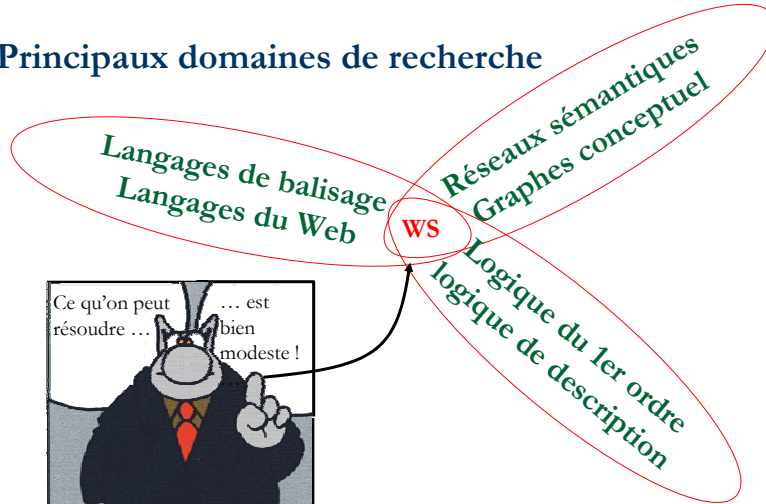
## Web sémantique : rêve et défi

### ■ Trois grands obstacles



## Web sémantique : rêve et défi

### ■ Principaux domaines de recherche



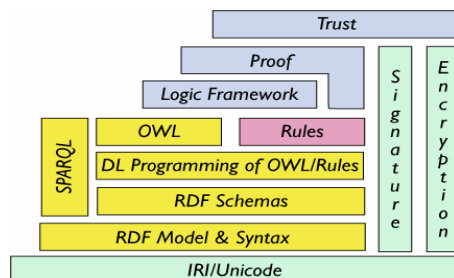
## Aspects théoriques

## Aspects théoriques : langages du web

- Langage de balisage
  - Sgml (Standard Generalized Markup Language)
    - Norme publiée en 1986 ISO 8879:1986
    - Projet européen FORMEX (Formalized Exchange of Electronic Publications) – code ASCII
  - Langages de balise pour le web
    - html : une application de sgml pour édition des pages web
    - xml : langage générique pour le web issu de sgml (unicode à la place ASCII, balises bien formées, espace de nom, ...)
- Description, échange, du contenu documentaire multimédia compréhensibles par homme mais l'absence de la compréhension par la machine

## Aspects théoriques : langages du web

- Les 4 principaux langages du Web sémantique
  - **RDF**: un modèle de triplets pour décrire et connecter des ressources anonymes ou identifiées par un URI (sujet, prédicat, objet) / graphe orienté étiqueté
  - **SPARQL**: un langage de requête sur les graphes RDF
  - **RDFS** est un langage de déclarations et descriptions légères; typage des ressources et de leurs relations subClassOf, subPropertyOf, range, domain
  - **OWL**: 3 couches d'extension de l'expressivité (logique)
  - Un modèle en couche dans une direction d'extension; RDF sans RDFS, RDFS sans OWL, ...



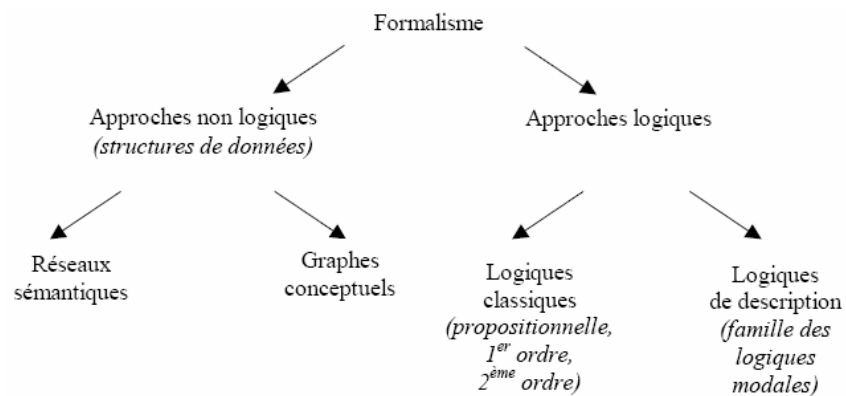


## Aspects théoriques : Représentation de connaissances

- Objectif de la représentation des connaissances
  - le support préalable aux traitements ultérieurs sur les connaissances
  - En allant du moins structuré vers le plus structuré, les opérations les plus usuelles sont
    - Organiser, classer,...
    - Chercher, extraire,...
    - Déduire, établir des contradictions, réviser,...
  - la représentation des connaissances explicites dans un formalisme vise la recherche de connaissances implicites mais inhérentes aux faits de base.

## Aspects théoriques : Représentation de connaissances

- Différents formalismes (selon Serge Haddad)

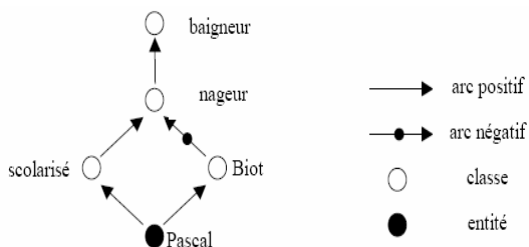


## Aspects théoriques : Réseaux sémantiques

- Définition : un réseau sémantique est un graphe orienté acyclique dont :
  - les **noeuds** sont soit des **classes** d'entités, soit des **entités** spécifiques,
  - les **arcs** sont de deux types arcs **positifs** et arcs **négatifs**.de plus une entité n'est jamais à l'extrémité d'un arc.
- Interprétation :
  - Un **arc positif** reliant un noeud C à un noeud D signifie : (un) C est généralement un D.
  - Un **arc négatif** reliant un noeud C à un noeud D signifie : (un) C n'est généralement pas un D.
  - L'article indéfini est présent dans le cas où C est une classe.
- la finalité de ce type de modèle est de *classer* entités et classes en supportant des *exceptions*

## Aspects théoriques : Réseaux sémantiques

### ■ Exemple :



Cette classification est bien entendu discutable et elle a pour seul but d'illustrer le modèle.

- Un nageur est généralement un baigneur (il faut bien aller dans l'eau pour nager).
- Un élève scolarisé sait généralement nager puisqu'il a suivi une ou deux années d'initiation à la natation.
- Un biotoir ne sait généralement pas nager
- Enfin Pascal est un touareg qui a été scolarisé.

## Aspects théoriques : Réseaux sémantiques

### ■ Requête sur les réseaux :

La question évidente que l'on désire poser est l'existence d'une relation de **spécialisation** que peut entretenir une classe C envers les autres classes D du réseau (ou une entité envers les autres classes) avec trois réponses possibles :

- 1. (Un) C est généralement un D.
- 2. (Un) C n'est généralement pas un D.
- 3. Le réseau ne permet pas de conclure.

Ce type de question possède donc un unique paramètre C.

## Aspects théoriques : Réseaux sémantiques

### ■ Problème d'ambiguïté :

- Nous construisons les successeurs positifs à partir de C et nous calculons ensuite les successeurs négatifs.

- Examinons le résultat de cet algorithme sur l'exemple appliqué à Pascal. Nous concluons que :

- 1- Pierre est un élève scolarisé, biotoir, nageur et baigneur.
- 2- Pierre n'est pas un nageur.

- Autrement dit, l'algorithme produit deux informations contradictoires sur le statut de Pierre en tant que nageur et de plus il déduit de manière hasardeuse que Pierre est baigneur en raison de l'ambiguïté précédente. Il s'agit de corriger l'algorithme pour intégrer le concept d'ambiguïté.

// suivant positif: un suivant par un arc positif  
Pile=∅;

Marquer les suivants positifs de C;  
Empiler les suivants positifs de C;

**Tant que** la pile est non vide **faire**  
Dépiler un noeud D;

**Pour tout** suivant positif E de D non marqué  
**faire** E est un successeur positif de C;

Marquer E;  
Empiler E;

**Fin pour**  
**Fin tant que**

**Mais si on rencontre un arc négatif :**

// suivant négatif: un suivant par un arc négatif

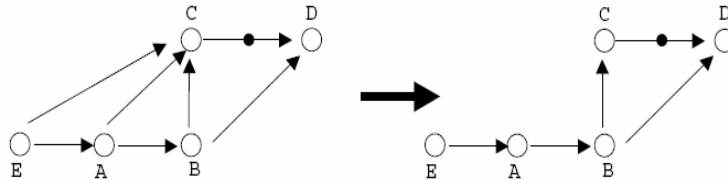
**Pour tout** successeur positif E de C **faire**

**Pour tout** suivant négatif D de E **faire**

D est un successeur négatif de C à D;

## Aspects théoriques : Réseaux sémantiques

### ■ Problème de redondance



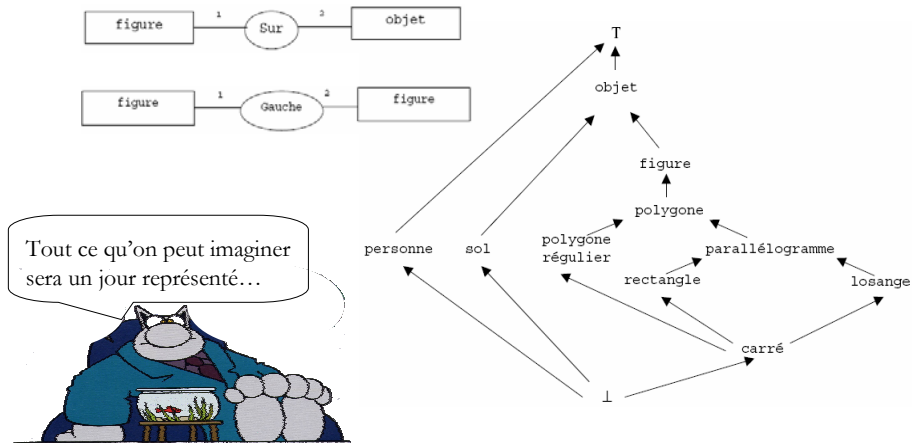
- Sur la partie gauche de la figure, les liens A-C et E-C sont redondants car ils se déduisent des chemins A-B-C et E-A-B-C. Cependant cette redondance n'est pas inoffensive car s'agissant de la relation de spécialisation qui lie E à D, on conclut à une ambiguïté. En effet, aucun des chemins E-C-D et A-B-D n'est plus spécifique
- Si on supprime maintenant les liens redondants (partie droite), l'ambiguïté disparaît, le chemin E-A-B-D est plus spécifique que le chemin E-A-B-C-D. E est donc généralement un D.

## Aspects théoriques : Graphes Conceptuels

- Formalisme introduit par Sowa (86-02) et Chein (92)
  - Un des apports significatifs du modèle des graphes conceptuels est la **séparation** entre la **connaissance générale** d'un domaine (concepts, relations, etc.) qui sera représentée par un **support** et la description d'une **situation particulière** qui sera modélisée par un **S-graphe**.
  - Un support comprend
    - un ensemble des **concepts organisé en treillis** avec la relation binaires spécialisation/généralisation entre concepts
    - un ensemble des **relations binaires** entre concepts
    - un ensemble (éventuellement dénombrable) de **marqueurs**. Chaque marqueur désigne un objet individuel excepté le marqueur \* qui désigne n'importe quel objet.

## Aspects théoriques : Graphes Conceptuels

- Un support : sous forme d'un treillis de concepts



## Aspects théoriques : Graphes Conceptuels

- Les marqueur

- Chaque marqueur désigne un objet individuel excepté le marqueur \* qui désigne n'importe quel objet
- Il y a implicitement une relation d'ordre partiel entre les marqueurs telle que  $m < *$  pour  $m$  marqueur individuel et telle que deux marqueurs ne sont jamais en relation par cet ordre.
- Il faut indiquer à quels concepts peuvent être associés les marqueurs. On note alors  $\text{conf}(c,m)$  si  $m$  est associé au concept  $c$  :
  - Si  $\text{conf}(c,m)$  et si  $c'$  généralise  $c$  alors  $\text{conf}(c',m)$
  - Si  $m$  est un marqueur individuel, si  $\text{conf}(c,m)$  et si  $\text{conf}(c',m)$  alors  $\text{conf}(c \wedge c',m)$  où  $c \wedge c'$  désigne le plus grand concept spécialisant  $c$  et  $c'$
  - $\text{conf}(c,*)$  est vérifié excepté si  $c$  est le concept absurde.

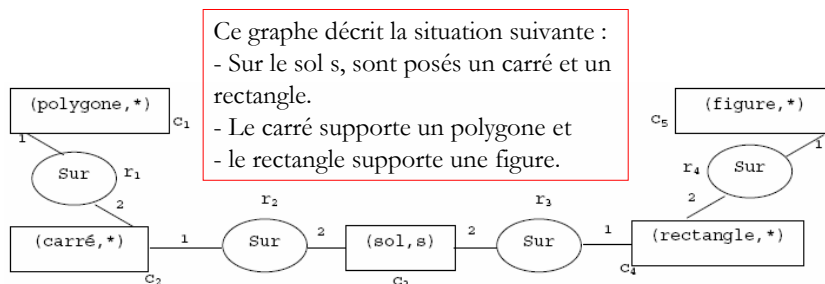
## Aspects théoriques : Graphes Conceptuels

### ■ S-graphe

- Un S-graphe est un **graphe biparti** dont les deux types de noeuds sont des **instances de relation** et des **instances de concept**.
  - Un noeud **instance de relation** est étiqueté par une relation et
  - un noeud **instance de concept** est étiqueté par un couple (c,m) où c est un concept et m un marqueur et où  $\text{conf}(c,m)$  est vérifié.
  - L'ensemble des couples  $\{(c,m)\}$  est partiellement ordonné par l'ordre produit sur les deux composantes
  - Enfin un noeud **instance d'une relation n-aire** a exactement **n noeuds concepts voisins** (dont les arcs sont étiquetés de 1 à n) tels que si c est le concept associé au ième paramètre de la relation et (c',m) étiquette le ième voisin du noeud instance de relation alors  $c' \leq c$ .

## Aspects théoriques : Graphes Conceptuels

### ■ Exemple un S-graphe



Cette même situation peut être décrite par une formule de logique de premier ordre ne comportant que des quantificateurs existentiels et des conjonctions :

$$\exists x \exists y \exists z \exists t \text{ polygone}(x) \wedge \text{carré}(y) \wedge \text{sol}(s) \wedge \text{rectangle}(z) \wedge \text{figure}(t) \wedge \text{sur}(x,y) \wedge \text{sur}(y,s) \wedge \text{sur}(z,s) \wedge \text{sur}(t,z)$$

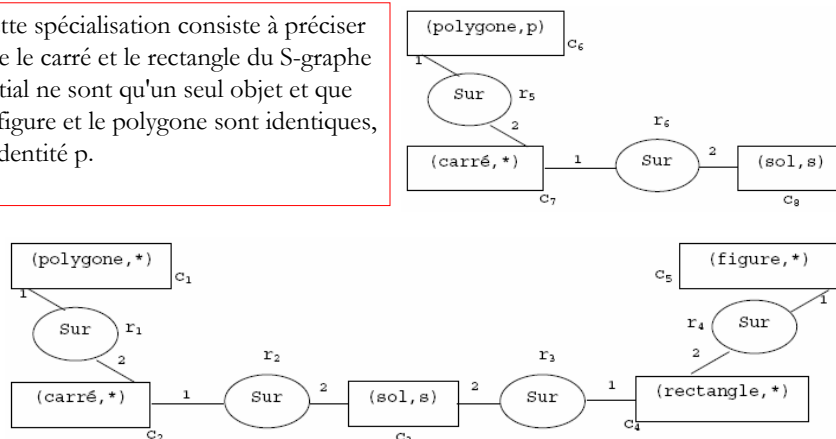
## Aspects théoriques : Graphes Conceptuels

- Spécialisation/généralisation : opérations fondamentales et duales
  - Soit H et G deux S-graphes, H est une **spécialisation** de G si et seulement si il existe **f**, une application de l'ensemble des **noeuds concepts** de G vers ceux de H, et **g**, une application de l'ensemble des **noeuds relations** de G vers ceux de H, telles que :
    - Le couple associé (c,m) associé au noeud f(nc), nc noeud concept, spécialise celui de nc.
    - La relation associée au noeud f(nr), nr noeud relation, est identique à celle de nr.
    - Pour chaque arc étiqueté i reliant nr à nc, il y a un arc étiqueté i reliant f(nr) à f(nc).
  - La généralisation est la relation inverse de la spécialisation.

## Aspects théoriques : Graphes Conceptuels

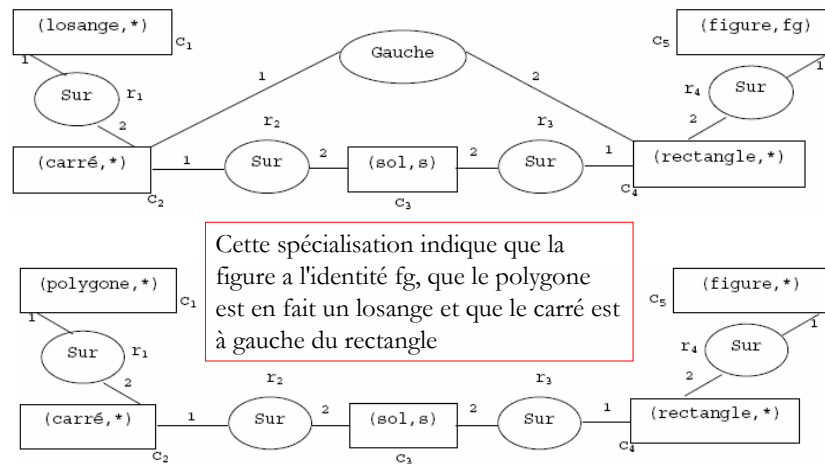
- Exemple de bande d'images (Serge Haddad)

Cette spécialisation consiste à préciser que le carré et le rectangle du S-graphe initial ne sont qu'un seul objet et que la figure et le polygone sont identiques, d'identité p.



## Aspects théoriques : Graphes Conceptuels

### ■ Exemple de bande d'images (Serge Haddad)



## Aspects théoriques : Graphes Conceptuels

### ■ Complexité algorithmique

- On ajoute aux axiomes de la logique du premier ordre, les formules suivantes :
  - " $\forall c(x) \Rightarrow c'(x)$  pour tout concept  $c$  spécialisant  $c'$  «
- La puissance expressive du modèle : On démontre que  $H$  spécialise  $G$  si et seulement si la formule associée à  $H$  implique la formule associée à  $G$ .
- De nombreux problèmes simples (tels que le test de spécialisation entre deux S-graphes) sont NP-complets et n'admettent des algorithmes polynomiaux que dans des cas restreints



## Aspects théoriques : Logique de description

### ■ Historique

- ❑ Une famille de formalismes de représentation de connaissances basés sur la logique descendants de réseaux sémantique and KL-ONE
- ❑ Description du domaine d'application en terme de **concepts** (classes), **rôles** (propriétés, relations) and **individus** (objets)
- ❑ Caractéristiques :
  - **Sémantique formelle** (modèle théorique)
    - ❑ **fragments décidables de la LPO** (sous ensemble de classe  $C_2$ )
    - ❑ Lien étroit aux Logiques Modales Propositionnelles & Logiques Dynamiques
  - **Production des services d'inférence**
    - ❑ Procédure de décision pour des problèmes clés (satisfiabilité, subsumption, etc)
    - ❑ Systèmes implantés hautement optimisés

## Aspects théoriques : Logique de description

### ■ Éléments basiques

- ❑ Les **Concepts** sont équivalents aux prédicats unaires
  - En général, les concepts sont équivalents aux formules avec une variable libre
- ❑ Les **Rôles** sont équivalents aux prédicats binaires
  - En général, les rôles sont équivalents aux formules avec deux variables libres
- ❑ Les **Individus** sont équivalents aux constantes
- ❑ Les **Constructeurs** sont restreints pour que :
  - Le langage reste décidable et, si possible, avec une complexité faible
  - L'utilisation des variables n'est pas explicite
  - Les caractéristiques telles que calcul peuvent être exprimées succinctement

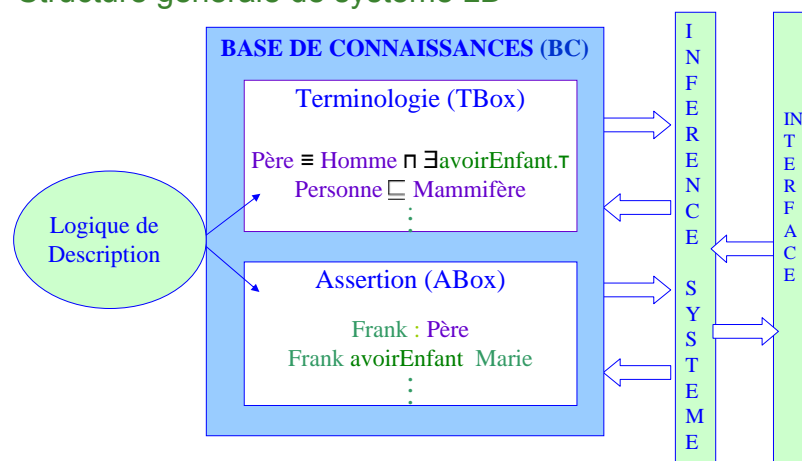
## Aspects théoriques : Logique de description

### ■ Grands principes

- Les éléments du monde réel sont représentés par des *concepts*, des *rôles* et des *individus*.
- Adéquation syntaxe – sémantique
  - les concepts et rôles possèdent une *description structurée* à laquelle est associée une *sémantique*
  - les manipulations opérées sur les concepts et les rôles sont réalisées en accord avec la sémantique
- Deux types de connaissances sont prises en compte :
  - les concepts avec leurs composants, et
  - les *faits* ou *assertions*, où interviennent les concepts et les instances
  - La relation de subsomption organise concepts et rôles en hiérarchies
- Raisonnement : *classification et instanciation*

## Aspects théoriques : Logique de description

### Structure générale de système LD



## Aspects théoriques : Logique de description

### ■ Base de connaissances

- Une base de connaissances  $K$  est une paire  $\{T, A\}$  où
- $T$  est un ensemble d'axiomes "terminologiques" (TBox)
- $A$  est un ensemble d'axiomes "assertionnels" (ABox)
- Les axiomes de TBox sont sous forme :
  - $A, C \sqcup D, C \sqcap D, r \sqcup s, r \sqcap s$  et  $R^+ \sqcup R$
  - où  $A$  est un concept atomique,  $C, D$  sont des concepts,  $r, s$  sont des rôles, et  $R^+$  l'ensemble de rôles transitifs
- Les axiomes de Abox sont sous forme :
  - $x:D, \langle x, y \rangle : R$
  - où  $x, y$  sont des individus,  $D$  est un concept et  $R$  est un rôle

## Aspects théoriques : Logique de description

### ■ Exemple : syntaxe d'une LD

La syntaxe d'une logique de description			
$C \sqsubseteq A$	Subsompion de concepts	$\forall r.C$	Restriction universelle
$\perp$	Concept absurde	$\exists r.C$	Restriction existentielle
$\top$	Concept universel	$(\geq n \ r)$	Restriction supérieure de cardinalité
$C \sqcap D$	Conjonction de concepts	$(\leq n \ r)$	Restriction inférieure de cardinalité
$C \sqcup D$	Disjonction de concepts	$r \sqsubseteq a$	Subsompion de rôles
$\neg C$	Négation de concept	$r \sqcap s$	Composition de rôles

-  $C$  et  $D$  sont des expressions de concepts,  $r$  et  $s$  sont des expressions de rôles  
 -  $A$  est un concept primitif et  $a$  est un rôle primitif  
 -  $n$  est un entier non nul

## Aspects théoriques : Logique de description

### ■ Exemple : Base de connaissance

#### ■ Description en texte

- Un **Homme** est une **Personne**.
- Une **Femme** est une **Personne**.
- Aucune **Femme** n'est un **Homme** et vice-versa.
- Une **Equipe** est (définie comme) un **Ensemble** ayant au moins 2 **membres** qui sont tous des **Personnes**.
- Une **Petite-équipe** est (définie comme) une **Equipe** ayant au plus 5 **membres**.
- Une **Equipe-moderne** est (définie comme) une **Equipe** ayant au moins 4 **membres**, ayant au moins 1 **chef**, et dont tous les **chefs** sont des **Femmes**

#### ■ La base de connaissances DL

- Concepts primitifs : **Personne**, **Ensemble**
- Rôles primitifs : **membre**
- $Femme \sqsubseteq Personne$
- $Homme \sqsubseteq Personne \sqcap \neg Femme$
- $chef \sqsubseteq membre$
- $Equipe \sqsubseteq Ensemble \sqcap (\forall \text{membre } Personne) \sqcap (\geq 2 \text{ membre})$
- $Petite-équipe = Equipe \sqcap (\leq 5 \text{ membre})$
- $Equipe-moderne = Equipe \sqcap (\geq 4 \text{ membre}) \sqcap \exists \text{chef} \sqcap (\forall \text{chef. } Femme)$

## Aspects théoriques : Logique de description

### ■ Langage *FL<sub>E</sub>* avec 3 constructeurs (le plus simple)

Constructeur	Syntaxe	Exemple
Concept universel	$\top$	
Conjonction	$A \sqcap B$	$Personne \sqcap Jeune$
Quantification Universelle	$\forall r.C$	$\forall \text{enfant.Masculin}$
Quantification Existentielle	$\exists r. \top$	$\exists \text{enfant. } \top$

## Aspects théoriques : Logique de description

- Famille des langages  $\mathcal{AL}$

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

- $\mathcal{ALC} = \mathcal{AL} \cup \{\neg C\}$  (négation de concepts définis)  
(équivalent à la classe de logique modal  $K_{(m)}$ )
- $\mathcal{ALU} = \mathcal{AL} \cup \{C \sqcup D\}$  (disjonction de concepts)
- $\mathcal{ALE} = \mathcal{AL} \cup \{\exists r.C\}$  (quantification existentielle typée)
- $\mathcal{ALN} = \mathcal{AL} \cup \{\geq n r, \leq n r\}$  (cardinalité de rôles -  
remarque  $\exists r \equiv (\geq 1 r)$ )
- $\mathcal{ALR} = \mathcal{AL} \cup \{r_1 \sqcup r_2\}$  (conjonction de rôles)

## Aspects théoriques : Logique de description

### Famille des LD: *SHOIN* et *SHIQ* (OWL)

Sigle	Nom	Syntaxe	Exemple
$S$ ( $\mathcal{ALC}$ et $R^+$ )	Conjonction	$C \sqcap D$	personne $\sqcap$ jeune
	Disjonction	$C \sqcup D$	vieux $\sqcup$ jeune
	Négation	$\neg C$	$\neg$ (personne $\sqcap$ jeune)
	Quantificateur Universel	$\forall r.C$	$\forall$ enfant.male
	Quantificateur Existentiel	$\exists r.C$	$\exists$ enfant.male
	Rôle transitif ( $R^+$ )	$r.r$	père.père
$\mathcal{H}$	Rôle hiérarchique	$r \sqsubseteq s$	mère $\sqsubseteq$ parent
$\mathcal{O}$	Concepts nominaux	$\{x\}$	{Terre, Mars, Venus}
$\mathcal{I}$	Inverse de rôle	$r^{-}$	enfant $\Leftrightarrow$ parent
$\mathcal{N}$	Restriction de nombre	$\geq nr$	$\geq 2$ enfant
$\mathcal{Q}$	RN qualifiée	$\geq nr.C$	$\geq 2$ enfant.male

## Aspects théoriques : Logique de description

### ■ Sémantique de LD

- La sémantique de LD est définie par les **interprétations**
- Une interprétation  $\mathbf{I}$  est définie comme
  - $\mathbf{I} = (\Delta^{\mathbf{I}}, \mathfrak{I}^{\mathbf{I}})$ , où
  - $\Delta^{\mathbf{I}}$  est le **domaine** (un ensemble vide)
  - $\mathfrak{I}^{\mathbf{I}}$  est une **fonction d'interprétation** qui fait correspondre:
    - **Concept** (classe)  $A \Rightarrow$  Sous ensemble  $A^{\mathbf{I}}$  de  $\Delta^{\mathbf{I}}$
    - **Rôle** (propriété)  $r \Rightarrow$  relation binaire  $r^{\mathbf{I}}$  sur  $\Delta^{\mathbf{I}} \times \Delta^{\mathbf{I}}$
    - **Individus**  $i \Rightarrow i^{\mathbf{I}}$  élément de  $\Delta^{\mathbf{I}}$

## Aspects théoriques : Logique de description

### ■ Sémantique des expression LD

- La fonction d'interprétation  $\mathfrak{I}^{\mathbf{I}}$  sur les **expressions** de concept (ou de rôle) de la manière suivante :

$$(C \sqcap D)^{\mathbf{I}} = C^{\mathbf{I}} \cap D^{\mathbf{I}}$$

$$(C \sqcup D)^{\mathbf{I}} = C^{\mathbf{I}} \cup D^{\mathbf{I}}$$

$$(\neg C)^{\mathbf{I}} = \Delta^{\mathbf{I}} \setminus C^{\mathbf{I}}$$

$$\{x\}^{\mathbf{I}} = \{x^{\mathbf{I}}\}$$

$$(\exists R.C)^{\mathbf{I}} = \{x \mid \exists y. \langle x, y \rangle \in R^{\mathbf{I}} \wedge y \in C^{\mathbf{I}}\}$$

$$(\forall R.C)^{\mathbf{I}} = \{x \mid \forall y. \langle x, y \rangle \in R^{\mathbf{I}} \Rightarrow y \in C^{\mathbf{I}}\}$$

$$(\leq nR)^{\mathbf{I}} = \{x \mid \#\{y \mid \langle x, y \rangle \in R^{\mathbf{I}}\} \leq n\}$$

$$(\geq nR)^{\mathbf{I}} = \{x \mid \#\{y \mid \langle x, y \rangle \in R^{\mathbf{I}}\} \geq n\}$$

$$(R^-)^{\mathbf{I}} = \{(x, y) \mid (y, x) \in R^{\mathbf{I}}\}$$

## Aspects théoriques : Logique de description

- Exemple ABox – Interprétation sémantique :
  - Soit la base terminologique
  - $\Sigma = \{\text{enfant}(\text{pierre}, \text{marie}), (\forall \text{enfant}.\neg \text{Musicien})(\text{Pierre}), \text{femme}(\exists \text{enfant})(\text{Marie})\}$
  - La base  $\Sigma$  est satisfiable et l'interprétation  $\mathbf{I}$  ci-après est un modèle pour  $\Sigma$ 
    - $\Delta^{\mathbf{I}} = \{\text{Pierre}, \text{Marie}\}$
    - $\text{Pierre}^{\mathbf{I}} = \text{Pierre}$
    - $\text{Marie}^{\mathbf{I}} = \text{Marie}$
    - $\text{enfant}^{\mathbf{I}} = \{(\text{Pierre}, \text{Marie})\}$
    - $\text{femme}^{\mathbf{I}} = \{\text{Marie}\}$
    - $\text{Musicien}^{\mathbf{I}} = \emptyset$

## Aspects théoriques : Logique de description

### ■ Exemple de base de connaissance

#### ■ Niveau Terminologique

- Concepts primitifs : Cours, Professeur, Etudiants, DEA et Doctorat
- Rôles primitifs : enseigne et diplôme.
- Contraintes
  - Tout enseignant d'un cours est soit professeur, soit titulaire d'un diplôme de DEA.
  - $(\exists \text{enseigne.Cours}) \sqsubseteq (\text{Etudiant} \sqcap (\exists \text{diplôme.DEA})) \sqcup \text{Professeur}$
  - Tout professeur est titulaire d'un diplôme de Doctorat.
  - $\text{Professeur} \sqsubseteq (\exists \text{diplôme.Doctorat})$

- Pour obtenir un diplôme de Doctorat, il faut posséder un DEA.
- $(\exists \text{diplôme.Doctorat}) \sqsubseteq (\exists \text{diplôme.DEA})$
- Les DEA et les Doctorats sont des diplômes distincts.
- $\text{Doctorat} \sqcap \text{DEA} \sqsubseteq \perp$

#### ■ Niveau assertionnel

- Jean enseigne un "cours d'IA".
- $\text{enseigne}(\text{Jean}, \text{Cours-IA})$
- Jean a au plus un diplôme.
- $(\leq 1 \text{diplôme})(\text{Jean})$
- Un "cours d'IA" est un cours.
- $\text{Cours}(\text{Cours-IA})$

## Aspects théoriques : Logique de description

### ■ Les principaux tests d'inférence

- La connaissance est-elle **correcte** (intuitions capturées) ?
  - Est-ce que C est **subsumé** par D selon T ? ( $C^I \subseteq D^I$  pour *tout* modèle I de T)
- La connaissance est-elle **minimalement redondante** (pas de synonymes inattendus)
  - Est-ce que C est **équivalent** à D selon T ? ( $C^I = D^I$  dans *tout* modèle I de T)
- La connaissance a-elle **plein sens (meaningful)** (classes doivent être non vide)
  - Est-ce que C est **satisfiable** selon T ? ( $C^I \neq \emptyset$ ; dans *un* modèle I de T)
- La connaissance est-elle **consultable** (Querying knowledge)
  - Est-ce que x est une **instance de concept** C selon T ? ( $x^I \in C^I$  dans tout modèle I de T)
  - Est-ce que  $\langle x, y \rangle$  est une **instance de rôle** R selon T ? ( $\langle x^I, y^I \rangle \in R^I$  dans *tout* modèle I de T)
- Les connaissances sont-elles **incompatibles** :
  - Deux concepts C et D sont-ils **incompatibles** selon T ? ( $C^I \cap D^I = \emptyset$  dans tout modèle I de T)

## Aspects théoriques : Logique de description

### ■ Réduction de tests

- On peut réduire les tests de base :
  - Au test de **satisfaisabilité** :
    - C est **subsumé** par D si seulement si  $(C \sqcap \neg D)$  n'est pas satisfaisable
    - C et D sont **équivalents** si seulement si  $(C \sqcap \neg D) \sqcup (D \sqcap \neg C)$  n'est pas satisfaisable
    - C et D sont **incompatibles** si seulement si  $(C \sqcap D)$  n'est pas satisfaisable
  - Ou au test de **subsumption** :
    - C n'est pas satisfaisable si seulement si C est subsumé par  $\perp$
    - C et D sont **équivalents** si seulement si C est subsumé par D et D est subsumé par C
    - C et D sont **incompatibles** si seulement si  $(C \sqcap D)$  est subsumé par  $\perp$



## Aspects théoriques : Logique de description

- Propriétés de transformation
  - Les constructeurs  $\sqcap$  et  $\sqcup$  obéissent aux règles suivantes
    - Idempotence:  $C \sqcap C \equiv C$  et  $C \sqcup C \equiv C$
    - Commutativité:  $C \sqcap D = D \sqcap C$  et  $C \sqcup D = D \sqcup C$
    - Associativité :  $C \sqcap (D \sqcap E) = (C \sqcap D) \sqcap E$  et  $C \sqcup (D \sqcup E) = (C \sqcup D) \sqcup E$
    - Si  $C \sqsubseteq D$  et  $C \sqsubseteq E$  alors  $C \sqsubseteq D \sqcap C$
    - Si  $C \sqsubseteq D$  et  $E \sqsubseteq D$  alors  $C \sqcup E \sqsubseteq D$
    - Si  $C \sqsubseteq D$  alors  $C \sqcap X \sqsubseteq D$  pour tout description  $X$
    - Si  $C \sqsubseteq D$  alors  $C \sqsubseteq D \sqcup X$  pour tout description  $X$

## Aspects théoriques : Logique de description

- Deux mécanismes de raisonnement
  - Les *algorithmes de type normalisation-comparaison* (algorithmes NC) :
    - un processus de normalisation produit les *formes normales* des concepts définis qui sont ensuite
    - effectivement comparées à l'aide de règles de comparaison
  - La *méthode des tableaux sémantiques* : la question *est-ce que D subsume C* ( $C \sqsubseteq D$ ) est transformée en *est-ce que  $(C \sqcap \neg D)$  est non satisfiable*.
  - Note : le langage de description des concepts doit être muni de la négation des concepts définis

## Aspects théoriques : Logique de description

- **Algorithme de tableau basique (sans TBox)**
  - Utilisé pour le **test de satisfiabilité** (consistance) d'une expression de concepts **C**
  - Essai de **construire un arbre de modèles hypothétiques** dont la racine est  $\neg C$  (sous forme normale : négation devant concepts atomiques)
  - **Décomposer syntaxiquement** et successivement de C
    - En appliquant des **règles d'expansion** de tableau
    - En **évaluant des contraintes** sur les éléments du modèle
  - Les **règles d'expansion** de tableau correspondent aux constructeurs dans le langage LD utilisés ( $\forall, \wedge, \exists, \geq, \leq$ )
    - Certaines règles sont **non déterministes** (par exemple :  $\exists, \forall$ )
    - En pratique, cela **explose l'espace** de recherche
  - Arrêter quand **aucune règle soit applicable** ou **motifs contradictoires** (clash) par exemple  $(A(x), \neg A(x))$
  - Le **contrôle de cycle (blocking)** peut-être nécessaire pour la terminaison
  - C sera **satisfiable** ssi les règles peuvent-être appliquées telles que l'arbre construit **ne contenant que** des feuilles avec **motifs contradictoires** (clash)

## Aspects théoriques : Logique de description

- **Propriétés algorithmique**
  - **Procédure de décision** : un algorithme est appelé une procédure de décision si et seulement si il vérifie 3 propriétés suivantes :
    - **Arrêt** : l'algorithme doit donner le résultat à un temps fini
    - **Correction** : les inférences produites sont en accord avec la sémantique associée, autrement dit, ce qui est vrai sur le plan syntaxique l'est sur le plan sémantique.
    - **Complétude** : toutes les formules valides — vraies sur le plan sémantique — peuvent être démontrées sur le plan syntaxique.
  - **Complexité d'un algorithme** : Un algorithme doit appartenir à une classe de complexité qui qualifie son « coût » en temps et en espace de mémoire
    - $NLOGSPACE \sqsubseteq P \sqsubseteq NP \sqsubseteq PSPACE \sqsubseteq EXPTIME \sqsubseteq NEXPTIME \sqsubseteq EXPSPACE$
    - $P \neq PSPACE, PSPACE = NPSpace, PSPACE \neq EXPSPACE$

## Aspects théoriques : Logique de description

- Propriétés des algorithmes de tableau
  - Théorème :
    - Les algorithmes de tableaux sont des **procédures** de décision pour la satisfiabilité de concepts i.e., l'algorithme retourne "SAT" ssi le concept est satisfiable
    - Preuve (schématique)
      - **Terminaison**
        - La largeur (nombre de règles applicables par nœud) et la profondeur (avec la stratégie de blocking) de l'arbre de tableaux sont bornées
      - **Correct**
        - A partir d'un arbre de tableaux d'une formule satisfiable on peut construire un un modèle pour cette formule
      - **Complet**
        - A partir d'un modèle d'une formule satisfiable, on peut diriger le processus d'application de règles d'expansion tel que arbre de tableaux obtenu contenant un tableau correspondant à ce modèle

## Aspects théoriques : Logique de description Complexité des langages (sans TBox)

Expressivité		$\models C \sqsubseteq D$	$\models C(a)$
$\mathcal{FL}\mathcal{E}$	$C \sqcap D$ $\forall r.C$ $\exists r.T$	P	P
$\mathcal{AL}$	$\neg A$	P	P
$\mathcal{AL}\mathcal{E}$	$\exists r.C$	NP	PSPACE
$\mathcal{ALC}$ ( $\sim$ modal $K_{(m)}$ )	$\neg C$	PSPACE	
$\mathcal{ALCO}$	$\{a_1, \dots\}$	PSPACE	
$\mathcal{SHOIN}(\mathcal{D})$		EXPTIME	
$\mathcal{SHIQ}$		EXPTIME	
$\text{NLOGSPACE} \sqsubseteq \text{P} \sqsubseteq \text{NP} \sqsubseteq \text{PSPACE} = \text{NPSpace} \sqsubseteq \text{EXPTIME} \sqsubseteq \text{NEXPTIME} \sqsubseteq \text{EXPSpace}$			

## Aspects théoriques : Logique de description

### Complexité des langages (suivant TBox)

Expressivité	$\models C \sqsubseteq D$	$\models C(a)$
$\mathcal{FL}\mathcal{E}$	P	P
$\mathcal{AL}$	EXPTIME	EXPTIME
$\mathcal{AL}\mathcal{E}$	PSPACE	PSPACE
$\mathcal{ALC}$ (avec domaine concret et TBox cyclique)	NEXPTIME	
$\mathcal{ALC}$ (avec $R\text{InS}$ , $R\text{uS}$ , $\neg R$ et TBox vide)	NEXPTIME	
$\mathcal{SHOIN}(\mathcal{D})$	NEXPTIME	
$\mathcal{SHIQ}$	NEXPTIME	
$\text{NLOGSPACE} \sqsubseteq \text{P} \sqsubseteq \text{NP} \sqsubseteq \text{PSPACE} = \text{NPSpace} \sqsubseteq \text{EXPTIME} \sqsubseteq \text{NEXPTIME} \sqsubseteq \text{EXSPACE}$		

## Aspects théoriques : Logique de description

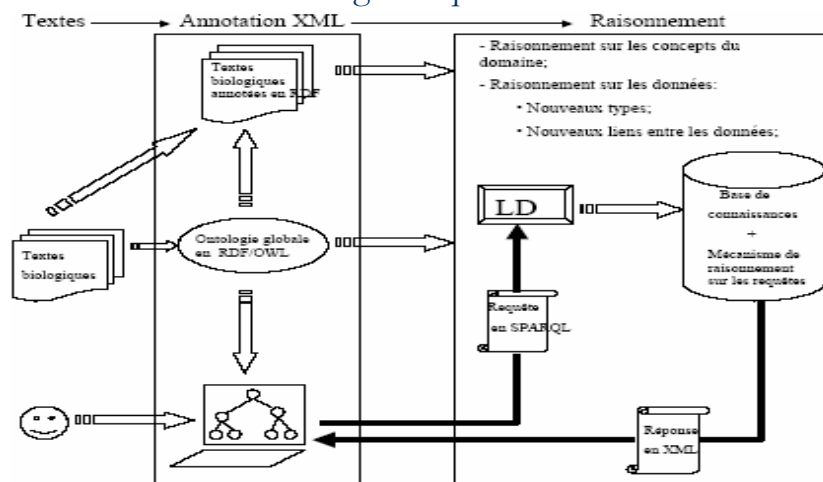
### Exemple : Famille (DL Handbook)

<p><b>TBOX</b></p> <p>Femme <math>\equiv</math> Personne <math>\sqcap</math> Féminin            Homme <math>\equiv</math> Personne <math>\sqcap</math> <math>\neg</math>Féminin            Mère <math>\equiv</math> Femme <math>\sqcap</math> <math>\exists</math>aEnfant.Personne            Père <math>\equiv</math> Homme <math>\sqcap</math> <math>\exists</math>aEnfant.Personne            Parent <math>\equiv</math> Père <math>\sqcup</math> Mère            GrandeMère <math>\equiv</math> Femme <math>\sqcap</math> <math>\exists</math>aEnfant.Parent            MèrePlusDe3Enfants <math>\equiv</math> Mère <math>\sqcap</math> <math>\geq 3</math>aEnfant            MèreSansFille <math>\equiv</math> Femme <math>\sqcap</math> <math>\forall</math>aEnfant. <math>\neg</math>Femme            Epouse <math>\equiv</math> Femme <math>\sqcap</math> <math>\exists</math>eEpoux.Homme</p>	<p><b>Concepts atomiques</b></p> <p>Personne Féminin</p> <p><b>Rôles atomiques</b></p> <p>aEnfant eEpoux</p> <p><b>Concepts définis</b></p> <p>Femme Homme Mère Père Parent GrandeMère MèrePlusDe3Enfants MèreSansFille Epouse</p>
<p><b>ABOX</b></p> <p>MèreSansFille(Manon)            aEnfant(Manon, Pierre)            aEnfant(Manon, Paul)            Père(Pierre)            aEnfant(Pierre, Alexandre)</p>	<p><b>Individus</b></p> <p>Manon Pierre Paul Pierre Alexandre</p>

# Aspects pratiques

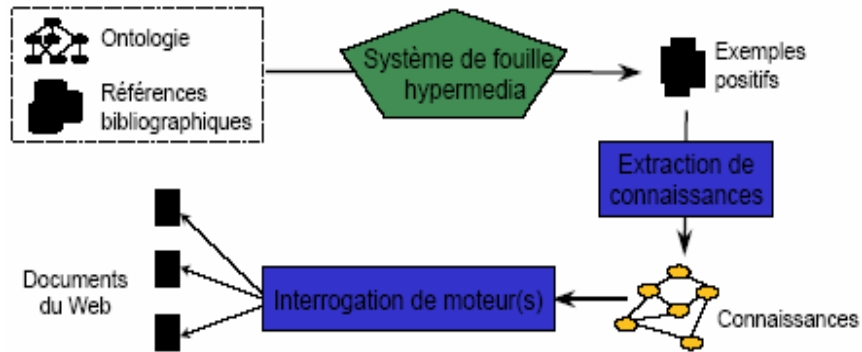
## Aspects pratiques : Annotation de la sémantique

### ■ Schéma d'annotation générique



## Aspects pratiques : Annotation de la sémantique

- Fouille de données et annotation : un point de vue
  - Moteur de recherche = passerelle entre la fouille de données et le Web



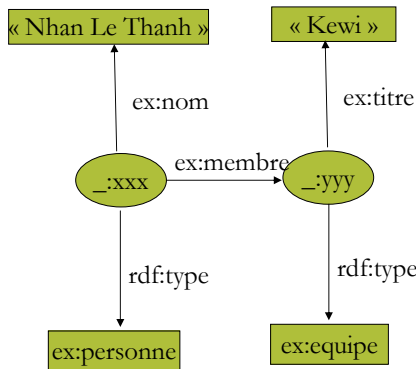
## Aspects pratiques : Annotation de la sémantique

- Langage RDF (Resource Description Framework)
  - un modèle de **graphe** destiné à décrire de façon formelle les **ressource Web** et leurs **métadonnées**, de façon à permettre le traitement automatique de telles descriptions.
  - Un document RDF ainsi formé correspond à un multi-graphe orienté étiqueté. Chaque triplet correspond alors à un arc orienté dont le label est le prédicat, le nœud source est le sujet et le nœud cible est l'objet.



## Aspects pratiques : Annotation de la sémantique

### ■ Exemple : graphe RDF



```

<rdf:Property rdf:ID='nom'>
  <rdfs:domain rdf:resource='#Personne'/>
  <rdfs:range rdf:resource='&rdfs:Literal'/>
</rdf:Property>
<rdf:Property rdf:ID='titre'>
  <rdfs:domain rdf:resource='#equipe'/>
  <rdfs:range rdf:resource='&rdfs:Literal'/>
</rdf:Property>
<rdf:Property rdf:ID='membre'>
  <rdfs:domain rdf:resource='#personne'/>
  <rdfs:range rdf:resource='#equipe'/>
</rdf:Property>
  
```

#### Anotation

```

<rdf:Description
  rdf:about='http://www.i3s.unice.fr/equipe'>
  <rdf:type equipe rdf:resource='#equipe'/>
  <titre>Kewi </titre>
  <membre>
    <personne><nom> Nhan Le
    Thanh</nom></personne>
    <personne><nom> Serge
    Miranda</nom></personne>
  </membre>
</equipe>
</rdf:Description>
  
```

## Aspects pratiques : Annotation de la sémantique

### ■ Syntaxe XML

```

<rdf:Description
  rdf:about='http://www.mnds.fr/cours/WS'>
  <titre> Web sémantique </titre>
</rdf:Description>
  
```

### ■ On y ajoute de sens :

- La ressource est un cours,
- un cours a des enseignants,
- l'enseignant est un chercheur de l'I3S, etc.

```

ex:WS rdf:type ns:Cours
ex:WS ns:titre "W"eb Sémantique
ns:Cours rdf:type rdfs:Class
  
```

```

<rdf:Description rdf:about='http://
www.mnds.fr/cours/WS'>
  <rdf:type rdf:resource='#Cours'/>
  <titre>Web sémantique</titre>
</rdf:Description>
<rdfs:Class ID='Cours'/>
  
```

## Aspects pratiques : Annotation de la sémantique

### ■ Les classes :

```
ns:Cours rdf:type rdfs:Class
ns:CoursMBDS rdf:type rdfs:Class
ns:CoursMBDS rdfs:subClassOf ns:Cours
ns:CoursEPU rdfs:subClassOf ns:Cours
```

```
<rdfs:Class rdf:ID='Cours' />
<rdfs:Class rdf:ID='CoursMBDS' />
<rdfs:subClassOf rdf:resource='#Cours' />
</rdfs:Class>
<rdfs:Class rdf:ID='CourseEPU' />
<rdfs:subClassOf rdf:resource='#Cours' />
</rdfs:Class>
```

## Aspects pratiques : Annotation de la sémantique

### ■ Les propriétés :

```
<rdf:Property rdf:ID='enseignant'>
  <rdfs:domain rdf:resource='#Cours' />
  <rdfs:range rdf:resource='#Personne' />
</rdf:Property>
<rdf:Property rdf:ID='nom'>
  <rdfs:domain rdf:resource='#Personne' />
  <rdfs:range rdf:resource='&rdfs:Literal' />
</rdf:Property>

<rdfs:Class rdf:ID='CoursCommun'>
  <rdfs:subClassOf rdf:resource='#CoursDEA' />
  <rdfs:subClassOf rdf:resource='#CoursESSI' />
</rdfs:Class>
<rdf:Property rdf:ID='titre'>
  <rdfs:domain rdf:resource='#Cours' />
  <rdfs:range rdf:resource='&rdfs:Literal' />
</rdf:Property>
```



## Aspects pratiques : Annotation de la sémantique

### ■ Annotation

```
<rdf:Description
  rdf:about='http://www.mbds.fr/cours/WS'>
  <rdf:type rdf:resource='#Cours' />
  <titre>Web sémantique</titre>
  <num>WS</num>
  <enseignant>
    <Professeur
      rdf:about='http://www.i3s.unice.fr/nlethanh'>
        <nom>Nhan Le Thanh</nom>
        <universite>Nice Sophia-Antipolis</université>
    </Professeur>
  </enseignant>
</rdf:Description>
```

## Aspects pratiques : Annotation de la sémantique

### ■ Exemple application RDF : Le Dublin Core :

- ❑ définit un certain nombre de champs de métadonnées utilisables pour décrire les ressources du Web
- ❑ 15 *champs* ou *éléments* standardisés
  - titre, créateur, description, sujet, date, source, droits...
- ❑ Les champs peuvent être précisés à l'aide de *raffinements*
  - description : résumé, table des matières
- ❑ Les valeurs de ces champs reprennent généralement d'autres standards

## Aspects pratiques : Annotation de la sémantique

### ■ Exemple Dublin Core en HTML:

```
<html>
<head><title>Cours MBDS</title>
<link rel="schema:DC" href="http://purl.org/dc/elements/1.1/" />
<meta name="DC.Title" lang="fr" content="Cours MBDS en HTML" />
<meta name="DC.Date.created" scheme="W3CDTF" content="2004-09-27" />
<meta name="DC.Date.modified" scheme="W3CDTF" content="2004-09-28" />
<meta name="DC.Subject" lang="fr" content="HTML,document, Dublin Core" />
<meta name="DC.Language" scheme="RFC3066" content="fr-FR" />
</head>
<body></body>
</html>
```

## Aspects pratiques : Annotation de la sémantique

### ■ Exemple Dublin Core en RDF:

```
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntaxns#"
xmlns:dc="http://purl.org/dc/elements/1.1/">
<rdf:Description about="Cours_MBDS.html">
<dc:title>Cours MBDS en HTML</dc:title>
<dc:created>2008-02-27</dc:created>
<dc:modified>2008-02-28</dc:modified>
<dc:language>fr</dc:language>
</rdf:Description>
</rdf:RDF>
```

## Aspects pratiques : Annotation de la sémantique

### ■ Application de RDF Creative Commons

- Objectifs :
  - adapter les droits des auteurs au nouveau médium qu'est Internet,
  - fournir un cadre juridique au partage sur le web d'oeuvres de l'esprit comme les images, les sons ou les textes
- Principe : "un jeu de logos"
  - la combinaison de ces 4 briques fournit 6 types de licence



## Aspects pratiques : Annotation de la sémantique

### ■ Application de RDF Creative Commons

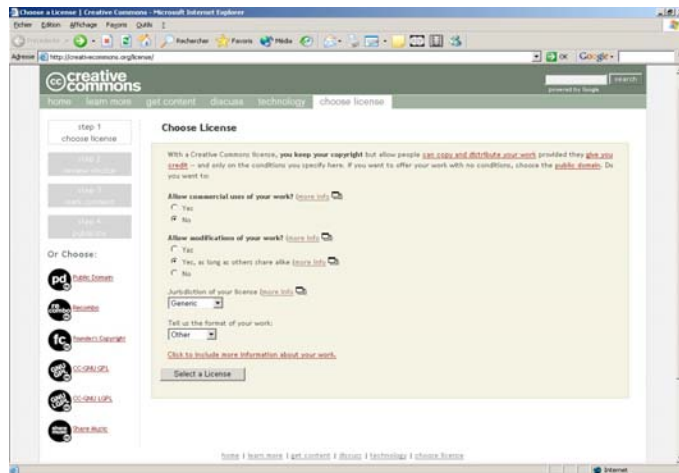
```
<rdf:RDF xmlns="http://web.resource.org/cc/"  
xmlns:dc="http://purl.org/dc/elements/1.1/"  
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
```

```
<Work rdf:about="">  
<license rdf:resource="http://creativecommons.org/licenses/by-nc-sa/2.0/" />  
</Work>  
</rdf:RDF>
```

```
<License rdf:about="http://creativecommons.org/licenses/by-nc-sa/2.0/">  
<permits rdf:resource="http://web.resource.org/cc/Reproduction" />  
<permits rdf:resource="http://web.resource.org/cc/Distribution" />  
<requires rdf:resource="http://web.resource.org/cc/Notice" />  
<requires rdf:resource="http://web.resource.org/cc/Attribution" />  
<prohibits rdf:resource="http://web.resource.org/cc/CommercialUse" />  
<permits rdf:resource="http://web.resource.org/cc/DerivativeWorks" />  
<requires rdf:resource="http://web.resource.org/cc/ShareAlike" />  
</License>
```

## Aspects pratiques : langages du web

### ■ Application de RDF Creative Commons



## Aspects pratiques : Annotation de la sémantique

### ■ Application de RDF Creative Commons



## Aspects pratiques : Annotation de la sémantique

- Application de RDF : RSS (RDF Site Summary)
  - permet de publier une succession d'éléments d'information (syndication des contenus sur le web).
  - dès qu'il est présenté en RDF, il permet la manipulation et la sélection automatique
  - utilisé pour obtenir les mises à jour d'information dont la nature change fréquemment, typiquement cela peut être :
    - des listes des tâches dans un projet, des prix, des alertes de toutes natures, des nouveaux emplois proposés, les sites d'information ou les blogs, les podcasts et videocasts.
  - deux autres formats : Rich Site Summary et Really simple Syndication

## Aspects pratiques : Annotation de la sémantique

- Exemple RSS+DC+CC :

```
<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF xmlns:ct="http://xmlfr.org/dtd/content" ...>
<channel rdf:about="http://xmlfr.org/documentations/articles/channel.rss10">
<title>Articles</title>
<link>http://xmlfr.org/documentations/articles/</link>
<description>Articles de fond sur XML.</description>
<dc:source rdf:resource="http://xmlfr.org/documentations/articles/" />
<dc:language>fr</dc:language>
<dc:publisher>Eric van der Vlist (mailto:vdv@dyomedeia.com)</dc:publisher>
<dc:creator>Eric van der Vlist (mailto:vdv@dyomedeia.com)</dc:creator>
<cc:license rdf:resource="http://creativecommons.org/licenses/by-ncsa/2.0/" />
<image rdf:resource="http://xmlfr.org/bandeaux/xmlfr_88x31.gif" />
<items>
<rdf:Seq>
<rdf:li rdf:resource="http://xmlfr.org/documentations/articles/040331-0001" />
<rdf:li rdf:resource="http://xmlfr.org/documentations/articles/040130-0001" />
<rdf:li rdf:resource="http://xmlfr.org/documentations/articles/031126-0001" />
</rdf:Seq>
</items>
</channel>
```

## Aspects pratiques : Annotation de la sémantique

### ■ Synthèse

- ❑ Il existe de nombreuses ressources en RDF;
- ❑ Elles sont utilisées conjointement.
- ❑ Ces ressources constituent un embryon de web sémantique...

### ■ Besoins

- ❑ Liens vers les documents en ligne;
- ❑ Décrire les méta-données (DublinCore);
- ❑ Connaissance des droits (Creative Commons);
- ❑ Liens vers les personnes et les organisations (FOAF);
- ❑ Référence à des évènements (iCalendar);
- ❑ Engendrer des fils (RSS)

## Aspects pratiques : Ontologie et modélisation

### ■ Origine de l'Ontologie :

- ❑ Une branche de philosophie qui cherche un accord avec la nature et l'organisation de la **réalité**
- ❑ Science de l'Être (Aristote, Métaphysiques, IV, 1)
  - Recherche de la réponse aux questions:
    - ❑ *Qu'est-ce que c'est les caractéristiques de l'Être?*
    - ❑ *Eventuellement, Qu'est-ce que c'est l'Être ?*
- ❑ Comment les «choses » sont-elles **classifiées** ?
  - La classification a été étudiée depuis très long temps dans l'histoire

## Aspects pratiques : Ontologie et modélisation

- Ontologie en Informatique
  - Une ontologie est un produit d'ingénierie qui consiste en :
    - un **vocabulaire** utilisé pour décrire (une vue particulière de) un certain domaine
    - une **spécification explicite** des **sens attendus** du vocabulaire.
    - Presque toujours inclure comment les concepts sont-ils classifiés
  - Elle capture des contraintes, des **connaissances additionnelles** sur le domaine
  - Idéalement une ontologie permet de :
    - Capturer des **compréhensions partagées** d'un domaine
    - Produire un **modèle formel** du domaine **manipulable par ordinateur**

## Aspects pratiques : Ontologie et modélisation

- Où utiliserons les ontologies
  - e-Science, par exemple, Bioinformatiques
    - L'Ontologie de Gènes (G.O)
    - L'Ontologie de Protéines (MGED)
  - Médecine
    - Terminologies (ontologie UMLS)
  - Databases
    - Intégration
    - Requêtes d'interrogation
  - Interfaces d'utilisateur
  - Linguistiques
  - **Le Sémantique Web**

## Aspects pratiques : Annotation de la sémantique

- Modélisation avec RDF : Sémantique du langage
  - Tous les énoncés RDF sont considérés comme vrais, dans un même environnement, la conjonction des énoncés est considérée comme vrai également
  - Il n'y a pas de contexte local (dans lequel quelque chose serait vrai)
  - Il n'y a pas de négation
  - RDF est monotone : ce qui est vrai reste vrai au fur et à mesure que l'on rajoute de l'information
  - RDF est un sous-ensemble de la logique du premier ordre avec des prédicats binaires et des variables existentielles.

## Aspects pratiques : Annotation de la sémantique

- Typage en RDF :
  - Les règles de typage diffèrent des modèles objet traditionnels
- Domaine : L'utilisation d'une propriété sur une ressource entraîne que la ressource soit du type indiqué par le domaine de la propriété
- Rang : Si une ressource est la valeur d'une propriété, la ressource est du type indiqué par le range de la propriété
- Class : Si une ressource a pour type C, alors C est une classe



## Aspects pratiques : Annotation de la sémantique

### ■ Typage en RDF : Exemples

#### □ Domaine

`ex:title rdfs:domain ex:Cours`

`ex:log11 ex:title "Modélisation des connaissances"`

$\Rightarrow$  `ex:log11 rdf:type ex:Cours`

#### □ Range

`ex:enseignant rdfs:range ex:Person`

`ex:log11 ex:enseignant ex:olivier`

$\Rightarrow$  `ex:olivier rdf:type ex:Person`

#### □ Class

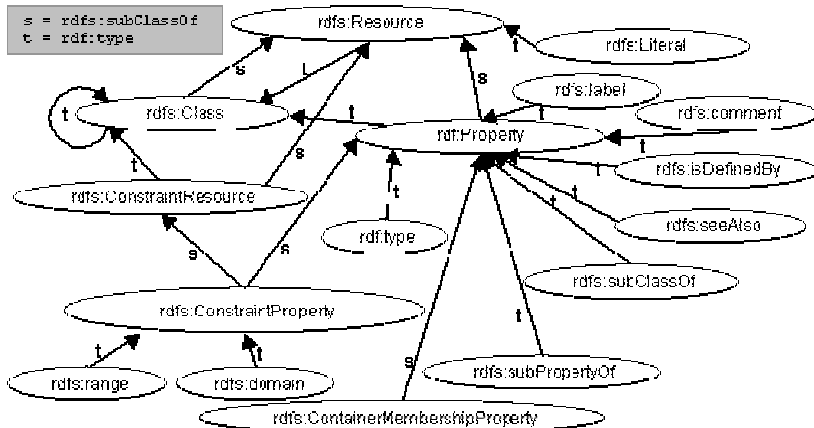
`rdf:type rdfs:range rdfs:Class`

`ex:olivier rdf:type ex:Person`

$\Rightarrow$  `ex:Person rdf:type rdfs:Class`

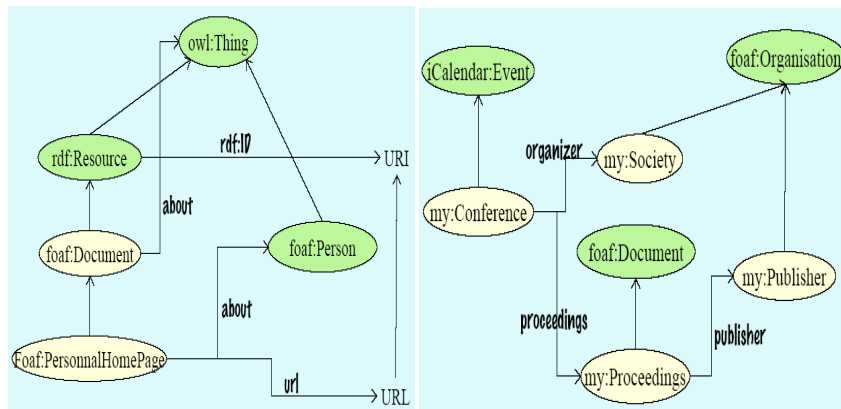
## Aspects pratiques : Annotation de la sémantique

### ■ Méta modèle RDF



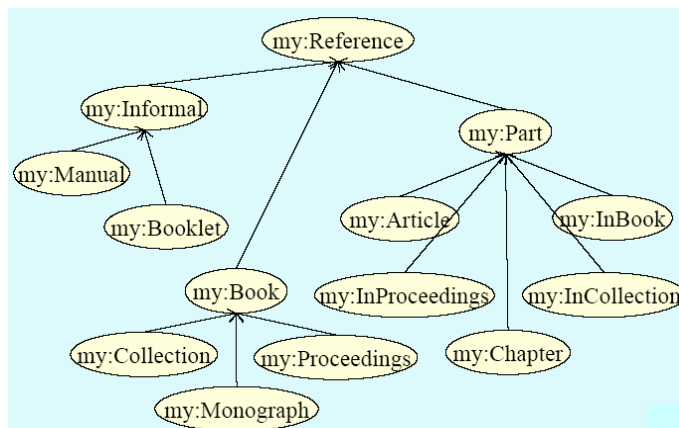
## Aspects pratiques : Ontologie et modélisation

### ■ Exemple : Intégration de ressources avec RDF/RDFS



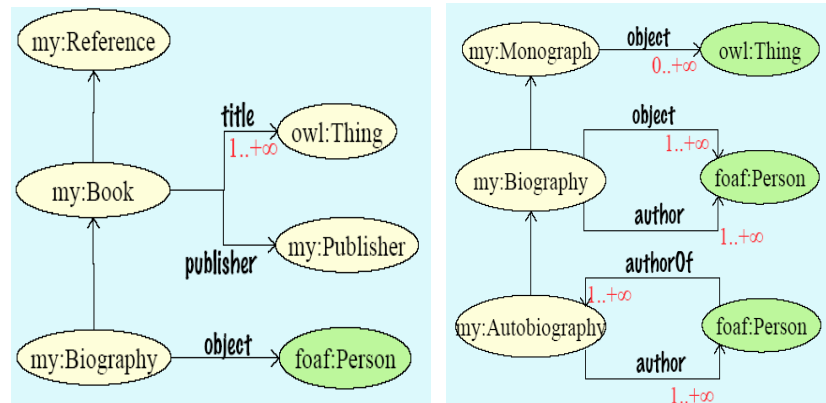
## Aspects pratiques : Ontologie et modélisation

### ■ Exemple : Création des ressources avec RDF/RDFS



## Aspects pratiques : Ontologie et modélisation

### ■ Exemple : Ajout des contraintes avec RDF/RDFS



## Aspects pratiques : Ontologie et modélisation

### ■ RDFS : RDF Schemas

- Introduction de classes et de propriétés;
  - Spécialisation de classes et de propriétés;
  - Restriction du domaine et codomaine des propriétés.
- Méta-modèle réflexif (tout est ressource, les classes et propriétés spécialisent les ressources, ressources et propriétés sont des classes...).

## Aspects pratiques : Ontologie et modélisation

### ■ Méta modèle de RDF

- Ressource : `rdfs:Resource` Classe racine de la hiérarchie de subsomption des classes, instance de `rdfs:Class`
- Classe : `rdfs:Class` La meta classe qui engendre des classes, instance d'elle même, sous-classe de `rdfs:Resource`
- spécification

```
rdfs:Resource rdf:type rdfs:Class
rdfs:Class rdf:type rdfs:Class
rdfs:Class rdfs:subClassOf rdfs:Resource
```
- La classe des propriétés `rdf:Property`

```
rdf:Property rdf:type rdfs:Class
rdf:Property rdfs:subClassOf rdfs:Resource
```

## Aspects pratiques : Ontologie et modélisation

### ■ Métamodèle de RDF

- Typage : `rdf:type` La relation d'instanciation

```
rdf:type rdf:type rdf:Property
```
- Relation de subsomption (en fait de spécialisation) :

```
rdfs:subClassOf
rdfs:subClassOf rdf:type rdf:Property
```
- Relation de subsomption des propriétés :

```
rdfs:subPropertyOf
rdfs:subPropertyOf rdf:type rdf:Property
```

## Aspects pratiques : Ontologie et modélisation

### ■ Exemple : Un schéma RDFS :

```
<rdfs:Class rdf:ID="Document">
<rdfs:subClassOf rdf:resource="#Entity"/>
<rdfs:subClassOf rdf:resource="#EntityConcerningATopic"/>
<rdfs:subClassOf rdf:resource="#NumberableEntity"/>
<rdfs:comment xml:lang="en">Entity including elements serving as
  a representation of thinking.</rdfs:comment>
<rdfs:comment xml:lang="fr">Entite comprenant des elements de
  representation de la pensee.</rdfs:comment>
<rdfs:label xml:lang="en">document</rdfs:label>
<rdfs:label xml:lang="fr">document</rdfs:label>
</rdfs:Class>
```

## Aspects pratiques : Ontologie et modélisation

### Ontologie et Logiques de description

- OWL est un standard de langage d'ontologie de W3C basé sur les logiques de description
  - Les axiomes et les constructeurs d'OWL sont restreints pour que le raisonnement soit décidable
- La Sémantique Web s'organise dans une architecture en couche
  - XML produit la couche de transport syntaxique
  - RDF(S) produit un langage relationnel basique et des primitives ontologiques simples
  - OWL produit un langage d'ontologie puissant mais restant décidable
  - Autres couches (par exemple SWRL – Semantic Web Rule Language) seront des extensions de OWL
    - La plupart seront indécidables
- Besoin des “expériences d'implémentation”

## Aspects pratiques : Ontologie et modélisation

### Concepts (classes) et constructeurs

Constructor	DL Syntax	Example	FOL Syntax
intersectionOf	$C_1 \sqcap \dots \sqcap C_n$	Human $\sqcap$ Male	$C_1(x) \wedge \dots \wedge C_n(x)$
unionOf	$C_1 \sqcup \dots \sqcup C_n$	Doctor $\sqcup$ Lawyer	$C_1(x) \vee \dots \vee C_n(x)$
complementOf	$\neg C$	$\neg$ Male	$\neg C(x)$
oneOf	$\{x_1\} \sqcup \dots \sqcup \{x_n\}$	{john} $\sqcup$ {mary}	$x = x_1 \vee \dots \vee x = x_n$
allValuesFrom	$\forall P.C$	$\forall$ hasChild.Doctor	$\forall y.P(x, y) \rightarrow C(y)$
someValuesFrom	$\exists P.C$	$\exists$ hasChild.Lawyer	$\exists y.P(x, y) \wedge C(y)$
maxCardinality	$\leq nP$	$\leq 1$ hasChild	$\exists^{\leq n} y.P(x, y)$
minCardinality	$\geq nP$	$\geq 2$ hasChild	$\exists^{\geq n} y.P(x, y)$

- C est un concept (classe); P est un rôle (propriété); x est un nom d'individu
- Types de données XMLS sont des classes dans 8P.C and 9P.C
  - Forme de restriction de DL avec domaines concrets

## Aspects pratiques : Ontologie et modélisation

### Ontologie et Base de connaissances LD

- Une Ontologie OWL est équivalente à une Base de Connaissances de LD (BCLD)
- Une ontologie OWL consiste en un ensemble d'axiomes et faits
  - *Note: une ontologie comprend usuellement seulement des axiomes dans la TBox (schéma)---OWL est donc non-standard pour cette catégorie*
- Rappel : une BCLD  $\mathbf{K}$  est la paire  $\langle \mathbf{T}, \mathbf{A} \rangle$  où
  - $\mathbf{T}$  est l'ensemble d'axiomes "terminologiques" (la TBox)
  - $\mathbf{A}$  est l'ensemble d'axiomes "assertionnels" (l'ABox)

## Aspects pratiques : Ontologie et modélisation

### ■ Ontologie / Axiomes TBox

OWL Syntax	DL Syntax	Example
subClassOf	$C_1 \sqsubseteq C_2$	Human $\sqsubseteq$ Animal $\sqcap$ Biped
equivalentClass	$C_1 \equiv C_2$	Man $\equiv$ Human $\sqcap$ Male
subPropertyOf	$P_1 \sqsubseteq P_2$	hasDaughter $\sqsubseteq$ hasChild
equivalentProperty	$P_1 \equiv P_2$	cost $\equiv$ price
transitiveProperty	$P^+ \sqsubseteq P$	ancestor <sup>+</sup> $\sqsubseteq$ ancestor

- Une équivalence à la logique PO/Modale
  - p.e. DL:  $C \vee D$  LPO:  $\forall x.C(x) \sqcup D(x)$  LM:  $C \sqcup D$
- Deux catégories différentes des axiomes TBox :
  - “Définitions”  $C \doteq D$  ou  $C \sqsubseteq D$  où  $C$  est un nom de concept
  - Axiomes d’Inclusion générale de concepts (GCI) où  $C$  peut-être complexe

## Aspects pratiques : Ontologie et modélisation

### Faits d’Ontologie / Axiomes ABox

OWL Syntax	DL Syntax	Example
type	$a : C$	John : Happy-Father
property	$\langle a, b \rangle : R$	$\langle$ John, Mary $\rangle$ : has-child

- *Note: dans une utilisation nominale (par exemple, dans SHOIN), on peut réduire les axiomes d’ABox aux axiomes d’inclusion de concept*
  - $a : C$  est équivalent à  $\{a\} \vee C$
  - $\langle a, b \rangle : r$  est équivalent à  $\{a\} \vee \exists r. \{b\}$

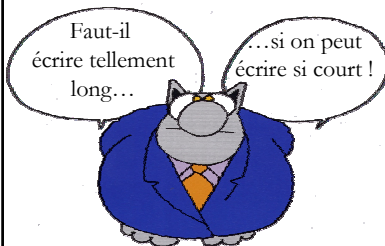
## Aspects pratiques : Ontologie et modélisation

- En langue naturelle

« Personne dont tous les enfants sont soit docteur soit ayant un enfant docteur »

- Dans la syntaxe DL :

$\text{Personne} \sqcap \forall \text{aEnfant} . (\text{Docteur} \sqcup \exists \text{aEnfant} . \text{Docteur})$



- Dans la syntaxe OWL

```
<owl:Class>
<owl:intersectionOf rdf:parseType=" collection">
<owl:Class rdf:about="#Personne"/>
<owl:Restriction>
<owl:onProperty rdf:resource="#aEnfant"/>
<owl:toClass>
<owl:unionOf rdf:parseType=" collection">
<owl:Class rdf:about="#Docteur"/>
<owl:Restriction>
<owl:onProperty rdf:resource="#aEnfant"/>
<owl:hasClass rdf:resource="#Docteur"/>
</owl:Restriction>
</owl:unionOf>
</owl:toClass>
</owl:Restriction>
</owl:intersectionOf>
</owl:Class>
```

## Aspects pratiques : Ontologie et modélisation

- Requêtes

- Trouver les cours dont N. Le Thanh est enseignant ?
- Qui enseigne le cours WS ?
- Y a-t-il un enseignant-chercheur ?
- Y a-t-il une Personne de l'UNS ?

- Moteurs de recherche RDF

- Jena : HP Lab
- SNOBASE : IBM
- Sesame : Java middleware, Administrator Nederland
- ICS-FORTH RDF Suite : BD, RDF Query Language
- Corese : INRIA



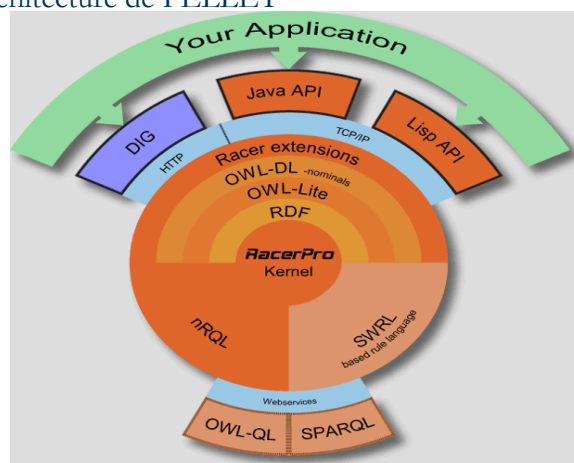
## Aspects pratiques : Web services sémantique

### ■ Moteur de recherche DL

- Les précurseurs DL : KL-ONE (77-79), CLASSIC et LOOM (89-92)
- FaCT++ (00-04) : Fast Classification of Terminologies
  - Université Manchester (ALC – SHIQ - OWL DL), C++
  - Issue commerciale avec CELEBRA (Network Inference)
- RACER (01-04) : **R**enamed **A**Box and **C**oncept **E**xpression **R**easoner
  - TBox et ABox (SHIQ (ALCQHIR+) – OWL DL)
  - Combiner DL et Algèbre Relationnelle
  - Issue commerciale avec RACERPro
- PELLET (03-05) :
  - Raisonnement avec Abox (SHION(D))
  - Java, Open Source

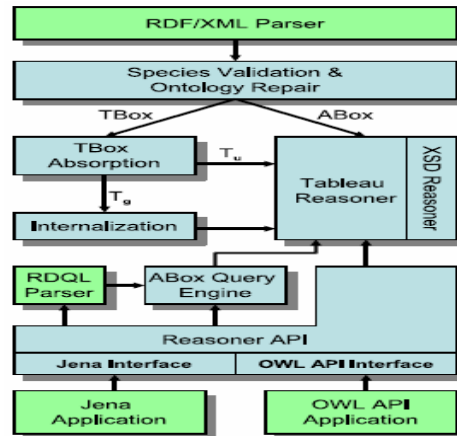
## Aspects pratiques : Web services sémantique Implémentation de systèmes

### □ Architecture de PELLET



## Aspects pratiques : Web services sémantique Implantation de systèmes

### □ Architecture de PELLET

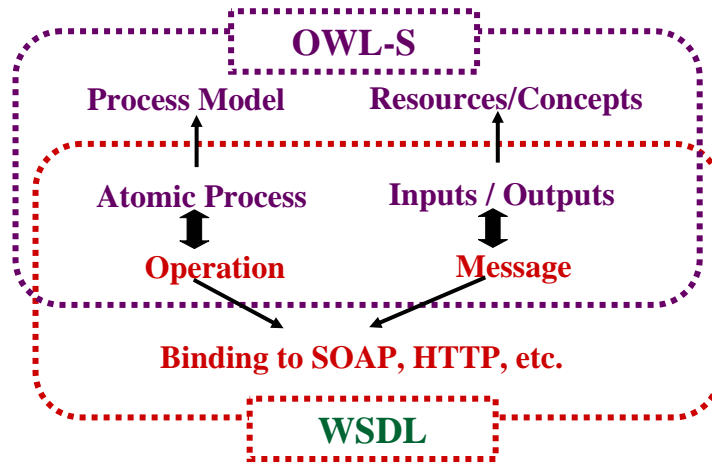


## Aspects pratiques : Web services sémantique WSDL-S et METEOR-S

### ■ Objectif :

- Principes de représentation de la sémantique (RS) des web services
- Le mécanisme de RS dans WSDL-S
- Comparaison avec la solution OWL-S (DAML-S)
- Etude des fonctionnalités de METEOR-S
- Vos remarques et critiques
- Identifier des différents problèmes à résoudre de RS dans les WS

## Aspects pratiques : Web services sémantique WSDL-S et METEOR-S



## Quelques tendances et thèmes de recherche

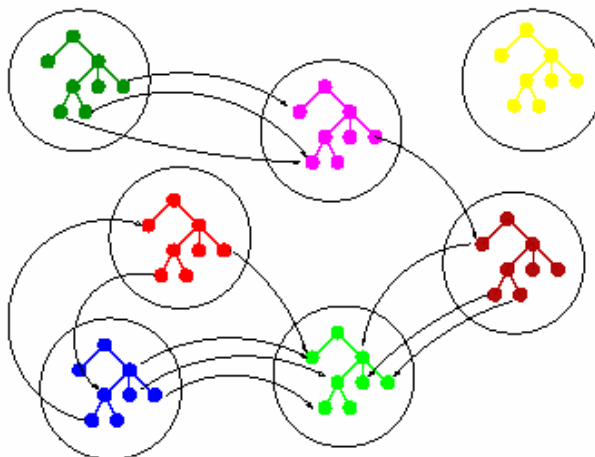
## Tendance et thèmes de recherche

### ■ Modèles de connaissances distribuées

- Objectif :
  - Modéliser des systèmes distribués par ontologies
- Approche de Coopération :
  - Principe : cohabiter des systèmes indépendants
  - Solutions : DDL et Drago,  $\epsilon$ -connection
- Approche de fédération :
  - Principe : Partager d'une base de connaissances communes dite « constitution »
  - Solution : décomposition de DL, ONDIL (en cours)

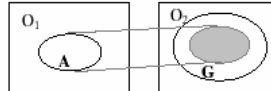
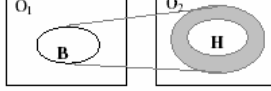
## Tendance et thèmes de recherche

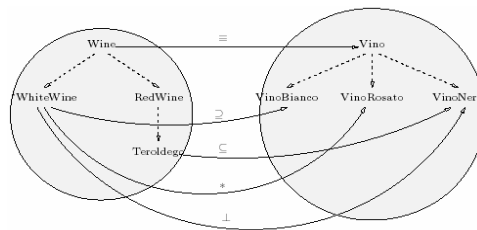
### ■ DDL et Drago



## Tendance et thèmes de recherche

### ■ DDL et Drago

Syntaxe	Sémantique	
Règle « <i>onto-bridge</i> » $i : A \sqsupseteq j : G$		
Règle « <i>into-bridge</i> » $i : B \sqsubseteq j : H$		



## Tendance et thèmes de recherche

### ■ Approche de fédération :

#### □ Création de fédération :

- Décomposition de la constitution : « chaque membre on des besoins différents »

#### □ Évolution de la fédération :

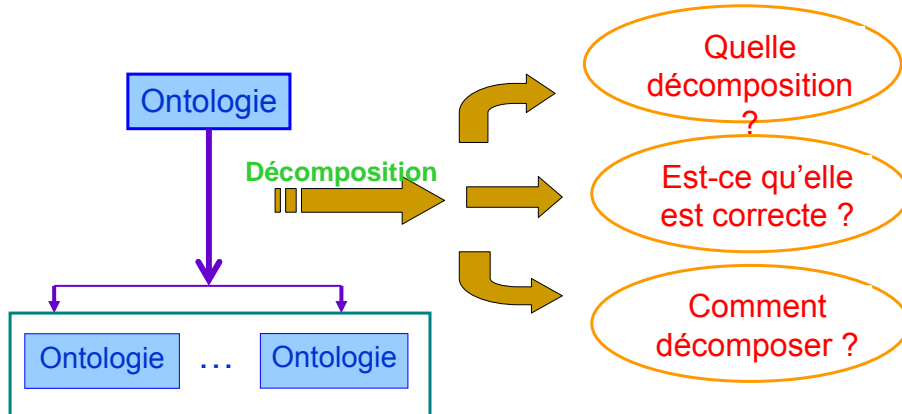
##### ■ Par chaque membre

- Soit des connaissances propres à lui (qui ne peuvent pas être déduites de la constitution)
- Soit des connaissances développées à partir de la constitution (qui doivent « respecter » la constitution)

#### □ Un thème de recherche de l'équipe Kewi

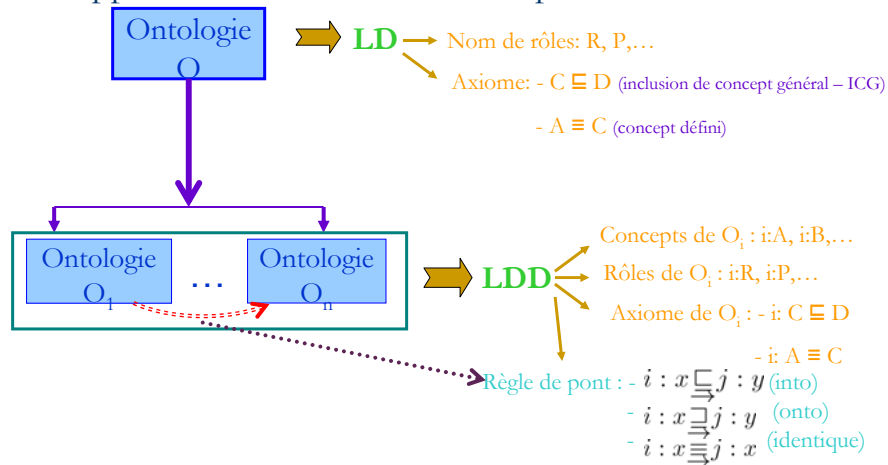
## Tendance et thèmes de recherche

- Approche de fédération : Création .décomposition



## Tendance et thèmes de recherche

- Approche de fédération : décomposition



## Tendance et thèmes de recherche

### ■ Approche de fédération : algorithme de décomposition

- Transformer une TBox en un graphe
- Décomposer le graphe présentant la TBox en des sous-graphes qui sont représentés dans un graphe d'intersection.
- Transformer le graphe d'intersection en une TBox distribuée.

Décomposition  
par séparateurs minimaux

Décomposition  
par coupes normalisées

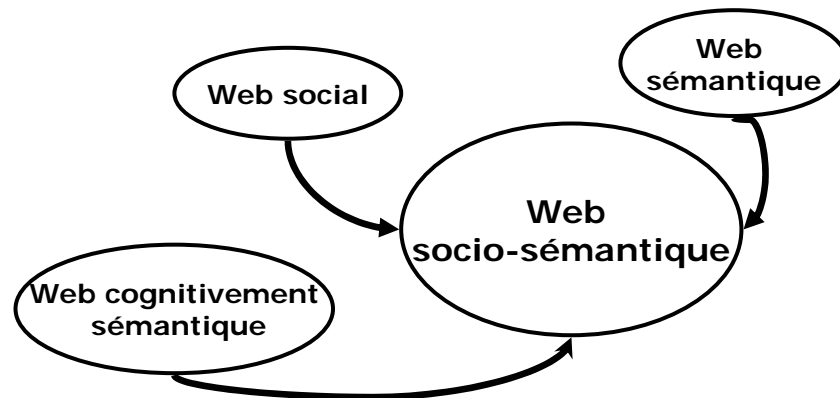
## Tendance et thèmes de recherche

### ■ Thème de recherche 1 : approche fédération

- Modélisation des échanges entre les membres dans une fédération : alignement et/ou importation des connaissances
- L'optimisation des requêtes : décompositions, exécution en parallèle, inférence
- Moteur d'inférence de fédération

## Tendance et thèmes de recherche

- Web socio-sémantique et centre d'usage



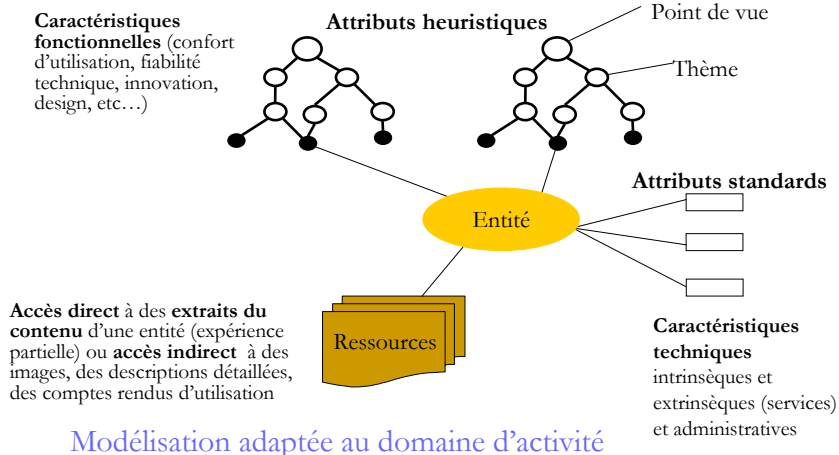
## Tendance et thèmes de recherche

- Web socio-sémantique selon Manuel Zacklad (UTT)
  - ❑ Un Web qui s'adresse à des communautés d'utilisateurs poursuivant des objectifs similaires
  - ❑ Un Web social qui participe de la construction d'une représentation structurée du domaine et du collectif
  - ❑ Une structuration progressive des réseaux sémantiques gérés par le collectif (présentant des enjeux pour le réseau social lui-même)
  - ❑ Approche qui peut être pertinente sur des marchés électroniques notamment dans le cadre de « produits complexes »
  - ❑ Importance des activités de [recherche d'information](#)



## Tendance et thèmes de recherche

### ■ Web socio-sémantique selon Manuel Zacklad (UTT)



## Tendance et thèmes de recherche

### ■ Web socio-sémantique selon Manuel Zacklad (UTT)

#### □ Questionnement

- L'utilisation d'ontologies sémiotiques demande un « investissement définitionnel » important. Y a-t-il un retour sur cet investissement ? (par rapport à un moteur)
- Quels types de représentations graphiques sont les plus adaptées pour la navigation?
- Quels types de processus cognitifs situés & distribués sont mobilisés pour la recherche de l'entité?
- Quels type de connaissances sont requises pour l'usage du système? Quels apprentissages se mettent en place?
- Comment faire gérer le système de classification par le collectif? Comment le faire évoluer par la délibération.


## Tendance et thèmes de recherche

- Thème de Web social : veille et surveillance
  - L'activité veille consiste à détecter une tendance mais aussi des exceptions (activités innovantes par exemple)
    - Web mining dirigé par ontologies d'usage et sociales
    - Modèle de calcul de la pertinence des exceptions
    - Etc.
  - L'activité de surveillance consiste à des contrôles d'un processus et de détection des anomalies. Cela s'applique dans le domaine de sécurité mais aussi

Master MBDS - Séminaire de réseaux sémantiques au Web sémantique - 115

## Tendance et thèmes de recherche

- Thème de Web social : veille et surveillance : Capture et annotation d'émotions et projet EMOTICA  
«**Emotion Modelling by Ontology Techniques for Interactive Capture mechanisms with Automatic learning**»
- Questionnement
  - Est-ce possible de percevoir les traits émotionnels humains à partir des paramètres physiologiques ?
  - Est-ce possible de déterminer un ensemble des paramètres biométriques plus ou moins simples à observer (c-à-d captifs et mesurables par les dispositifs simples et économiques) permettant de percevoir avec une certaine précision les états émotionnels humains ?
  - Est-ce possible d'annoter les états émotionnels par les traits d'usage et sociaux ?
  - Enfin, peut-on modéliser ces notions afin de pouvoir les utiliser algorithmiquement dans les systèmes à comportement adaptatif ?



A cartoon illustration of a man in a purple suit and tie, sitting at a desk. He has a thought bubble above his head that says "MERCI!". A speech bubble coming from his mouth says "Et demain ... ... c'est un autre jour ...".

Master MBDS      Séminaire: des réseaux sémantiques au Web sémantique      N. Le Thanh      117

## VI. Références

- [Baa03] F. Baader, D. Calvanese, D. McGuinness, D. Nardi and P.F. Patel-Schneider, « *The Description Logic Handbook: Theory, Implementation and Applications* » Cambridge University Press, 2003
- [Bor03] Borgida A., Serafini L. "Distributed Description Logics : Assimilating Information from Peer Sources" *Journal of Data Semantics* (1). 2003, pp.153-184
- [Don97] F.M. Donini, M. Lenzerini, D. Nardi, W. Nutt "*The Complexity of Concept Languages*" *Information and Computation*, 134, pp 1-58, 1997
- [Nap97] Amedeo Napoli, « *Une introduction aux logiques de description* " Rapport de recherche INRIA n°3314, décembre, 1997