

Ingénierie ontologique

Yannick Prié

UFR Informatique

Université Claude Bernard Lyon 1

Objectifs

- Comprendre ce qu'est une ontologie formelle, à quoi ça sert
- Découvrir deux principaux langage de représentation de connaissances ontologiques : graphes conceptuels, logiques de description
- Découvrir OWL (Ontology Web Language) dans le contexte du Web sémantique

Plan

- Représentation de connaissances
- Ingénierie des connaissances
- Ontologies
- Graphes conceptuels
- Logiques de description
- Introduction au Web sémantique
- OWL
- Conception d'ontologies

Connaissance ?

- Une connaissance se manifeste en acte, est liée à une action
 - elle est révélée par une capacité à agir dans un monde
- Au niveau Intelligence Artificielle
 - Connaissance = information (donnée) utilisée dans une inférence ou un raisonnement
 - Une inscription qui permet une inférence est une inscription de connaissances

Représentation des connaissances

- Approche d'inscription symbolique
- Approche de modélisation et de formalisation
 - Modéliser pour éliciter des connaissances
 - construire des modèles de connaissances
 - Formaliser pour pouvoir
 - représenter des connaissances en machine
 - raisonner automatiquement dessus
- Représentation de connaissances
 - Inscription de connaissances formelles à l'aide d'un langage de représentation de connaissances possédant une sémantique formelle permettant de faire des inférences

Langages formels

- Niveau syntaxique
 - alphabet
 - procédé de construction
 - axiomes = expressions de base
 - règles de dérivation pour produire des théorèmes à partir d'autres théorèmes
 - théorème = expression du système
 - les axiomes sont des théorèmes
- Niveau sémantique
 - Interprétation des expressions dans un monde, par exemple {Vrai, Faux}
- Système correct
 - Tous les théorèmes sont interprétés comme *Vrai*
- Système complet
 - Toutes les expressions interprétées à Vrai sont des théorèmes

Exemple : le système "peu"

- alphabet = l'ensemble des trois symboles "p" , "e" , et "u"
- procédé = concaténation
- axiome = upueuu
- règles :
 - R1 : si une expression de la forme AeB est un théorème (où "A" désigne n'importe quelle suite de "u", de "p", et de "e", et B de même), alors l'expression uAeBu est aussi un théorème
 - R2 : si une expression de la forme AeB est un théorème, alors l'expression AueuB est aussi un théorème
- Questions
 - Q1 = uupuueuuuu est-il un théorème?
 - Q2 = upuueuuuu ?
 - Q3 = upupueuuu ?
- Mécanismes pour décider
 - Si une expression est un théorème : exploration systématique
 - Si une expression n'est pas un théorème : pas de procédure

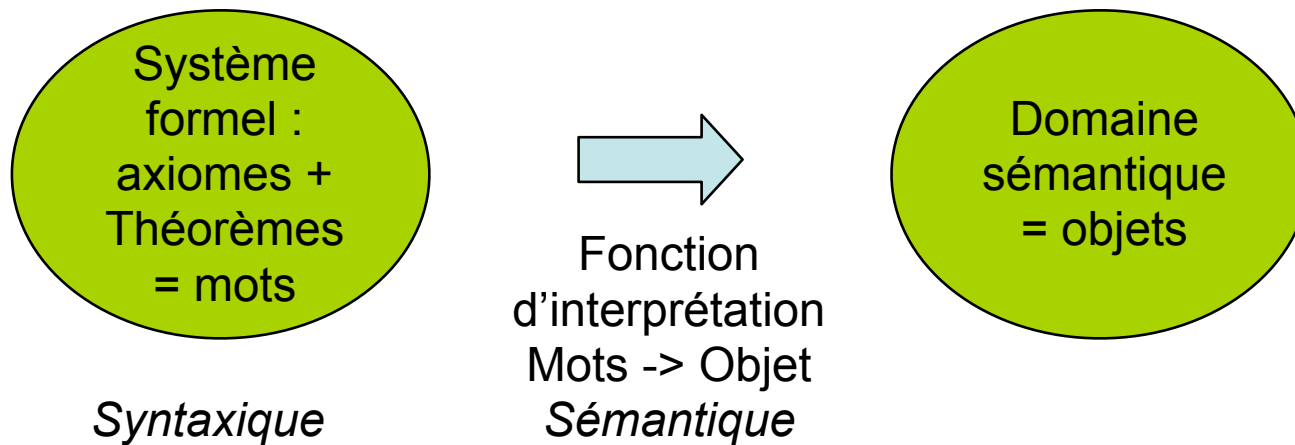
Autre exemple : système « plus égal un »

- alphabet = l'ensemble des trois symboles "plus " , "égal " , et "un "
- procédé = concaténation
- axiome = un plus un égal un un
- règles :
 - R1 : si une expression de la forme *A égal B* est un théorème (où "A" désigne n'importe quelle suite de "un " , de "plus " , et de "égal " , et B de même), alors l'expression *un A égal B un* est aussi un théorème
 - R2 : si une expression de la forme *A égal B* est un théorème, alors l'expression *A un égal un B* est aussi un théorème
- Questions :
 - Q1 = un un plus un un égal un un un un est-il un théorème?
 - Q2 = un plus un un égal un un un un ?
 - Q3 = un plus un plus un égal un un un ?
- Isomorphisme entre les deux systèmes

Interprétation

- Relire avec : « plus » = « + », « égal » = « = », « un » = « 1 »
- L'axiome se lit : $1+1=11=2$ (cf. chiffres romains, concaténation = addition)
- Les règles se lisent :
 - R1 : si $A=B$, alors $1+A=B+1$
 - R2 : si $A=B$, alors $A+1=1+B$
- Les questions donnent
 - Q1 : $2+2=1111=4$ est un théorème
 - Q2 : $1+2=1111=4$ est un non-théorème
 - Q3 : $1+1+1=111=3$ est un non-théorème
- Pourtant dans le monde de l'arithmétique où on s'est placé
 - Q1 et Q3 : vrai
 - Q2 : faux
 - $1+1+1=3$: vrai, mais n'est pas un théorème !
- Deux mondes distincts
 - le « monde des théorèmes » : syntaxe
 - le « monde de la vérité » : sémantique

Interprétation – sémantique



- Possibilité de notion de vérité dans le domaine sémantique = possibilité de « bonne interprétation »
 - Idéalement théorème \Rightarrow interprétation vraie

Syntaxe / sémantique / inférence

- Syntaxe
 - Théorie des langages
 - Langage
- Sémantique
 - Théorie des modèles
 - Conséquence (\models)
- Inférence
 - Théorie de la preuve
 - Démonstration (\vdash)

Exemple : système ED (1)

- Grammaire (BNF):
 - énoncé := phrase ["," énoncé]
 - phrase := nom opérateur nom
 - opérateur := "E" | "D"
 - nom := ("a".."z" | "_")+
- Exemple d'énoncé :
 - a E b , b E c , c D d

Interprétation et modèle

- Une **interprétation** \mathfrak{I} est définie par
 - un domaine d'interprétation $\Delta^{\mathfrak{I}}$ (ensemble non vide)
 - une fonction d'interprétation $\cdot^{\mathfrak{I}}$ qui associe les structures syntaxiques du langage au domaine
- Un énoncé ε du langage est **vérifié** par \mathfrak{I} si ses structures vérifient des propriétés définies par la sémantique du langage
 - on note : $\models_{\mathfrak{I}} \varepsilon$
 - \mathfrak{I} est appelée un **modèle** de ε

Exemple : système ED (2)

- pour tout nom n , $n^{\mathfrak{S}} \in \Delta^{\mathfrak{S}}$
- *pour tout énoncé \mathcal{E} , \mathfrak{S} vérifie \mathcal{E} si elle vérifie chaque phrase de \mathcal{E}*
- $\vDash_{\mathfrak{S}} x E y$ si et seulement si $x^{\mathfrak{S}} = y^{\mathfrak{S}}$
- $\vDash_{\mathfrak{S}} x D y$ si et seulement si $x^{\mathfrak{S}} \neq y^{\mathfrak{S}}$




Exemple : système ED (3)

- $\mathcal{E} = a \in b, b \in c, c \in d$
- $\mathfrak{S}1$ est un modèle de \mathcal{E} avec
 - $\Delta^{\mathfrak{S}1} = \mathbb{N}$
 - $a^{\mathfrak{S}1} = b^{\mathfrak{S}1} = c^{\mathfrak{S}1} = 1, d^{\mathfrak{S}1} = 2$
- $\mathfrak{S}2$ est un modèle de \mathcal{E} avec
 - $\Delta^{\mathfrak{S}2} = \mathbb{R}$
 - $a^{\mathfrak{S}2} = b^{\mathfrak{S}2} = c^{\mathfrak{S}2} = \pi, d^{\mathfrak{S}2} = \sqrt{2}$
- $\mathcal{E}' = a \in b, b \in c, c \in a$
n'admet aucun modèle

Conséquence

- Un énoncé \mathcal{E}' est une *conséquence* d'un autre énoncé \mathcal{E} si et seulement si tout modèle de \mathcal{E} est aussi un modèle de \mathcal{E}'
- On le note : $\mathcal{E} \models \mathcal{E}'$

Exemple : système ED (4)

- \mathcal{E} = clark_kent E superman,
lois_lane D clark_kent,
lex_luthor D superman
- $\mathfrak{S}1$ est un modèle de \mathcal{E} avec
 - clark_kent $^{\mathfrak{S}1}$ = superman $^{\mathfrak{S}1}$ = 
 - lois_lane $^{\mathfrak{S}1}$ =  lex_luthor $^{\mathfrak{S}1}$ = 
- $\mathfrak{S}2$ est un modèle de \mathcal{E} avec
 - $\Delta^{\mathfrak{S}2} = \mathbb{R}$
 - clark_kent $^{\mathfrak{S}2}$ = superman $^{\mathfrak{S}2}$ = π ,
lois_lane $^{\mathfrak{S}2}$ = $\sqrt{2}$, lex_luthor $^{\mathfrak{S}2}$ = $\sqrt{3}$

Exemple : système ED (4')

- \mathfrak{S} est aussi un modèle de \mathcal{E} avec

– $\text{clark_kent}^{\mathfrak{S}} = \text{superman}^{\mathfrak{S}} =$



$\text{lois_lane}^{\mathfrak{S}} = \text{lex_luthor}^{\mathfrak{S}} =$



- $\mathcal{E} \not\models \text{lois_lane} \text{ N } \text{lex_luthor}$
- \mathcal{E} n'implique pas non plus que
 - *lois_lane soit une femme ou que clark_kent soit un homme*
 - *lois_lane aime clark_kent*
 - *lois_lane aime superman*

Limitations

- La sémantique formelle d'un langage a vocation à **capturer** une partie de la sémantique (humaine) de ce qui est énoncé dans ce langage
- Dépend
 - de ce qu'on a pensé à mettre dans l'énoncé
 - de l'expressivité du langage (ce qu'on a *pu mettre dans l'énoncé*)


Expressivité

- Plus un langage est expressif, plus il sera possible de capturer de sémantique dans sa sémantique formelle...
- ... mais plus les algorithmes qui calculeront les conséquences d'un énoncé seront **complexes**
Rappel : la logique du premier ordre est très expressive, mais le problème de la conséquence est **indécidable**

Représentation de connaissances

- Système définissant une série de symboles et une série d'opérations sur ces symboles + interprétation
 - En tant que représentations, les symboles ont un contenu sémantique mais les opérations qu'on leur associe ne sont déterminées que par leur syntaxe
 - Les inférences sont des déductions formelles donnant de nouvelles représentations
- Enjeu de la représentation des connaissances :
 - Que les représentations données soient significatives pour un humain
 - Que les représentations déduites soient toujours significatives pour un humain !

Réseaux sémantiques

- Au delà *des* logiques
 - différemment expressives et décidables
- Représentation graphique : graphes


The diagram illustrates a graph with two nodes, A and B, represented as green circles. A horizontal arrow labeled 'R' points from node A to node B. To the right of this arrow is a double-headed arrow (↔) pointing to the text 'R(A,B)', indicating that the relationship R between A and B is represented by the expression R(A,B).
- Différentes relations, certaines canoniques, liées aux notions de classes et d'individus
 - AKO (A Kind Of) : $B(A)$: A appartient à la classe B
 - IsA : $B(A)$: A est une classe sorte_de classe B
- Exemples
 - Le Chat est un Félin, Félix est un Chat, Le Chat n'aime pas le Chien, Le Chat a des Griffes
- On peut se ramener à la logique (classique), ou utiliser des opérations de graphe (propagation de marqueurs)

Graphes conceptuels

- Support
 - Hiérarchie de concepts + hiérarchie de relations
- Exemples de graphes
 - [Bird]<-(Agnt)<-[Sing]
 - "A bird is singing"
 - [Person: John]<-(Agnt)<-[Go]->(Dest)->[City: Aalborg]
 - "John is going to Aalborg »
- Raisonnement
 - Projection entre deux graphes
 - si l'un est plus général que l'autre
- Interprétation logique

Logiques de description

- Concepts et individus
- Rôle = relation binaire entre individus
- Concepts
 - primitifs (donnés)
 - définis (par des rôles et des restrictions de rôles)
- Raisonnement = subsomption entre les concepts
 - Un concept est plus général qu'un autre
 - Permet la classification
- Interprétation ensembliste
- Famille de langages
 - expressivité / complétude et calculabilité

Plan

- Représentation de connaissances
- **Ingénierie des connaissances**
- Ontologies
- Graphes conceptuels
- Logiques de description
- Introduction au Web sémantique
- OWL
- Conception d'ontologies

Différentes connaissances

- Connaissances stratégiques, savoirs-faire
 - Comment une tâche va être effectuée
- Connaissance du domaine
 - Concepts manipulés et leur relations. Description du domaine
- Modèle conceptuel
 - Connaissances stratégiques + du domaine
 - Abstraction permettant d'opérationnaliser une expertise de résolution de problème, une partie d'un monde...

Systemes à base de connaissances

- Tout système mobilisant explicitement des modèles de connaissances
 - Manipulation d'expressions symboliques (ou inscriptions), règles de réécriture symboliques
 - L'utilisateur utilise (en les interprétant) les inscriptions dans le cadre d'une tâche
 - SBC = « Systèmes sémiotiques de manipulation d'inscriptions symboliques dont le fonctionnement informatique doit permettre à l'utilisateur d'interpréter et de comprendre le système dans le cadre de son activité, en utilisant les termes du domaine »
- Inscription de connaissances dans un support formel
 - Modèle d'inscription
 - Raisonnements disponibles
 - Classes d'applications (objectifs)
 - Méthodologies
 - Pour définir les connaissances (abstraites ou instances)
 - Pour choisir des langages
 - Pour faire des systèmes

Ingénierie des connaissances

- SBC = systèmes techniques plongés dans des systèmes d'usage
- « L'ingénierie des connaissances correspond à l'étude de modèles symboliques formels plongés dans des systèmes d'usage : c'est l'ingénierie informatique et logique des modèles en fonction des usages qu'ils rendent possibles et des appropriations qu'ils permettent »

Concevoir des SBC

- Trouver les expressions linguistiques permettant d'explicitier le problème à traiter
- Proposer un modèle conceptuel au « niveau des connaissances » permettant de décrire les objets et les raisonnements
- Proposer une opérationnalisation en machine des connaissances, et un système autour qui utilise cette dernière

Questions de l'IC

- Quels modèles pour les bases de connaissances ?
 - Type de modèle (domaine, raisonnement, etc.)
- Quels langages et formalismes pour décrire les modèles ?
 - Langages de modélisation formelle : logique, LD, GC...
- Comment construire les modèles nécessaires à une application, suivant quelles étapes, avec quels outils ?
 - Méthodes et techniques : KADS...
- Comment construire une application à base de connaissances liée à un modèle ?
 - Navigation, inférences, explications
- Comment faire évoluer un modèle ?
 - Maintenance et gestion
- ...

Domaines associés

- Linguistique
 - Traitement Automatique des Langues, terminologie
- Logique
 - Raisonnement, logiques formelles
- Ergonomie
 - IHM
- Epistémologie
 - Connaissances et inscriptions
- Anthropologie, sociologie
 - Usages, partage
- Sciences de gestion
 - Gestion des connaissances
- Génie logiciel
- ...

Systemes à base de connaissances

- Inscription de connaissances dans un support formel
 - Modèle d'inscription
 - Raisonnements disponibles
 - Classes d'applications (objectifs)
 - Méthodologies
 - Pour définir les connaissances (abstraites ou instances)
 - Pour choisir des langages
 - Pour faire des systèmes

Plan

- Représentation de connaissances
- Ingénierie des connaissances
- **Ontologies**
- Graphes conceptuels
- Logiques de description
- Introduction au Web sémantique
- OWL
- Conception d'ontologies

Partie à refaire

- Car elle est un peu bâtarde (tout de qui vient de C. Reynaud)
- Mixer aussi avec ce qui vient du cours intro Traces (transparents suivants)

Ontologies ?

- Définitions
 - « Étude de ce qui est » (philosophie)
 - « Explicit, formal specification of a shared conceptualisation. » (Gruber 95)
- Objectifs
 - Se mettre d'accord sur le sens des termes employés dans une organisation, une communauté, un métier
 - Faire en sorte que les personnes et les logiciels se comprennent

Passer des termes aux concepts

- “chambre” ?
 - chambre d'hôtel, chambre d'écho, chambre des députés, chambre d'enregistrement, chambre noire, chambre funéraire ?
- Terme
 - Mot ou groupe de mots désignant (dénotant) des concepts
 - « cat », « chat », « matou » désignent le concept de *chat*
 - Terme = signifié normé (Rastier)
 - Synonymie
 - plusieurs termes dénotent le même concept
 - Ambiguïté
 - plusieurs concepts dénotés par le même terme

Pour quoi faire ?

- Un système d'indexation et de recherche d'information
- Un système d'annotation
- Une base de connaissances à objets
- Pour
 - bibliothèques numériques
 - commerce électronique
 - systèmes à base de connaissances

A partir de

<http://www-sop.inria.fr/acacia/cours/essi/ontologie.ppt>

Différentes acceptions du mot ontologie

- Du vocabulaire à l'ontologie formelle
 - Vocabulaire technique
 - Référentiel métier
 - Terminologie/thésaurus
 - Système de classes d'une représentation par objet : UML ?
 - Base de connaissances terminologique
- Plus ou moins
 - de formalisation
 - de possibilités inférentielles
 - (... de relations – supposées – fixes terme / concept...)

A partir de

<http://www-sop.inria.fr/acacia/cours/essi/ontologie.ppt>

Conception

- Identifier/modéliser les concepts et les termes pertinents
- Identifier les relations pertinentes : subClassOf, isa, partOf, hasPart, closeTo, over, under, contains, connected, etc.
- Règles pour combiner les concepts et les relations : partOf est transitive

Connaissances modélisées

- Connaissances de résolution de problème :
 - conception, diagnostic, évaluation, planification
 - tâches, inférences
- Connaissances du domaine
 - e.g. électronique, mécanique, médecine, etc.
 - ontologie = objets : réutilisable ?
 - modèles du domaine : fonctionnel, causal, structurel,
 - exprimés dans le vocabulaire de l'ontologie
 - réutilisables ?
- Remarque :
 - approche Common-Kads
 - séparation pas forcément pertinente
 - on peut parler d'ontologie des tâches...

A partir de

<http://www-sop.inria.fr/acacia/cours/essi/ontologie.ppt>

Ontologie

concept Object

concept Engine < Object

concept Vehicle < Object

relation partOf :

Object -> Object

A partir de

<http://www-sop.inria.fr/acacia/cours/essi/ontologie.ppt>

Modèles de domaine

structurel :

Engine e1 partOf Vehicle v2

causal :

**Engine breakdown => Vehicle
stop**

A partir de

<http://www-sop.inria.fr/acacia/cours/essi/ontologie.ppt>

Modèles de tâche

Conception :

- specify, design, assess

Diagnostic :

- identifier des symptômes, choisir un modèle causal, émettre des hypothèses, imaginer des conséquences, les tester, etc.

Ontologies

- L'accent est mis sur les *concepts* et les *relations* du domaine
- Orienté modélisation, spécification, réutilisation, partage, standardisation.
- Relations spécifiques du domaine considéré

Terminologie/Thésaurus

- Terminologie
 - termes utilisés pour dénoter les concepts d'un domaine spécialisé (technologique)
 - relations entre termes sont linguistiques
 - BT (Broader Term), NT (Narrower Term), TT (Top Term), RT (Related Term - other than BT, NT, TT, etc.), USE (Preferred Term), UF (Use for, non preferred synonym, quasi synonym), SN (Scope Note, note pour expliquer un terme)
 - orienté traduction / indexation
- Thésaurus
 - recueil documentaire alphabétique de termes servant de descripteur pour
 - analyser un corpus
 - indexer des documents
 - Relations prédéfinies standardisées (spec, seeAlso...)

A partir de

<http://www-sop.inria.fr/acacia/cours/essi/ontologie.ppt>

Ontologies formelles (objectifs)

- Composants
 - réutilisables
 - par généralisation, abstraction
 - partagés
 - consensus, standardisation
- Accord sur la conceptualisation partagée
 - engagement ontologique (commitment)

Caractéristiques voulues

- Formalisation : lever les ambiguïtés, cohérence
- Indépendance (relative) par rapport à une tâche, ou un problème précis, modularité
- Utilisable pour différentes tâches : conception, diagnostic, maintenance, recherche d'information : extensibilité
- Indépendante d'une implémentation : limiter le biais dû à un formalisme de représentation : niveau conceptuel
- Possibilité d'inférences (relations transitives, axiomes, etc.)

A partir de

<http://www-sop.inria.fr/acacia/cours/essi/ontologie.ppt>

Types d'ontologies

- Ontologies générales, abstraites, de haut niveau
 - Ontologie des catégories conceptuelles :
 - objet, événement, état, processus, action, temps, espace
 - SUMO
- Ontologies théoriques
 - physique, mathématique, cinématique
- Ontologie applicatives
 - Médecine, automobile, patrimoine culturel, organisation, etc.

A partir de

<http://www-sop.inria.fr/acacia/cours/essi/ontologie.ppt>

Outils/Langages

- KIF
 - Knowledge Interchange Format
- Logiques de description
 - orientées classification
- Graphes conceptuels
 - support
- RDF Schema
- DAML+OIL, OWL

A partir de

<http://www-sop.inria.fr/acacia/cours/essi/ontologie.ppt>

Exemples

- WordNet
 - ontologie linguistique : anglais
- Enterprise Ontology
- Kactus
 - ingénierie
- UMLS
 - Unified Medical Language System
- CYC
- EngMath, PhysSys
- ...

A partir de

<http://www-sop.inria.fr/acacia/cours/essi/ontologie.ppt>

Discussion

- Haut-niveau / bas-niveau
- Réutilisation ?
- Conception ?
- Web sémantique (cf. plus loin)
- ...

- [[Retour](#)]

A partir de

<http://www-sop.inria.fr/acacia/cours/essi/ontologie.ppt>

Ne pas confondre...

- *L'Ontologie* [philosophie]
 - Partie de la métaphysique qui s'applique à « l'être en tant qu'être » (Aristote), indépendamment de ses déterminations particulières (*sic* !) (Petit Robert)
- *Une ontologie* [ingénierie des connaissances]
 - Spécification explicite d'une conceptualisation (Gruber)

Histoire de définitions

- Une des premières définitions :
 - « An ontology defines the basic terms and relations comprising the vocabulary of a topic area as well as the rules for combining terms and relations to define extensions to the vocabulary » [Neches et al. 91]
- Suivie par une définition plus large :
 - « an ontology is an explicit specification of a conceptualization » [Gruber 93]
- Définition de Gruber plus élaborée :
 - « Ontologies are defined as a formal specification of a shared conceptualization » [Borst et al. 97]
 - « Conceptualization refers to an abstract model of some phenomenon in the world by having identified the relevant concepts of that phenomenon. Explicit means that the type of concepts used, and the constraints on their use are explicitly defined. Formal refers to the fact that the ontology should be machine-readable. Shared refers to the notion that an ontology captures consensual knowledge, that is, is not primitive to some individual, but accepted by a group. [Studer et al. 98]

Histoire de définitions (2)

- Autre définition
 - « An ontology may take a variety of forms, but necessarily it will include a vocabulary of terms, and some specification of their meaning. This includes definitions and an indication of how concepts are inter-related which collectively impose a structure on the domain and constrain the possible interpretation of terms. An ontology is virtually always the manifestation of a shared understanding of a domain that is agreed between a number of agents. Such agreement facilitates accurate and effective communication of meaning, which in turn leads to other benefits such as inter-operability, reuse and sharing ». [Uschold 98]
- Définition plus large
 - « We consider ontologies to be domain theories that specify a domain-specific vocabulary of entities, classes, properties, predicates, and functions, and to be a set of relationships that necessarily hold among those vocabulary terms. Ontologies provide a vocabulary for representing knowledge about a domain and for describing specific situations in a domain » [Fikes et al. 99]
- La littérature est pleine de définitions différentes du terme ontologie. Chaque communauté adopte sa propre interprétation selon l'usage qui en est fait et le but visé.

À quoi servent les ontologies ?

- Un produit d'ingénierie :
 - composé d'un certain vocabulaire utilisé pour décrire une certaine réalité, et
 - d'un ensemble d'hypothèses explicites sur le sens des termes.
- comme outil de communication
 - entre humains
 - entre humain et machine
 - entre machines
- Donc, une ontologie décrit une spécification formelle d'un certain domaine
 - une compréhension partagée d'un domaine d'intérêt
 - un modèle formel et manipulable par des machines d'un domaine d'intérêt
 - langages munis d'une sémantique formelle (deux derniers points)

Bref

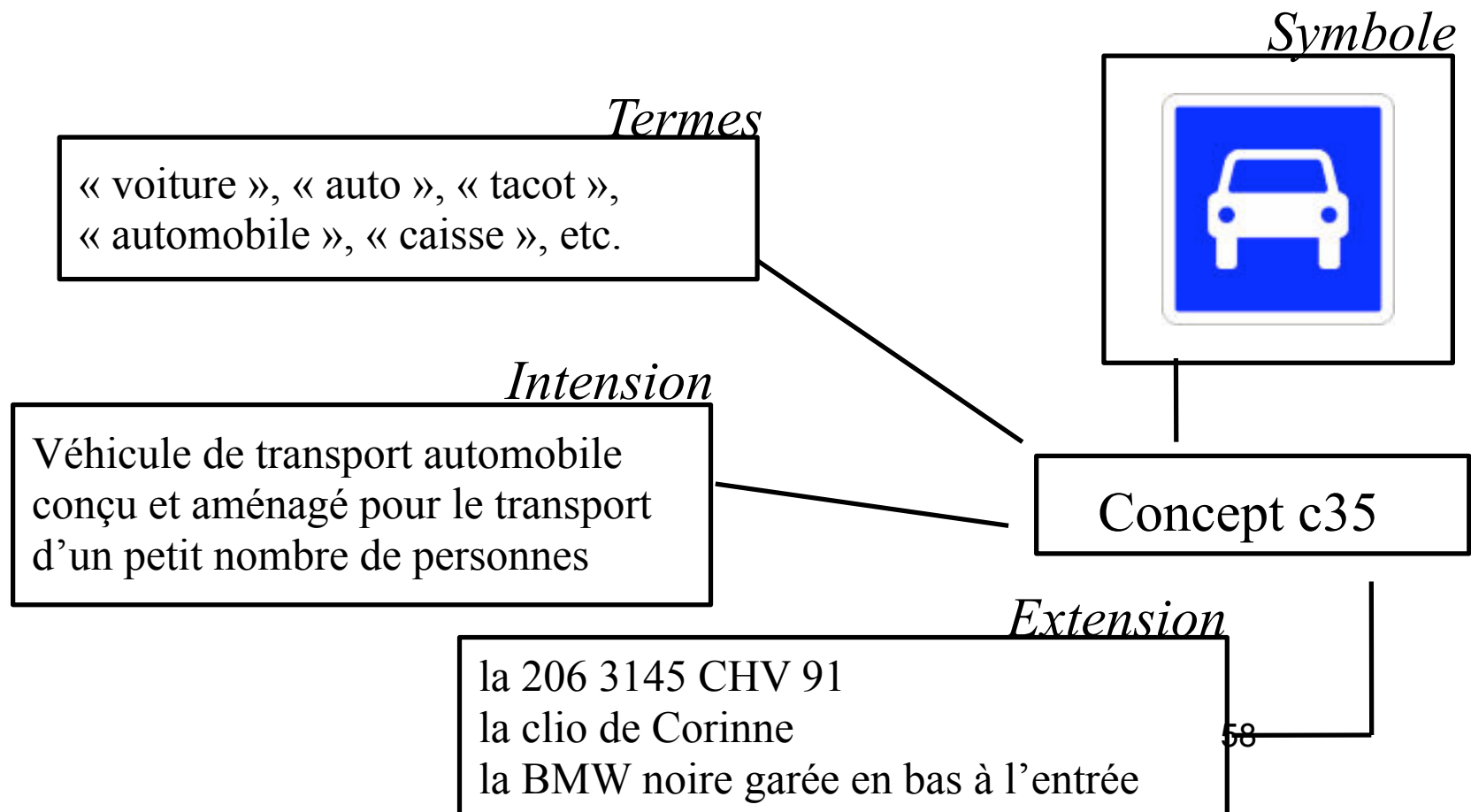
- Description **explicite** et **consensuelle** d'un **domaine d'intérêt**,
- généralement comme un ensemble d'**objets** et de **relations** entre eux.
- Exemples
 - ontologie de la salle de classe, ontologie de la voiture
- Remarque
 - Un rôle analogue à celui d'un schéma de base de données, mais
 - Un langage de spécification d'ontologies est syntaxiquement et sémantiquement plus riche qu'un langage de base de données
 - Une ontologie doit être une terminologie consensuelle et partagée (partage et échange d'information)
 - Une ontologie fournit une théorie du domaine et pas seulement la structure du « container » des données

Structure d'une ontologie

- Les ontologies se composent de deux éléments bien distincts :
- Les concepts importants d'un domaine
 - Eléphant est un concept dont les membres sont une sorte d'animal
 - Herbivore est un concept dont les membres sont des animaux qui ne mangent que des plantes ou des parties de plantes
 - Elephant_Adulte est un concept dont les membres sont des Eléphants ayant plus de 20 ans
- Connaissances de base / contraintes sur le domaine
 - Un Eléphant_Adulte pèse au moins 2 tonnes
 - Tous les Eléphants sont soit des Elephants_Afriques soit des Elephants_Indes
 - Aucun individu n'est à la fois Herbivore et Carnivore.

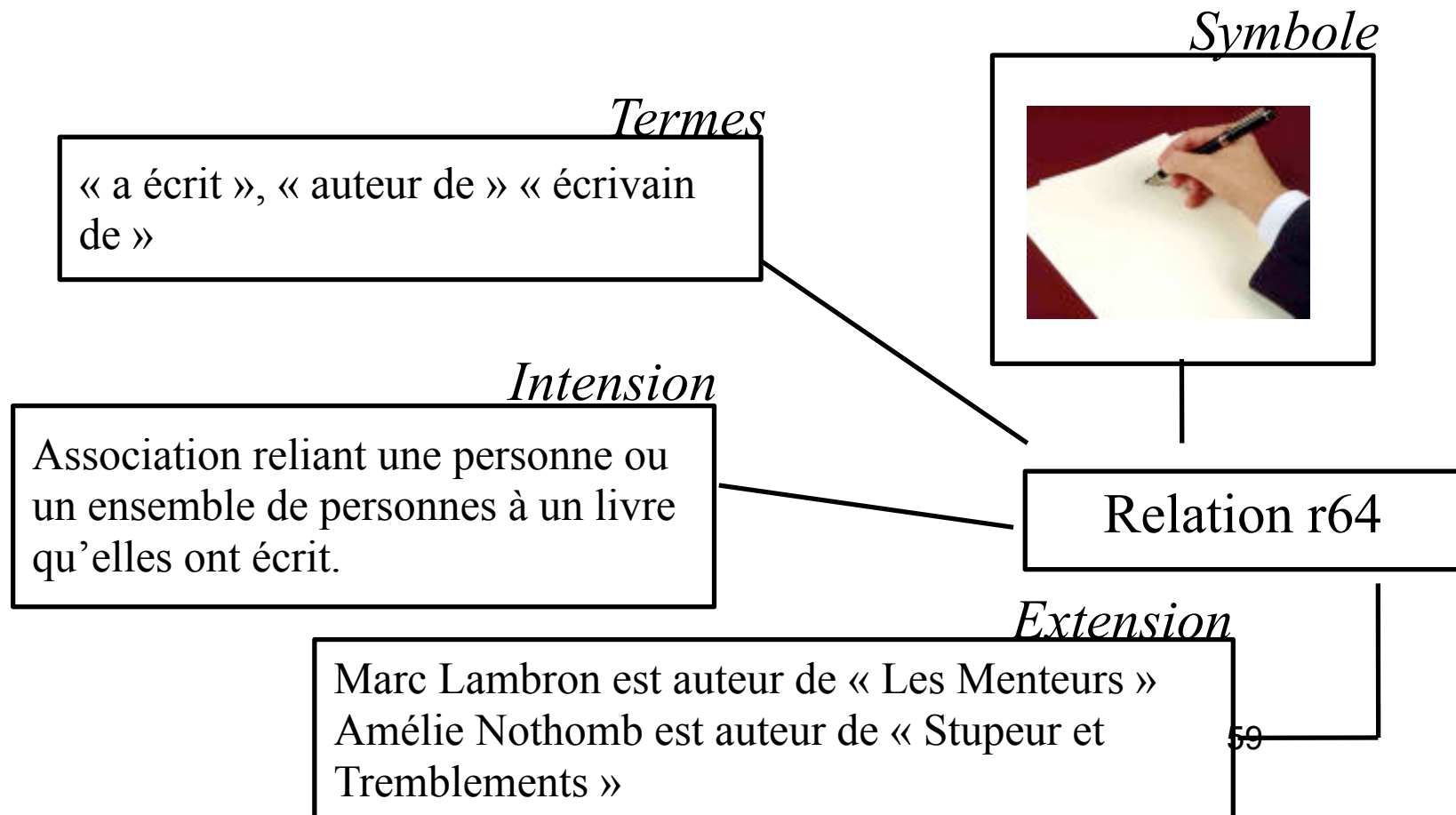
Les composants d'une ontologie

- Concepts / Classes
 - Des notions exprimées en général par un terme ou par un symbole littéral ou autre. Un concept représente un ensemble d'objets et leurs propriétés communes.



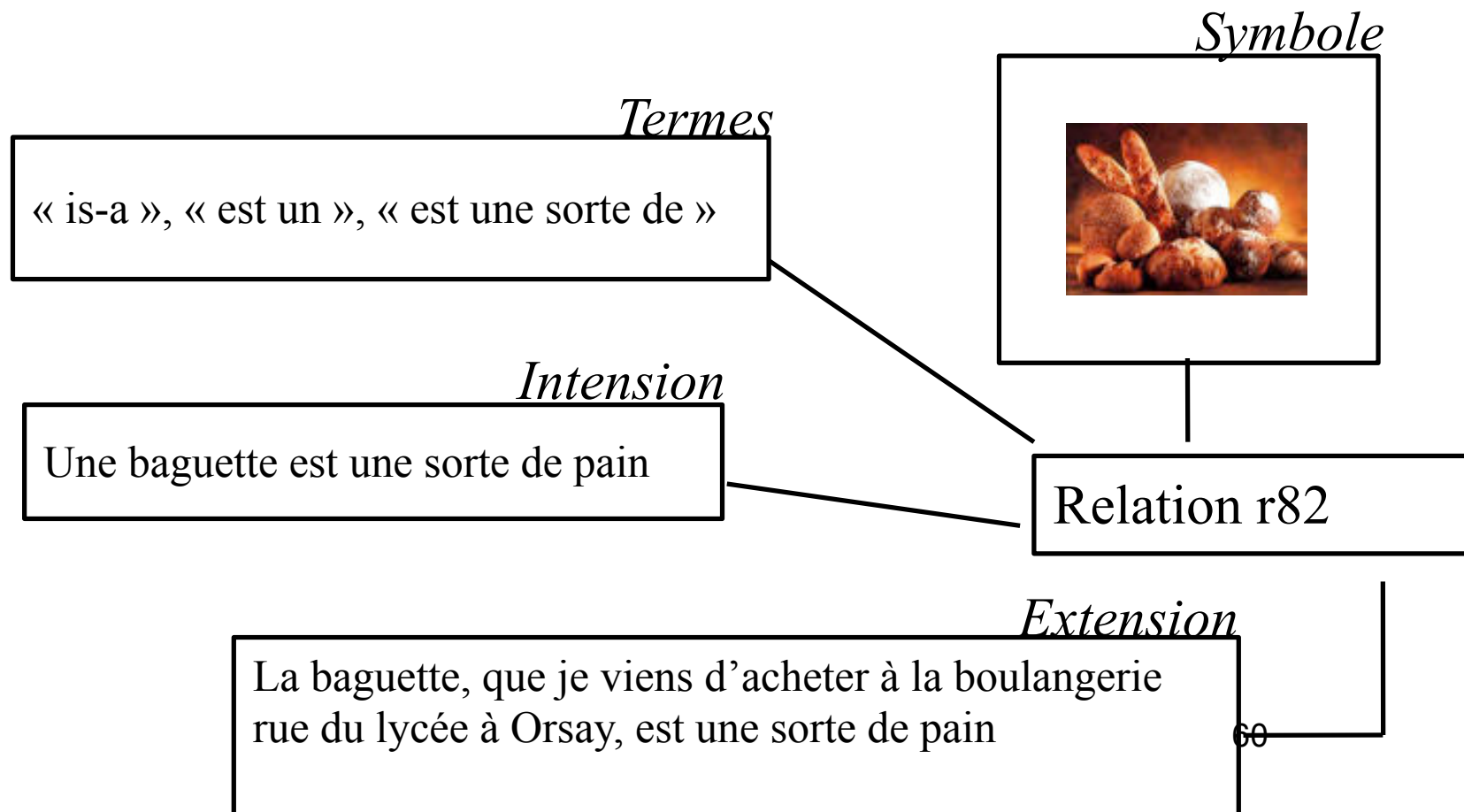
Les composants d'une ontologie

- Relations
 - Notions d'association s'établissant entre des concepts, exprimées en général par un terme ou par un symbole littéral ou autre.



Les composants d'une ontologie

- Relations
 - Relation binaire de généralisation entre un concept général et une version plus spécifique de ce concept.





Pains spéciaux

Pains régionaux



Pain complet



Pain aux
lardons

Pain blanc



Pain de seigle

Pain au son

Pain aux noix et aux raisins

Pain auvergnat

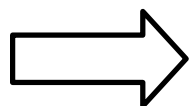
Couronne



Pain marguerite

Lyonnaise

Pain vaudois

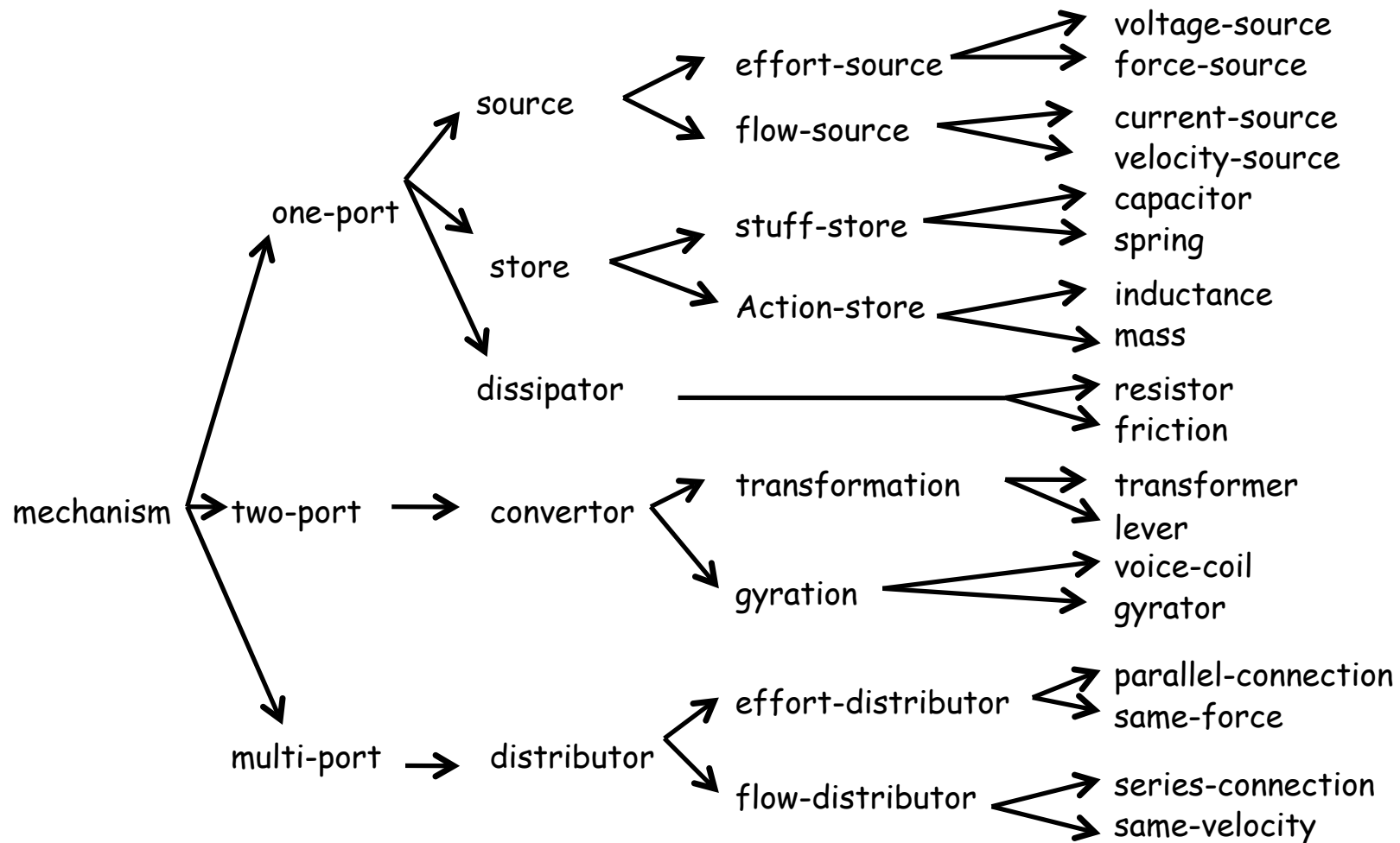


Structure taxinomique

61

44

Autre exemple : taxinomie de mécanismes physiques (Borst)



Connectivity

Type

Effort/flow

Domain (e.g. el, me)⁶²


Taxinomies et ontologies

- Taxinomie/Taxonomie : classification d'éléments (Petit Robert)
- Naturelle pour l'homme qui fonctionne souvent par abstraction et association
- Structure à la base de deux inférences élémentaires que nous faisons tous les jours
 - l'identification : capacité à reconnaître la classe d'un objet à partir de ses caractéristiques
 - la spécialisation : capacité à prendre en compte des catégories de niveaux de précision variables
- Les taxinomies sont des structures très courantes au sein des ontologies. Les concepts d'une ontologie sont souvent organisés en taxinomies.

Les composants d'une ontologie

- Fonctions
 - Relations n-aires particulières : le nième argument dépend des (n-1) précédents.
 - $F: A1 \times A2 \times \dots \times A_{n-1} \rightarrow A_n$
- Exemples :
 - Prix d'une voiture d'occasion, fonction :
 - du modèle
 - de l'année de la voiture
 - du kilométrage

Les composants d'une ontologie

- Axiomes
 - Des expressions toujours vraies.
 - Exemples :
 - Il n'y a pas de vols directs entre deux pays en conflit
 - Une cabine de bateau est une sorte de chambre. Elle en a toutes les caractéristiques sauf le fait d'avoir une terrasse.
 - Il est possible de pratiquer la natation dans un lieu montagneux à condition qu'il y ait une piscine.
 - Ne peuvent plus se représenter sous forme de relations
-  Besoin d'autre formalismes

Les composants d'une ontologie

- Instances
 - Des éléments
- Exemples :
 - La clio de Corinne
 - Marc Lambron est auteur de « Les menteurs »
 - La baguette, que je viens d'acheter à la boulangerie rue du lycée à Orsay, est une sorte de pain

Degrés de formalité

- Du plus informel au plus formel
 - Tags
 - Listes d'autorité
 - Terminologies
 - Thesaurii
 - Langages à facettes
 - Taxonomies
 - Ontologies formelles
 - Différentes expressivités, applications, utilisations
- } Ontologies ?

Engagement ontologique

- Accord sur ce qui est exprimé par l'ontologie qu'on met en place
 - Au niveau du cadre
 - vocabulaire (e.g. concepts, quantificateurs...)
 - Au niveau signification / domaine
 - modélisation, signification capturée par l'ontologie

Ontologie universelle ?

- Tout décrire dans une ontologie unique
 - Interopérabilité générale
- Une tentative : CYC
 - Dizaines de milliers de concepts et de faits
 - A peu près ingérable
- Plus généralement
 - Impossible en théorie
 - l'humain est toujours là pour utiliser au final
 - ce qu'est le monde ne cesse de varier
 - On préfère construire des ontologies liées à des domaines plus restreints
 - Certains cherchent à définir des ontologies de haut-niveau (universelles) à spécialiser

Différents types d'ontologies

- Ontologies de représentation
 - définit un ensemble de primitives de représentation
 - exemple : la définition des éléments de RDF(S) : classe, propriétés, relations de sous-classe, de sous-propriété, etc.
- Ontologies génériques
 - exemples : ontologies spatiale, temporelle
- Ontologies d'un domaine
 - le plus classique
- Ontologies d'une tâche / méthode
 - le rôle joué par chaque concept dans une méthode particulière est rendu explicite
- Ontologies d'application
 - peuvent être vues comme une double spécialisation : d'une ontologie du domaine et d'une ontologie de méthode

Diversité des ontologies

- Ontologie d'un domaine
 - Application précise
 - Energies marines (EDF), Textes juridiques (Mondeca pour Lexis-Juris), Menelas plus général...
 - Famille d'application
 - Documents audiovisuels (INA), STEP (conception), TOVE (modélisation de l'entreprise)...
 - Communauté « mondiale » d'utilisateurs qui peut être très large
 - CRM (Musées), Gene ontology (GO)...
- Autres distinctions
 - Théorique versus pragmatique
 - Descriptive ou « catégoriale » (Poli 2002)
 - Conditions de construction et d'évolution
 - Expressivité
 - Niveau de formalisation
 - Fondations logiques, capacités inférentielles

Autres exemples

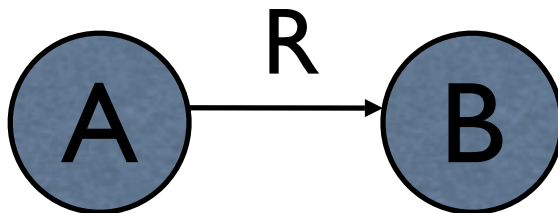
- Cf. Annexe

Plan

- Représentation de connaissances
- Ingénierie des connaissances
- Ontologies
- **Graphes conceptuels**
- Logiques de description
- Introduction au Web sémantique
- OWL
- Conception d'ontologies

Réseaux sémantiques

- Un réseau sémantique est un graphe orienté et étiqueté (un multi-graphe en fait car rien n'empêche deux nœuds d'être reliés par plusieurs arcs).
- Une « sémantique » (au sens de la logique) est associée par le biais des relations.
- Réseau = conjonction de formules logiques associées à chacun des arcs



74

$R(A, B)$
Prédicat binaire

Sémantique de quelques relations « privilégiées »

- Appartenance d'individus à une classe
d'individus AKO (A kind of) \Leftrightarrow appartenance
 - $B(A)$ /* A appartient à la classe B */
- Relation de spécialisation (IsA). \Leftrightarrow sorte_de
 - $B(A)$ /* A est une classe sorte_de classe B*/

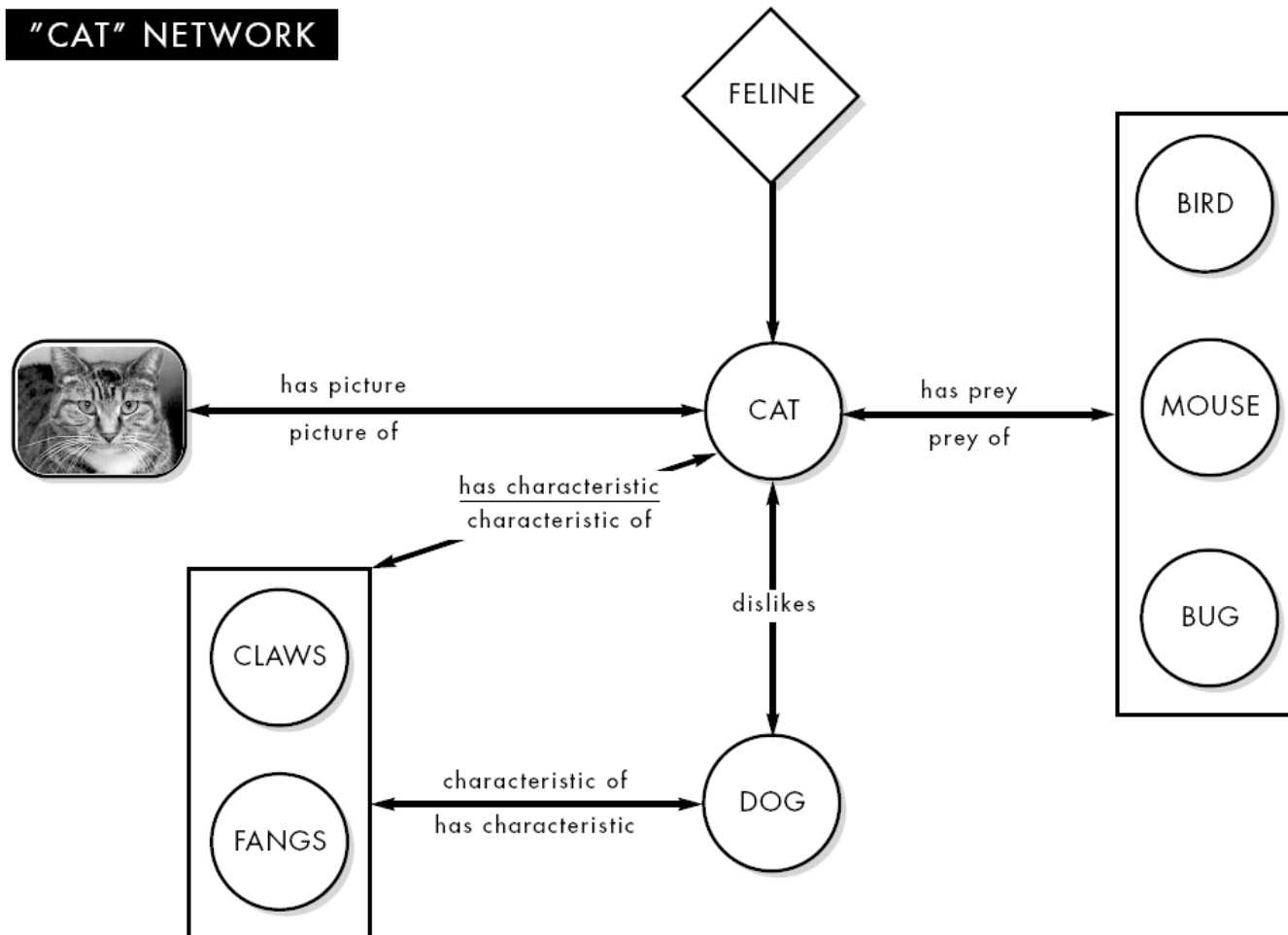
Nécessité de différencier les concepts individus (appartenance) des concepts classe (sorte_de) !

Liens réseaux sémantiques / phrases

- Le chat est un félin
- Le chat a pour proies la souris, l'oiseau, l'insecte
- Le chat n'aime pas le chien
- Le chat est représenté par son image par « id_photo »
- Le chat a des griffes

Exemple de réseau sémantique

"CAT" NETWORK



Exercice

- Décrivez un réseau sémantique représentant un étudiant/enseignant du master recherche
- Pour une tâche d'évaluation au sein du Master (évaluation de l'étudiant)
- Pour une tâche d'évaluation au sein du Master (évaluation de l'enseignant)
- Proposez une sémantique pour les relations proposées

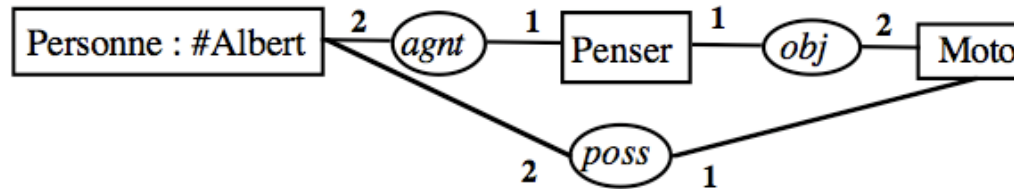
Réseau sémantique - exploitation

- Si on considère un réseau sémantique comme une conjonction de formules logiques, alors mêmes méthodes que pour un modèle logique
- Si on considère un réseau sémantique comme un graphe, alors on peut utiliser les techniques de propagation de marqueurs

Graphes conceptuels : notions fondamentales

- Sowa-84 : « Conceptual graphs form a knowledge representation language based on linguistics, psychology, and philosophy ».
- Au niveau conceptuel : langage de communication pour différents spécialistes impliqués dans une tâche commune.
- Au niveau implantation informatique : outil de représentation commun pour les différentes parties d'un système complexe.

S-graphs

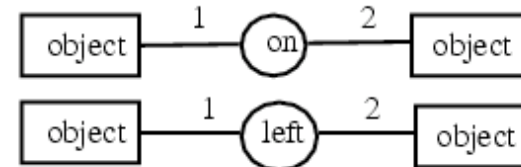
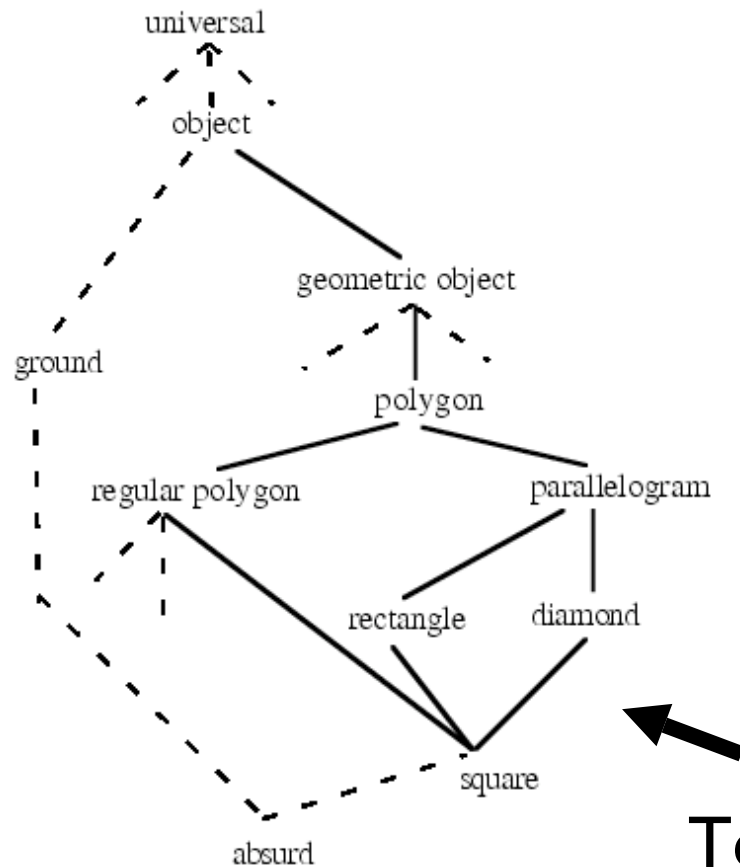


- Un graphe conceptuel est constitué de deux types de nœuds : les nœuds relations et les nœuds concepts.
- Les concepts représentent des entités, des attributs, des états et des événements, et
- Les relations montrent comment les concepts sont interconnectés.
- Concepts sommets
 - « individuels » ⇔ ont trait à des entités particulières
 - « génériques » et représentent des concepts non spécifiés d'un type donné.

S-Graph : graphe conforme à un support.

- Un graphe conceptuel est relatif à un « support », qui définit des contraintes syntaxiques permettant de décrire un domaine d'application.
- Cette notion de support regroupe:
 - Un ensemble de « types de concepts », structurés en hiérarchie « sorte-de » acceptant l'héritage multiple
 - Un ensemble de « types de relations »
 - Un ensemble de « graphes étoiles », appelés « bases », montrant pour chaque relation quels types de concepts elle peut connecter (signature de relation).
 - Un ensemble de « marqueurs » pour les « sommets concepts » : un marqueur « générique » et un marqueur « individuel »
 - Une relation de conformité, qui définit les contraintes d'association entre un type de concept et un marqueur (si le type « t » est associé au marqueur « m » alors il existe un individu m qui « est_un » t.

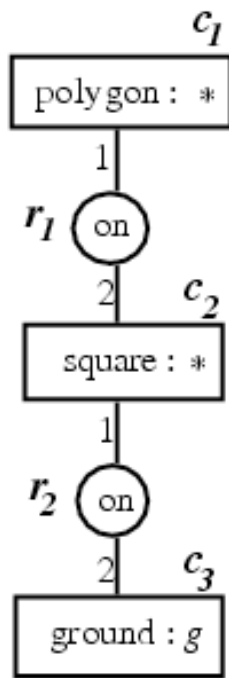
Support : exemple.



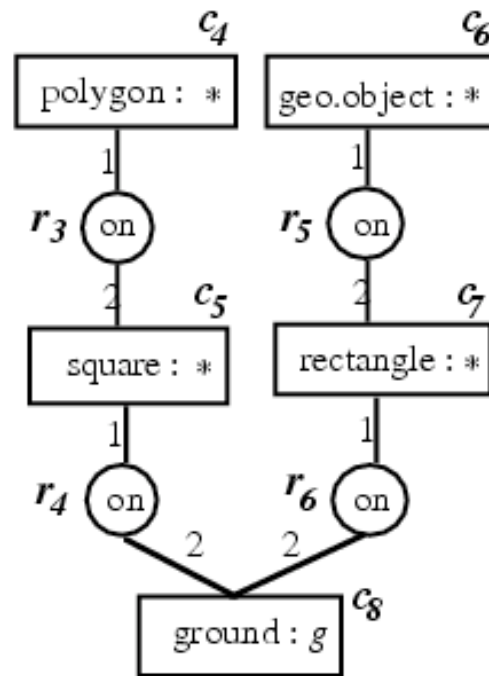
B, Base des relations
(graphes étoile)

Tc, Le treillis des concepts

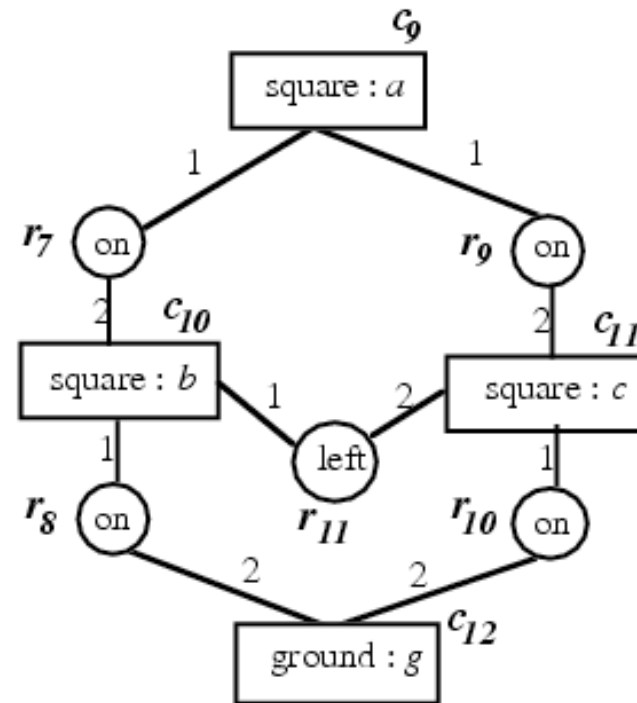
Des S-graphes conformes au support S



G

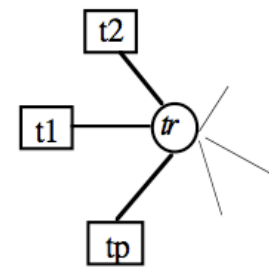
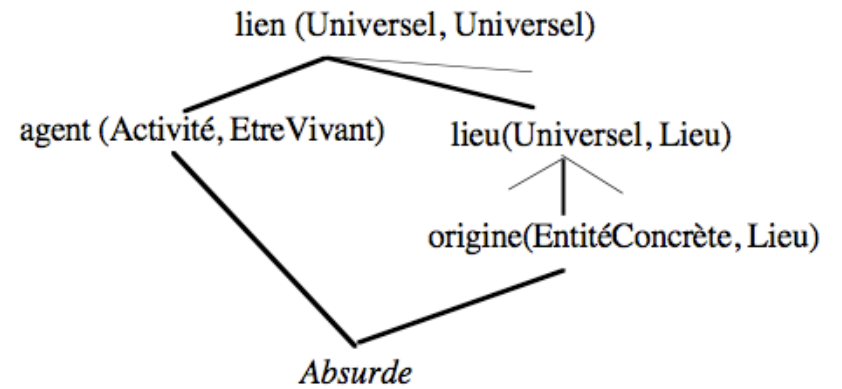
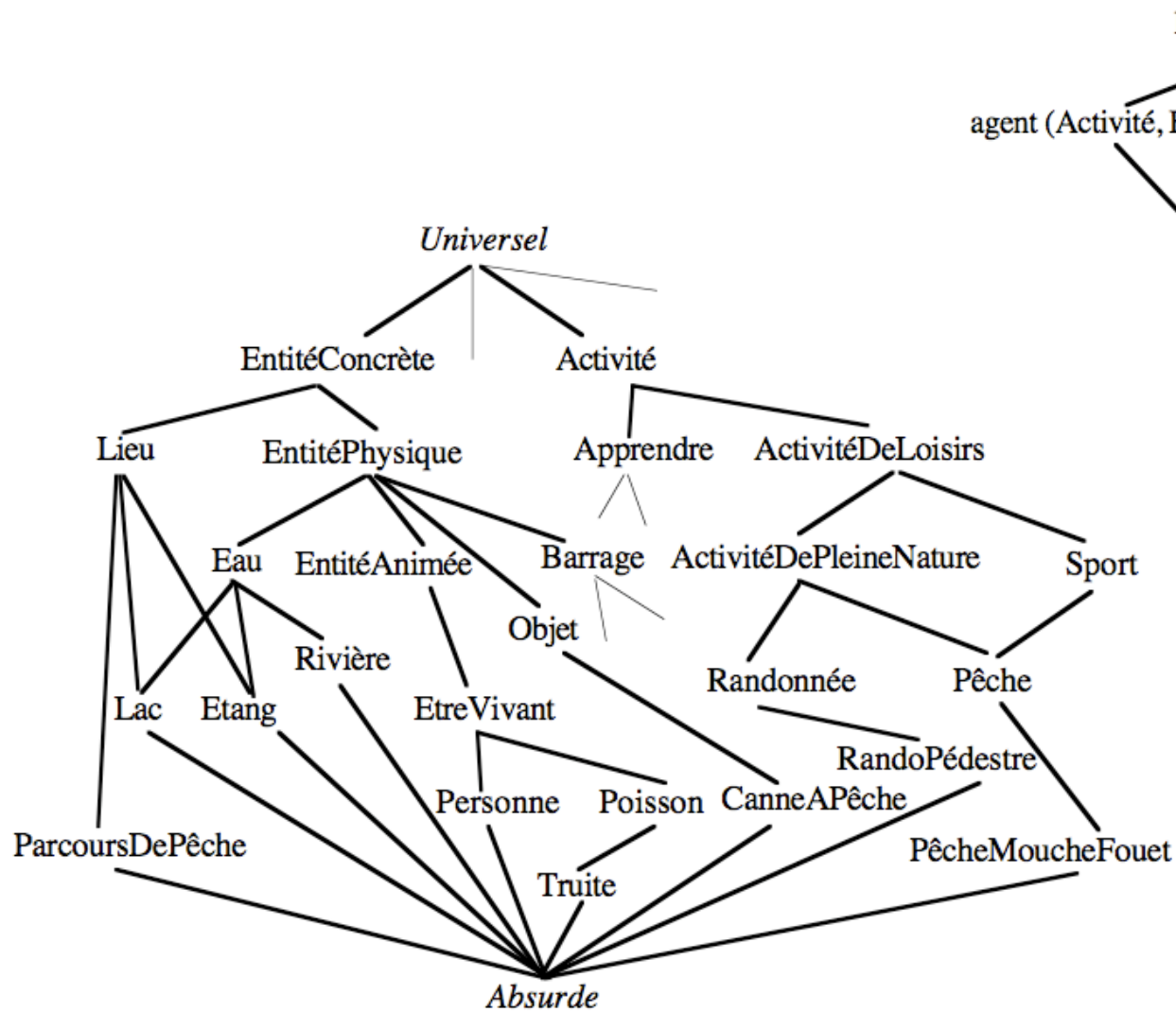


H



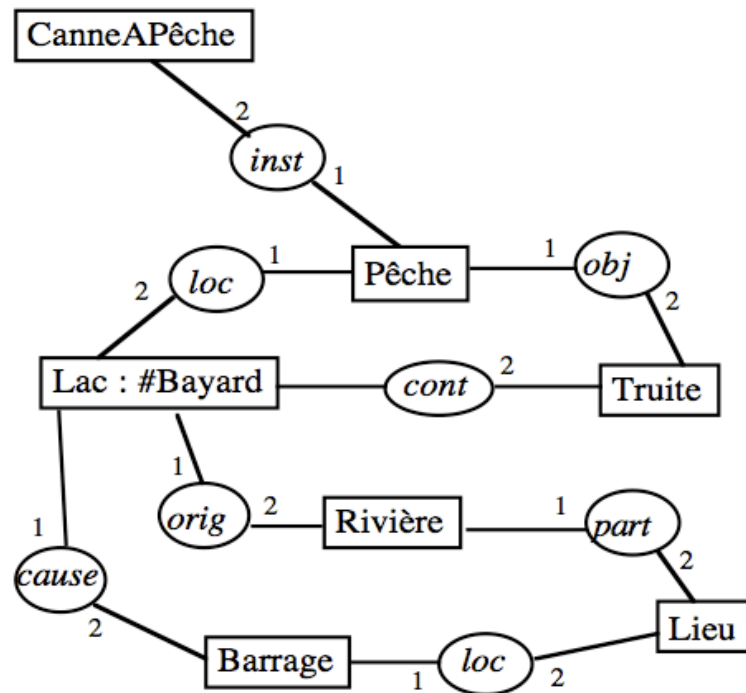
K

Support : autre exemple



graphe étoile

Exemple GC



Définition GC

Définition. Un graphe conceptuel simple $G = (R, C, U, \text{étiq})$ est un multigraphe³ non orienté, biparti, non nécessairement connexe, avec :

- R et C sont les deux classes de sommets, appelés sommets concepts et sommets relations, $C \neq \emptyset$

- U est l'ensemble des arêtes; l'ensemble des arêtes adjacentes à tout sommet relation r est totalement ordonné, ce qu'on représente en numérotant les arêtes (de 1 au degré de r). On note $G_i(r)$ le $i^{\text{ème}}$ voisin de r dans G .

- étiq est une application, qui à tout sommet associe une étiquette :

Si $r \in R$, $\text{étiq}(r) \in T_R$, c'est le type du sommet relation.

Si $c \in C$, $\text{étiq}(c) \in (T_C \setminus \{\perp\}) \times (I \cup \{*\})$. L'étiquette d'un sommet concept est un couple $(\text{type}(c), \text{marqueur}(c))$. Si c possède un marqueur générique, c est dit sommet concept générique, sinon c est un sommet concept individuel.

- étiq obéit aux contraintes fixées par les applications σ et τ .

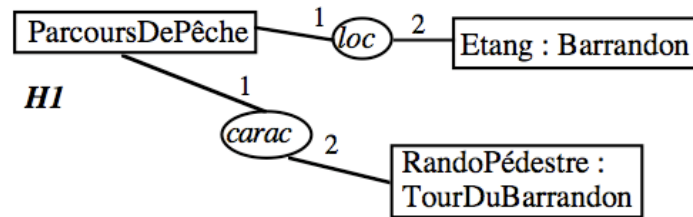
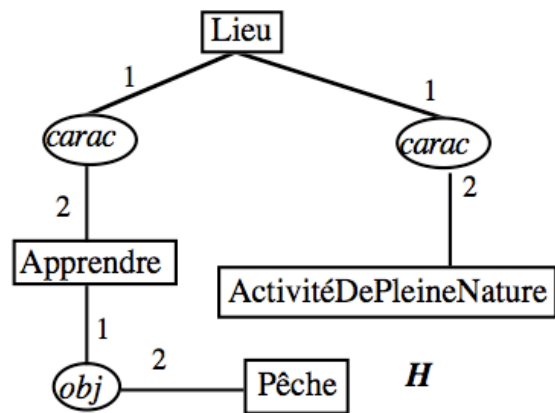
Pour tout $r \in R$, $\text{type}(G_i(r)) \leq \sigma_i(\text{type}(r))$.

Pour tout $c \in C$, si $\text{marqueur}(c) \in I$, alors $\text{type}(c) = \tau(\text{marqueur}(c))$ ⁴

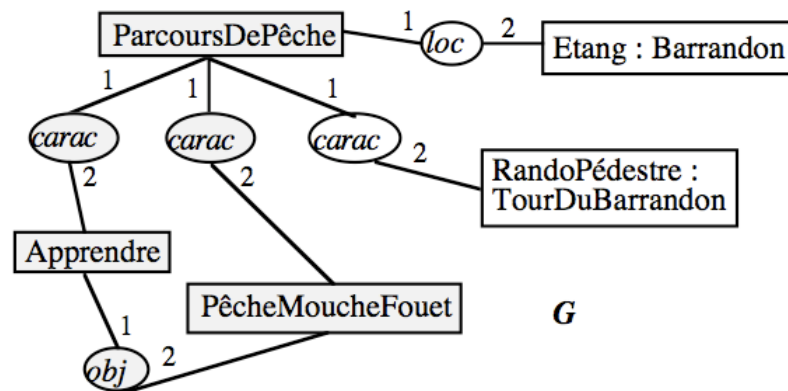
Spécialisation \leq

- Opération élémentaires de spécialisation
 - Simplification : suppression des sommets-r « jumeaux » (même type, mêmes voisins)
 - Restriction de relation : remplacement d'une étiquette de relation par un type « plus petit »
 - Restriction de concept : une étiquette « e » d'un sommet-c par une autre étiquette « e' » telle que $e' \leq e$ et e' conforme au type.
 - Joint interne : deux sommets-c d'un même graphe ayant même étiquette sont fusionnés
 - Joint externe : deux sommets-c de même étiquette et appartenant à 2 s-graphes sont fusionnés pour donner un nouveau s-graphe.
- G est une spécialisation de H si H appartient à une séquence de spécialisation (dérivation) arrivant à G.

Spécialisation \leq



$G \leq H$



Projection

- Permet de vérifier la spécialisation
- Opération fondamentale du modèle : toutes les autres opérations s'expriment en projection

Définition. Une *projection* d'un graphe conceptuel $H=(R_H, C_H, U_H, \text{étiq}_H)$ dans un graphe conceptuel $G=(R_G, C_G, U_G, \text{étiq}_G)$ est un couple d'applications $\Pi=(f,g)$, $f: R_H \rightarrow R_G$, $g: C_H \rightarrow C_G$, qui :

1. conserve les arêtes et la numérotation des arêtes :

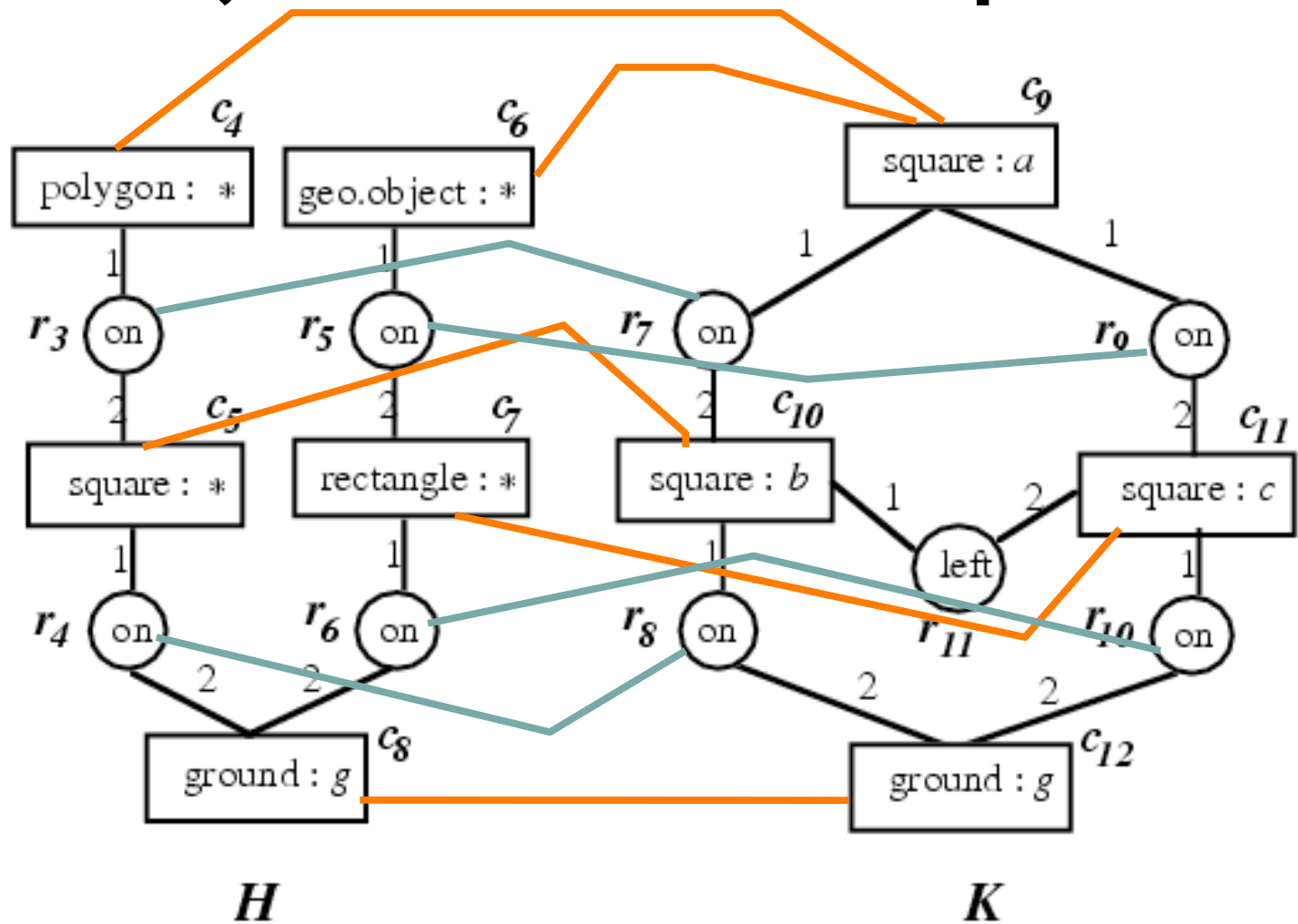
Pour tout arête rc de U_H , $f(r)g(c)$ est une arête de U_G . De plus, si $c = H_i(r)$, alors $g(c) = G_i(f(r))$.

2. peut restreindre les étiquettes des sommets

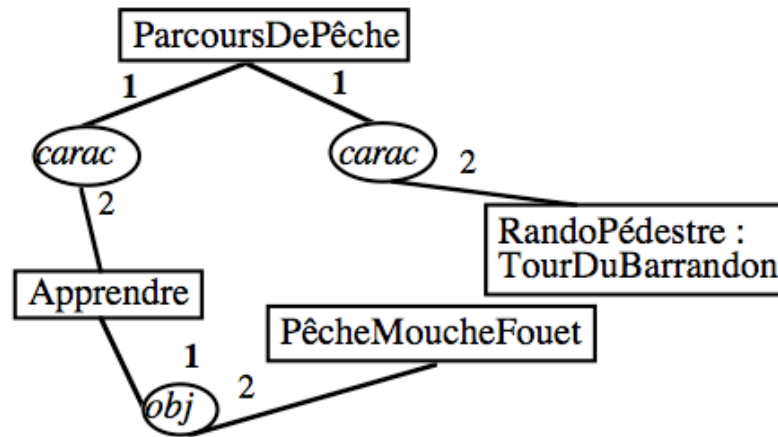
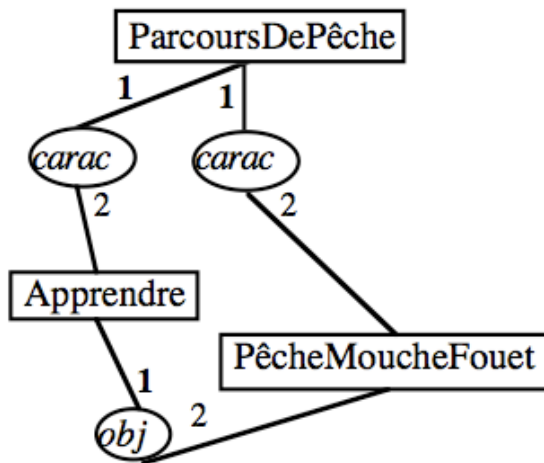
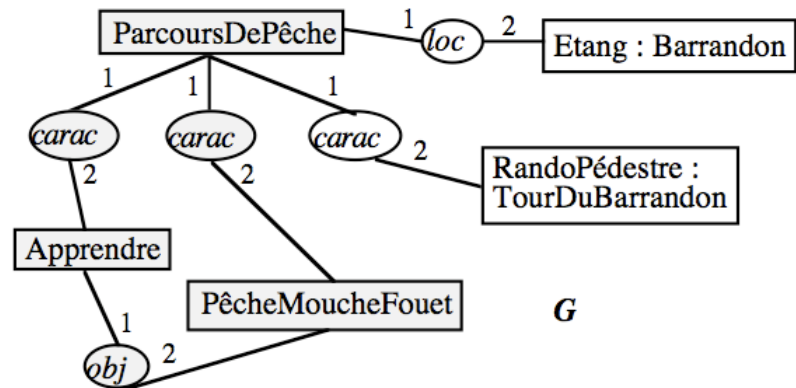
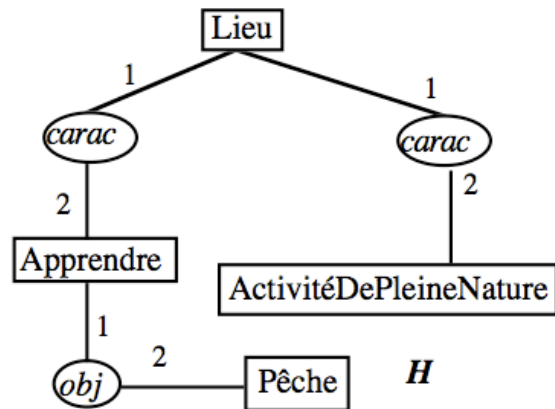
Pour tout sommet r de R_H , $\text{étiq}_G(f(r)) \leq \text{étiq}_H(r)$

Pour tout sommet c de C_H , $\text{étiq}_G(g(c)) \leq \text{étiq}_H(c)$

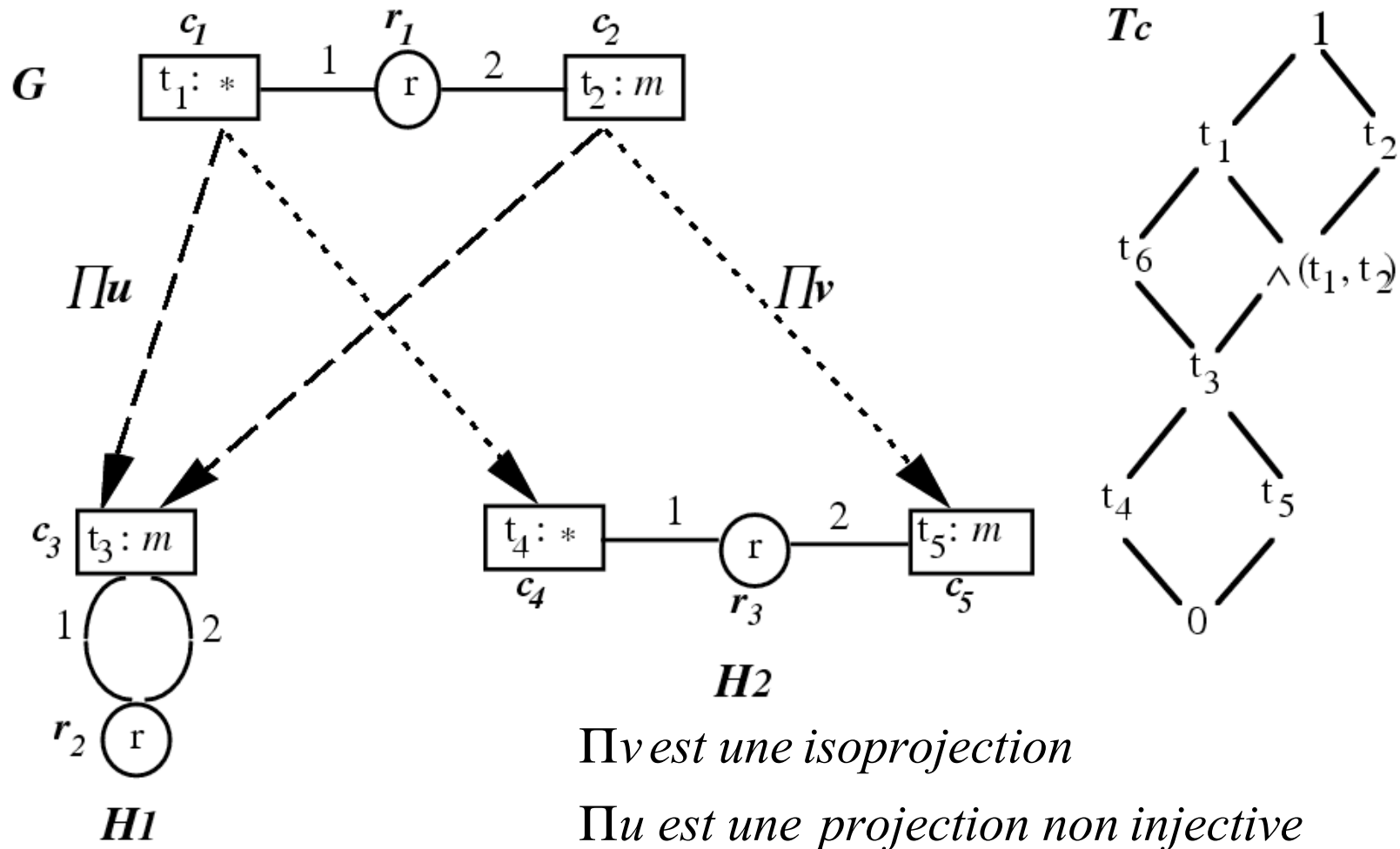
Projection : exemple



Deux projections de H dans G



Isomorphisme de S-graphe (iso-projection)



Opérations de généralisation = inverses de la spécialisation

- Règles de généralisation
 - Duplication : Soit r un sommet relation de G . On obtient H en ajoutant un sommet relation de même type que r et jumeau de r .
 - Augmentation de relation : Soit r un sommet relation de G . H s'obtient en augmentant le type de r .
 - Augmentation de concept : Soit c un sommet concept de G . On obtient H en augmentant l'étiquette de c : remplacer $\text{étiq}(c)=(t,m)$ par $e'=(t',m')$ avec $t' \geq t$ et $m' = *$. Cette opération doit conserver les contraintes fixées par σ : pour tout sommet relation r voisin de c , soit $c = G^{-1}(r)$, on a $t' \leq \sigma_i(\text{type}(r))$.
 - Eclatement : Soit c un sommet concept de G . On obtient H en éclatant c en deux sommets c_1 et c_2 , de même étiquette que c , tel que les ensembles d'arêtes adjacentes à c_1 et c_2 forment une bipartition (au sens large, l'un des deux ensemble peut être vide) de l'ensemble des arêtes adjacentes à c .
 - Décomposition : Supprimer certaines composantes connexes (pas toutes) de G .
- G est une généralisation de H si H appartient à une séquence d'opération de généralisation arrivant à G .

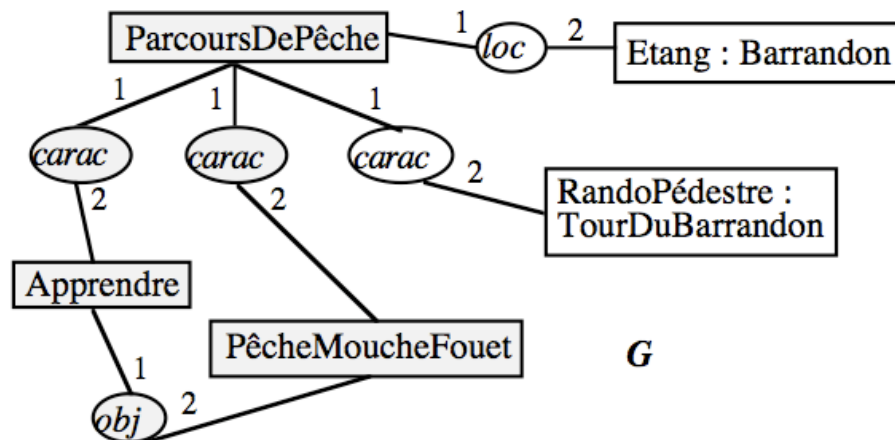
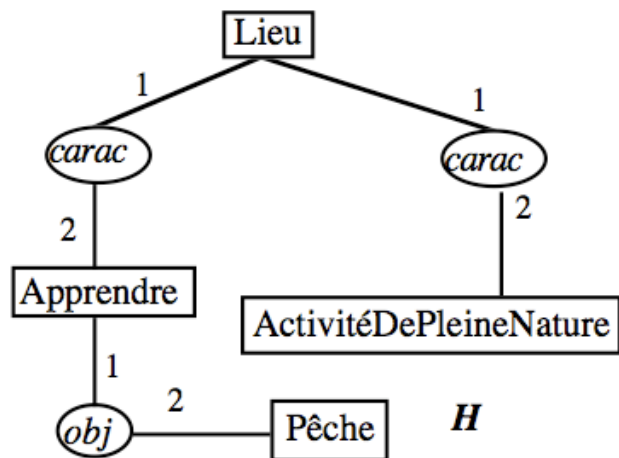
Quelques propriétés

- La relation de spécialisation \leq est un préordre (réflexive mais pas antisymétrique).
- Si $G \leq H$ et $H \leq G$ alors H et G sont équivalents : $H \equiv G$.
- Il existe une projection de H dans G ssi $G \leq H$
- Toute classe d'équivalence pour \leq admet un unique graphe irredondant (c'est d'ailleurs le plus petit graphe de la classe, au sens de la taille).
- graphe irredondant : pas équivalent à un de ses sous-graphes stricts

Interprétation logique des GC

$\Phi(H) = \exists x \exists y \exists z \exists t \text{ Lieu}(x) \text{ Apprendre}(y) \wedge \text{ActivitéDePleineNature}(z) \wedge \text{Pêche}(t) \wedge \text{carac}(x,y) \wedge \text{carac}(x,z) \wedge \text{obj}(y,t)$

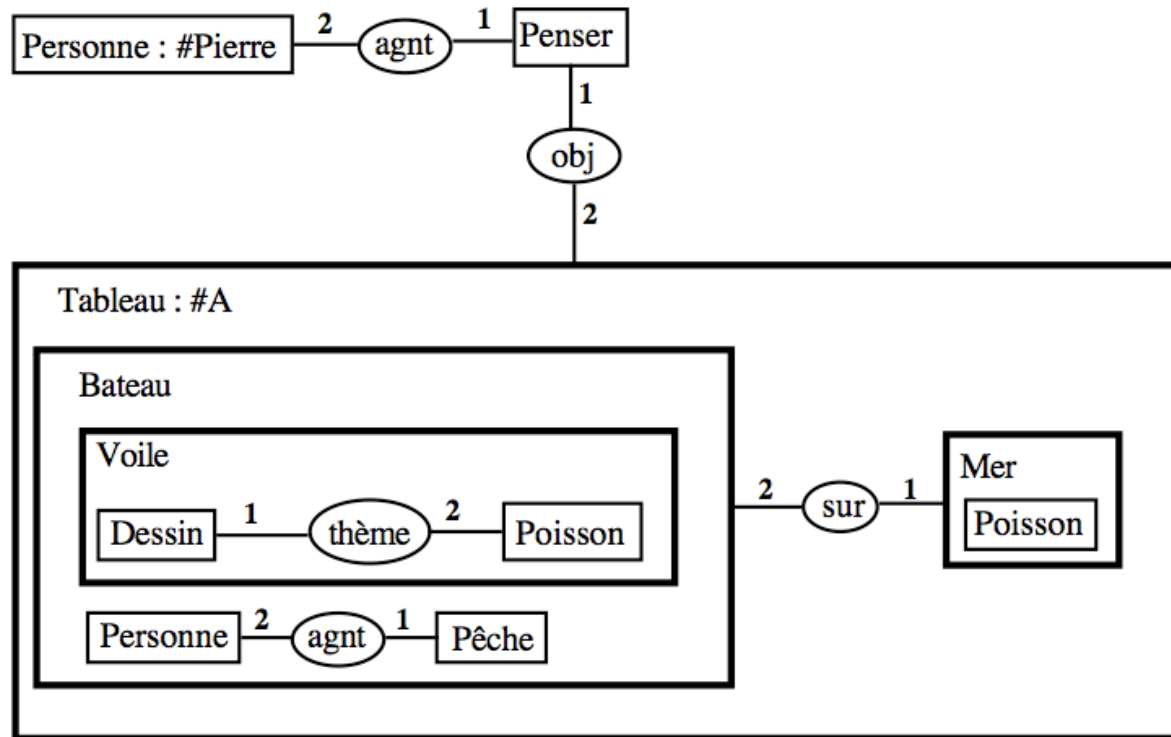
$\Phi(G) = \exists x \exists y \exists z \text{ ParcoursDePêche}(x) \text{ Apprendre}(y) \wedge \text{PêcheMoucheFouet}(z) \wedge \text{carac}(x,y) \wedge \text{carac}(x,z) \wedge \text{obj}(y,z) \wedge \text{RandoPédestre}(\text{TourDuBarrandon}) \wedge \text{carac}(x,\text{TourDuBarrandon}) \wedge \text{Etang}(\text{Barrandon}) \wedge \text{loc}(x,\text{Barrandon})$.



Complexité de la projection

- Le problème de la projection est NP-Complet
- Le problème de la recherche de sous-graphe (projection injective) est NP-Complet
- Le problème de l'isoprojection est NP-Complet (même si on a un ordre total sur les étiquettes des sommets-c!).
- Le problème de l'équivalence est NP-Complet.

Raffinements : graphes emboîtés



Plan

- Représentation de connaissances
- Ingénierie des connaissances
- Ontologies
- Graphes conceptuels
- **Logiques de description**
- Introduction au Web sémantique
- OWL
- Conception d'ontologies

Logiques de description

- **Famille** de langages de représentation des connaissances, offrant de bons compromis entre
 - lisibilité et formalisation
 - réseaux sémantiques, *frames*
 - *communication homme - machine*
 - expressivité et décidabilité/complexité
 - \subseteq logique du premier ordre
 - communication machine – machine
- Bons candidats pour représenter des ontologies

Principes de base (1)

Les termes du langage sont séparés en familles *disjointes* :

- Les concepts (ou classes)
 - interprétés comme des sous-ensembles de $\Delta^{\mathfrak{S}}$
 - *exemples : Personne, Étudiant, Document*
- Les rôles (ou propriétés)
 - interprétés comme des relations binaires sur $\Delta^{\mathfrak{S}}$
 - *exemples : père, auteur*

Principes de base (2)

- Dans certaines LD, d'autres familles de termes sont utilisées :
 - Les individus
 - interprétés comme des éléments de Δ
 - exemples : pa_champin, cette_présentation
 - Les valeurs concrètes (rare)
 - interprétés comme des éléments de ΔD
 - ΔD Domaine concret
 - exemple : 42, 3.1415, "hello world"

Différentes LD

- Les diverses LD se différencient par :
- Les termes complexes que l'on peut construire à partir de termes de bases
 - *exemple : « toute personne qui est l'auteur d'un document »*
- Les axiomes que l'on peut exprimer (« phrases »)
 - *exemple : « un étudiant est toute personne qui suit un cours »*
- Les mécanismes d'inférence qu'elles offrent

Concepts complexes (1)

- constructeurs ensemblistes

- $- C \sqcap D \xrightarrow{\cong} C^{\mathfrak{S}} \cap D^{\mathfrak{S}}$

- $- C \sqcup D \xrightarrow{\cong} C^{\mathfrak{S}} \cup D^{\mathfrak{S}}$

- $- \neg C \xrightarrow{\cong} \Delta^{\mathfrak{S}} \setminus C^{\mathfrak{S}}$

- quantificateurs

- $- \exists R.C \xrightarrow{\cong} \{ x \mid \exists (x,y) \in R^{\mathfrak{S}}, y \in C^{\mathfrak{S}} \}$

- $- \forall R.C \xrightarrow{\cong} \{ x \mid \forall (x,y) \in R^{\mathfrak{S}}, y \in C^{\mathfrak{S}} \}$

- restrictions de cardinalité

- $- \leq n R \xrightarrow{\cong} \{ x \mid \text{card}\{(x,y) \in R^{\mathfrak{S}}\} \leq n \}$

- $- \geq n R \xrightarrow{\cong} \{ x \mid \text{card}\{(x,y) \in R^{\mathfrak{S}}\} \geq n \}$

- $- = n R \xrightarrow{\cong} \{ x \mid \text{card}\{(x,y) \in R^{\mathfrak{S}}\} = n \}$

Concepts complexes (2)

- restrictions de cardinalité qualifiées

- $\leq n$ R.C $\xrightarrow{\mathfrak{S}}$ $\{ x \mid \text{card}\{(x,y) \in R^{\mathfrak{S}} \text{ et } y \in C^{\mathfrak{S}}\} \leq n \}$

- $\geq n$ R.C $\xrightarrow{\mathfrak{S}}$ $\{ x \mid \text{card}\{(x,y) \in R^{\mathfrak{S}} \text{ et } y \in C^{\mathfrak{S}}\} \geq n \}$

- $= n$ R.C $\xrightarrow{\mathfrak{S}}$ $\{ x \mid \text{card}\{(x,y) \in R^{\mathfrak{S}} \text{ et } y \in C^{\mathfrak{S}}\} = n \}$

- extension

- $\{ i, j, k, \dots \} \xrightarrow{\mathfrak{S}}$ $\{ i^{\mathfrak{S}}, j^{\mathfrak{S}}, k^{\mathfrak{S}}, \dots \}$

Exemples de concepts complexes

- \neg (Étudiant \sqcup Salarié)
- Groupe $\sqcap \exists$ membre . Étudiant
 - Peut contenir un *groupe sans aucun membre* ?
- Groupe $\sqcap \forall$ membre . Étudiant
 - Peut contenir un *groupe sans aucun membre* ?
- Groupe $\sqcap (\geq 10$ membre)
- (= 1 auteur)
- (= 1 auteur.Étudiant)
- { john, paul, george, ringo }

Rôles complexes (certaines LD)

- *constructeurs ensemblistes*

- $R \sqcap S \xrightarrow{\mathfrak{S}} R^{\mathfrak{S}} \cap S^{\mathfrak{S}}$

- $R \sqcup S \xrightarrow{\mathfrak{S}} R^{\mathfrak{S}} \cup S^{\mathfrak{S}}$

- $\neg R \xrightarrow{\mathfrak{S}} \Delta^{\mathfrak{S}} \times \Delta^{\mathfrak{S}} \setminus R^{\mathfrak{S}}$

- *composition*

- $R \circ S \xrightarrow{\mathfrak{S}} \{ (x,y) \mid \exists z, (x,z) \in R^{\mathfrak{S}} \text{ et } (z,y) \in S^{\mathfrak{S}} \}$

- *inverse*

- $R^{-} \xrightarrow{\mathfrak{S}} \{ (x,y) \mid (y,z) \in R^{\mathfrak{S}} \}$

- *fermeture transitive*

- $R^* \xrightarrow{\mathfrak{S}} \{ (x,y) \mid (x,y) \in (R^{\mathfrak{S}})^* \}$

Exemples de rôles complexes

- père \sqcup mère
- collègue \sqcap ami
- père \circ mère
- parent -
- parent *

Axiomes

- Définition de concept atomique
 - $A \sqsubseteq C \xrightarrow{\mathfrak{S}} A^{\mathfrak{S}} \subseteq C^{\mathfrak{S}}$ (subsumption, CN)
 - $A \equiv C \xrightarrow{\mathfrak{S}} A^{\mathfrak{S}} = C^{\mathfrak{S}}$ (équivalence, CNS)
 - dans les premières LD, les définitions récursives (ou C fait référence à A) n'étaient même pas autorisées
- *General Inclusion Axiom* ($C \sqsubseteq D, C \equiv D$)
 - entre concept complexes, limitations
- Hiérarchie de rôles ($R \sqsubseteq S, R \equiv S$)
- Autres axiomes (rôle transitif, rôle réflexif...)

A-Box

Dans les LD qui supportent les individus :

- T-Box (axiomes sur les classes et les rôles)
- A-Box (axiomes sur les individus)

$$- i:C \xrightarrow{\mathfrak{I}} i^{\mathfrak{I}} \in C^{\mathfrak{I}}$$

$$- i,j:R \xrightarrow{\mathfrak{I}} (i^{\mathfrak{I}}, j^{\mathfrak{I}}) \subseteq R^{\mathfrak{I}}$$

$$- i=j \xrightarrow{\mathfrak{I}} i^{\mathfrak{I}} = j^{\mathfrak{I}}$$

$$- i \neq j \xrightarrow{\mathfrak{I}} i^{\mathfrak{I}} \neq j^{\mathfrak{I}}$$

- *NB : pas de distinction dans les LD qui supportent l'extension*

$$- i:C \Leftrightarrow \{i\} \sqsubseteq C, \quad i,j:R \Leftrightarrow \{i\} \sqsubseteq \exists R.\{j\}, \text{ etc.}$$

Inférences dans les LD

- Inférences de base
 - satisfiabilité : $\exists \mathfrak{S}, C^{\mathfrak{S}} \neq \emptyset$
 - un concept peut-il avoir des instances ?
 - subsomption : $\forall \mathfrak{S}, C^{\mathfrak{S}} \subseteq D^{\mathfrak{S}}$
 - un concept en subsume-t'il un autre ?
 - classification
 - où placer un concept dans la hiérarchie ?
- Inférence sur les individus
 - identification ou test à l'instanciation
 - quel est le concept le plus spécifique décrivant un individu ?
- Raisonnement sur une base de connaissance
 - *T-Box seule* ou *T-Box + A-Box*

Inférence dans les LD

- On peut tout ramener à un problème de satisfiabilité, donc au test d'existence d'un modèle pour une expression
 - C est subsumé par $D \Leftrightarrow C \sqcap \neg D$ est insatisfiable
 - C et D sont équivalents $\Leftrightarrow C \sqcap \neg D$ et $D \sqcap \neg C$ sont *insatisfiables*
 - C et D sont disjoints $\Leftrightarrow C \sqcap D$ est insatisfiable
 - a est une instance de $C \Leftrightarrow ABox \cup \{a : \neg C\}$ est insatisfiable
- La complexité dépend de la logique de description choisie

LD – Méthode des tableaux (1)

- Un tableau est une représentation d'un ensemble de modèles
 - arbre fini étiqueté
 - chaque branche mémorise une série d'évaluations possibles pour les énoncés testé
- L'application de règles de transformation (adaptation de la méthode des tableaux en logique des propositions) garantit qu'on explorera tous les modèles possibles
- Principe : raisonnement par réfutation
 - on suppose qu'une instance x existe
 - on déduit tout ce qu'on peut sur cette instance
 - si on arrive à une contradiction ou clash : $x \in C \cap \neg C$, il n'y a pas de modèle
 - si on arrive à une branche complète : on a un modèle

LD – Méthode des tableaux (2)

Personne, père ≥ 2



- Homme $\sqsubseteq \neg$ Femme
- **Personne \sqsubseteq
 \exists père Homme \sqcap
 \exists mère Femme \sqcap
 \forall père Homme \sqcap
parent ≤ 2**
- père \sqsubseteq parent
- mère \sqsubseteq parent
- **Personne \sqcap père ≥ 2
est il satisfiable ?**

LD – Méthode des tableaux (3)

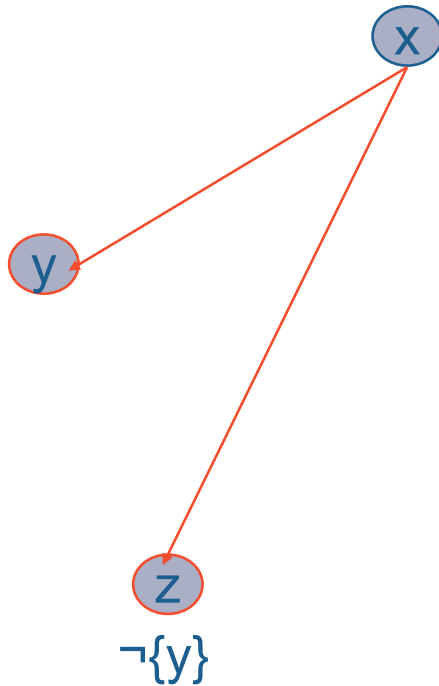
Personne, père ≥ 2 ,
 \exists père Homme, \exists mère Femme,
 \forall père Homme, parent ≤ 2

x

- Homme $\sqsubseteq \neg$ Femme
- Personne \sqsubseteq
 \exists père Homme \sqcap
 \exists mère Femme \sqcap
 \forall père Homme \sqcap
parent ≤ 2
- père \sqsubseteq parent
- mère \sqsubseteq parent
- Personne \sqcap père ≥ 2
est il satisfiable ?

LD – Méthode des tableaux (4)

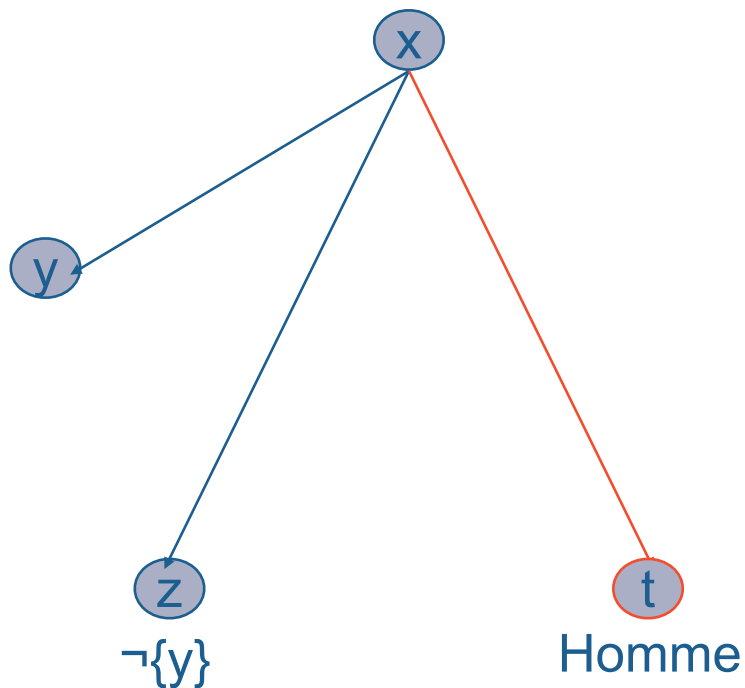
Personne, père ≥ 2 ,
 \exists père Homme, \exists mère Femme,
 \forall père Homme, parent ≤ 2



- Homme $\sqsubseteq \neg$ Femme
- Personne \sqsubseteq
 \exists père Homme \sqcap
 \exists mère Femme \sqcap
 \forall père Homme \sqcap
parent ≤ 2
- père \sqsubseteq parent
- mère \sqsubseteq parent
- Personne \sqcap père ≥ 2
est il satisfiable ?

LD – Méthode des tableaux (5)

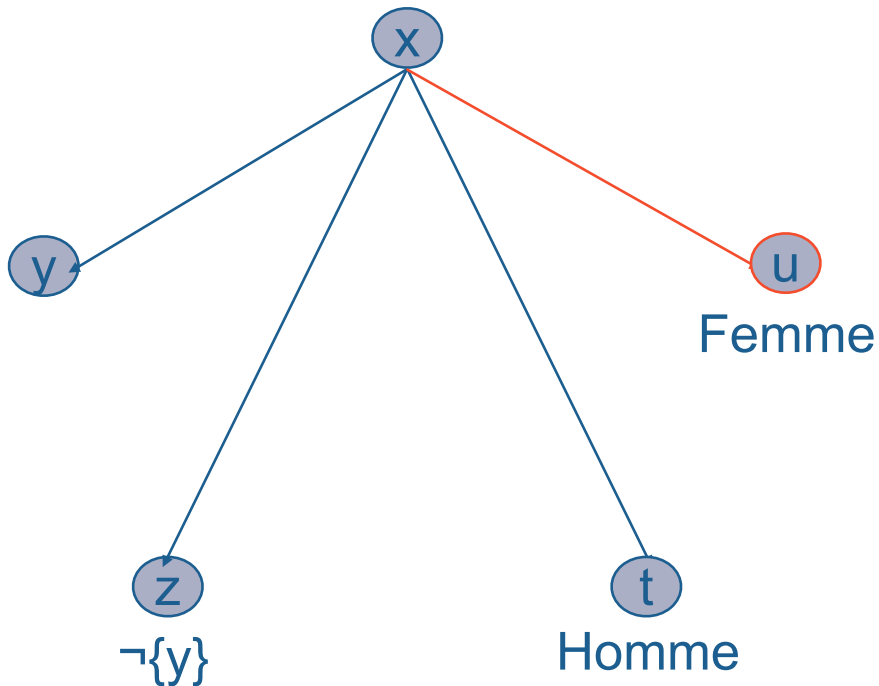
Personne, père ≥ 2 ,
 \exists père Homme, \exists mère Femme,
 \forall père Homme, parent ≤ 2



- Homme $\sqsubseteq \neg$ Femme
- Personne \sqsubseteq
 \exists père Homme \sqcap
 \exists mère Femme \sqcap
 \forall père Homme \sqcap
parent ≤ 2
- père \sqsubseteq parent
- mère \sqsubseteq parent
- Personne \sqcap père ≥ 2
est il satisfiable ?

LD – Méthode des tableaux (6)

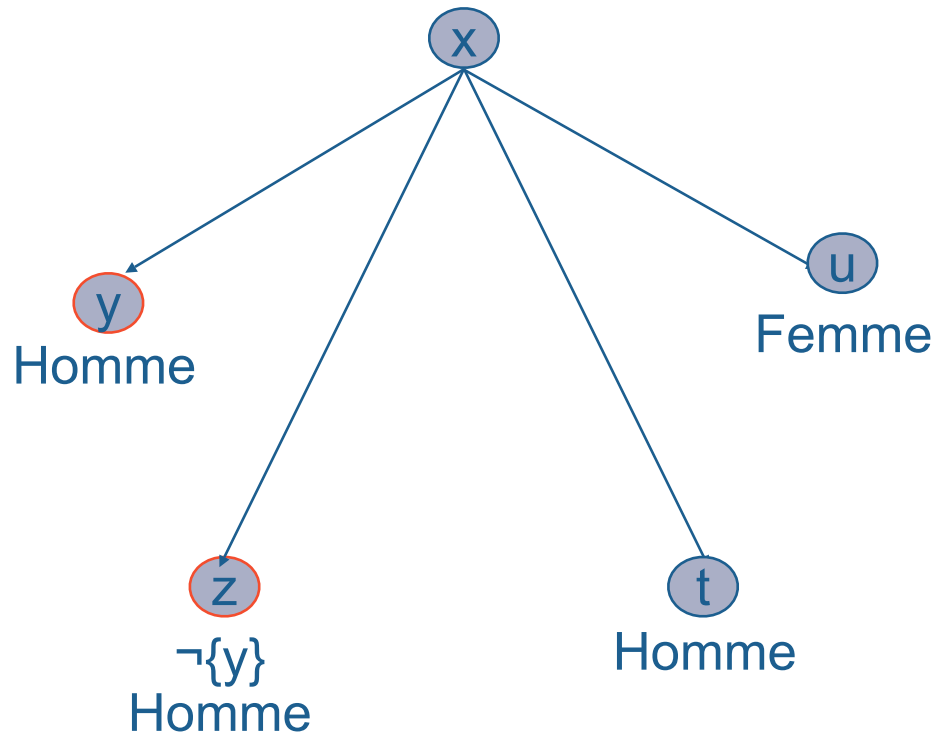
Personne, père ≥ 2 ,
 \exists père Homme, \exists mère Femme,
 \forall père Homme, parent ≤ 2



- Homme $\sqsubseteq \neg$ Femme
- Personne \sqsubseteq
 \exists père Homme \sqcap
 \exists mère Femme \sqcap
 \forall père Homme \sqcap
parent ≤ 2
- père \sqsubseteq parent
- mère \sqsubseteq parent
- Personne \sqcap père ≥ 2
est il satisfiable ?

LD – Méthode des tableaux (7)

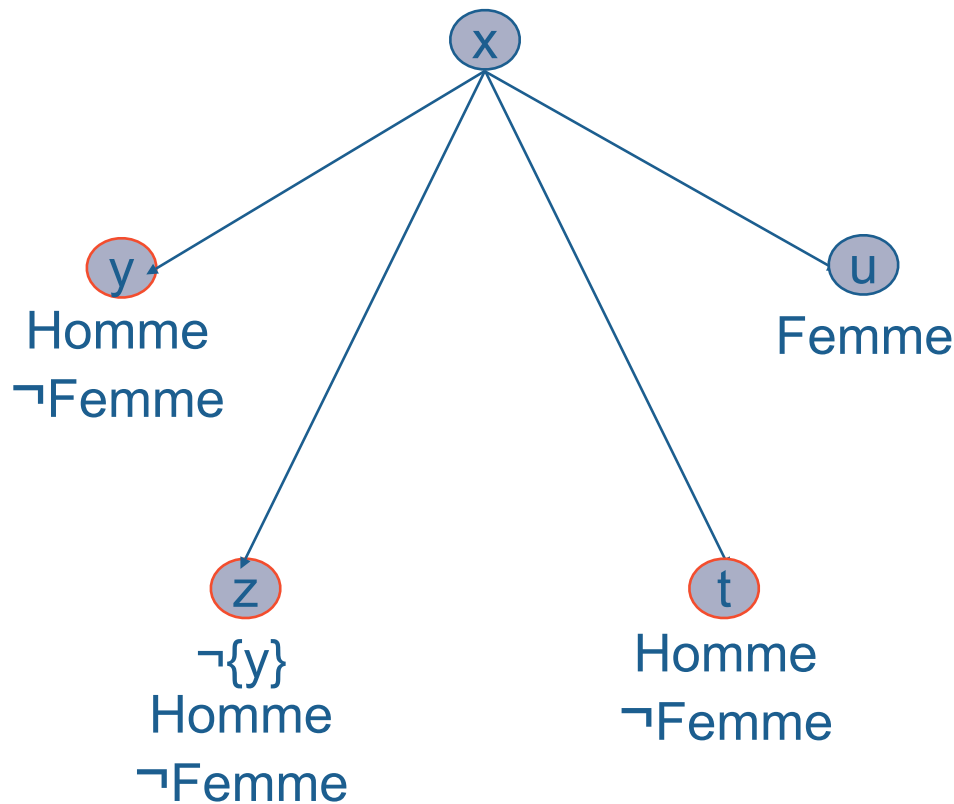
Personne, père ≥ 2 ,
 \exists père Homme, \exists mère Femme, \forall
père Homme, parent ≤ 2



- Homme \sqsubseteq \neg Femme
- Personne \sqsubseteq
 \exists père Homme \sqcap
 \exists mère Femme \sqcap
 \forall père Homme \sqcap
parent ≤ 2
- père \sqsubseteq parent
- mère \sqsubseteq parent
- Personne \sqcap père ≥ 2
est il satisfiable ?

LD – Méthode des tableaux (8)

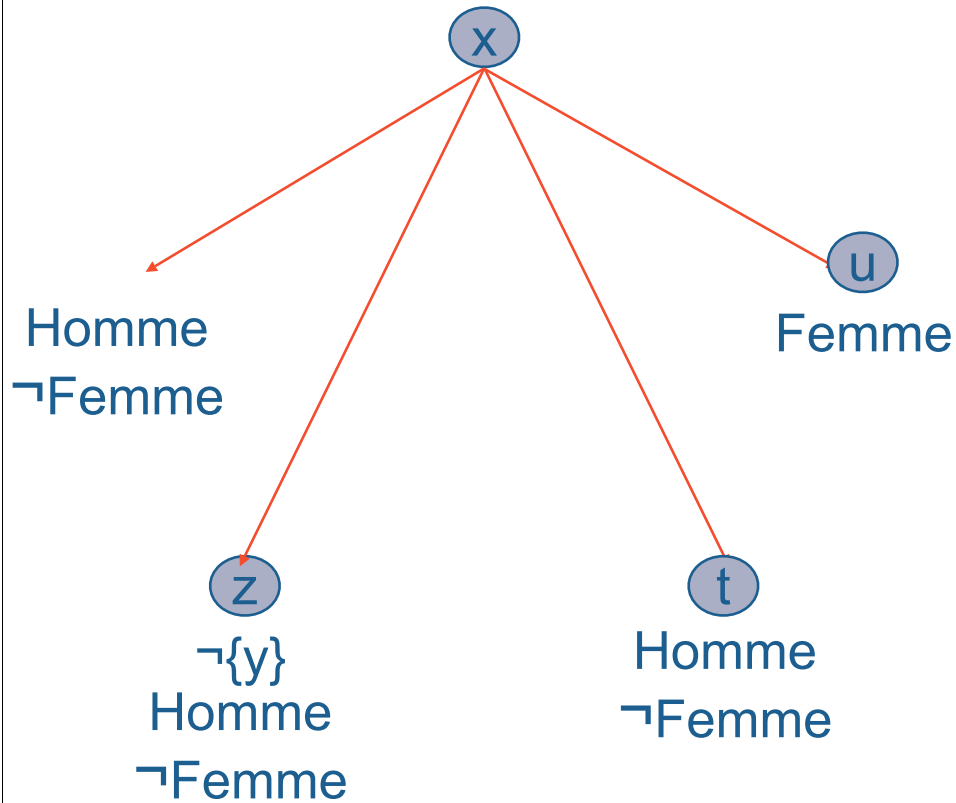
Personne, père ≥ 2 ,
 \exists père Homme, \exists mère Femme,
 \forall père Homme, parent ≤ 2



- Homme \sqsubseteq \neg Femme
- Personne \sqsubseteq
 - \exists père Homme \sqcap
 - \exists mère Femme \sqcap
 - \forall père Homme \sqcap
 - parent ≤ 2
- père \sqsubseteq parent
- mère \sqsubseteq parent
- Personne \sqcap père ≥ 2
est il satisfiable ?

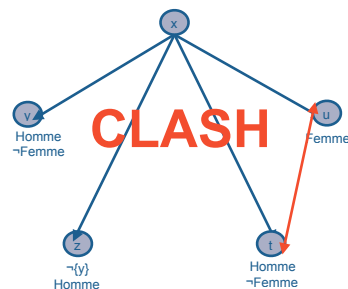
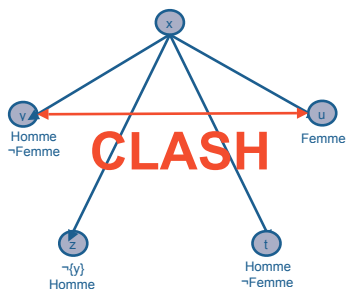
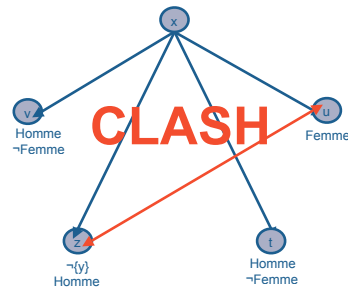
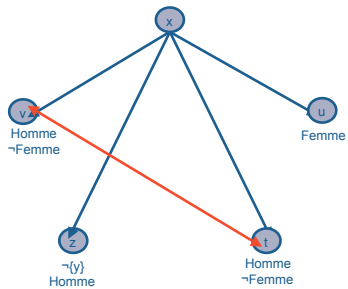
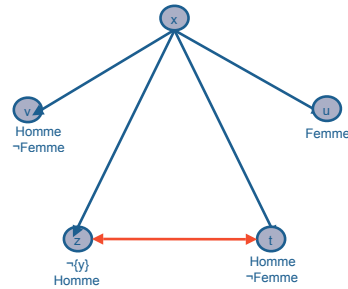
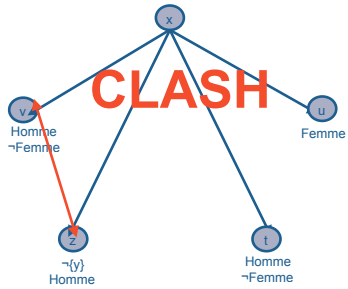
LD – Méthode des tableaux (9)

Personne, père ≥ 2 ,
 \exists père Homme, \exists mère Femme,
 \forall père Homme, parent ≤ 2



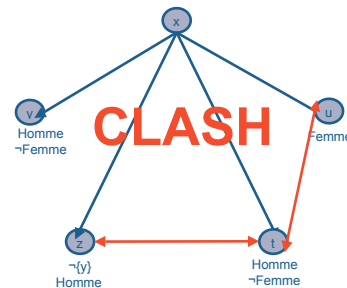
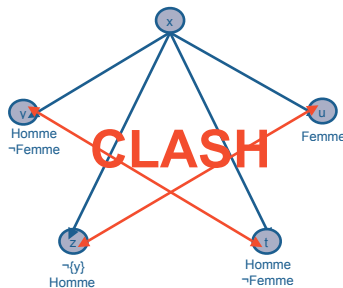
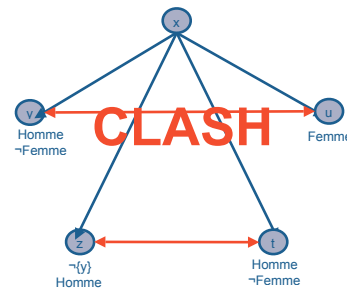
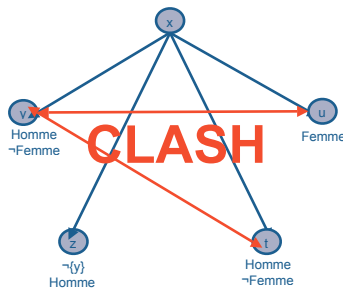
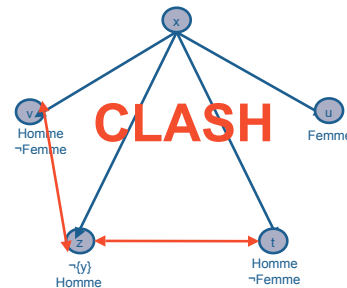
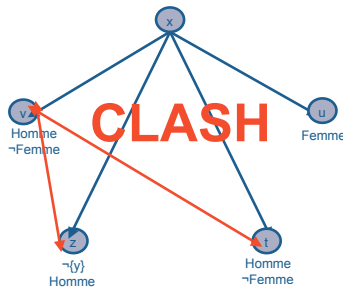
- Homme \sqsubseteq \neg Femme
- Personne \sqsubseteq
 \exists père Homme \sqcap
 \exists mère Femme \sqcap
 \forall père Homme \sqcap
parent ≤ 2
- père \sqsubseteq parent
- mère \sqsubseteq parent
- Personne \sqcap père ≥ 2
est il satisfiable ?

LD – Méthode des tableaux (10)



- Homme \sqsubseteq ¬ Femme
- Personne \sqsubseteq
 \exists père Homme \sqcap
 \exists mère Femme \sqcap
 \forall père Homme \sqcap
parent \leq 2
- père \sqsubseteq parent
- mère \sqsubseteq parent
- Personne \sqcap père \geq 2
est il satisfiable ?

LD – Méthode des tableaux (I I)



- $\text{Homme} \sqsubseteq \neg \text{Femme}$
- $\text{Personne} \sqsubseteq$
 $\exists \text{ père Homme} \sqcap$
 $\exists \text{ mère Femme} \sqcap$
 $\forall \text{ père Homme} \sqcap$
 $\text{parent} \leq 2$
- $\text{père} \sqsubseteq \text{parent}$
- $\text{mère} \sqsubseteq \text{parent}$
- $\text{Personne} \sqcap \text{père} \geq 2$
n'est **pas** satisfiable

Plan

- Représentation de connaissances
- Ingénierie des connaissances
- Ontologies
- Graphes conceptuels
- Logiques de description
- **Introduction au Web sémantique**
- OWL
- Conception d'ontologies

Le Web aujourd'hui

- Universalité
 - Homogénéité des techniques utilisées
 - HTTP, HTML, intranet
 - Puissance de l'hypertexte
 - toute ressource peut être liée à toute autre ressource
- Ressources du Web
 - Principalement constituées pour une exploitation humaine
 - Ecrites par des humains ou générées par des machines
- Nécessité de médiateurs
 - Moteurs de recherche plus ou moins spécialisés, plus ou moins efficaces
- Mais
 - trouver les bonnes ressources, analyser le contenu des pages, dégager les bonnes informations, combiner les différents résultats est coûteux en temps... et parfois fastidieux

Agents logiciels sur le Web

- Les moteurs de recherche par mots clés laissent beaucoup de travail à l'utilisateur
- Peu de descriptions structurées des informations (méta-données) utilisables et utilisées par des agents logiciels
- Problème
 - HTML ne donne que des pages et des liens
 - dont les étiquettes sont interprétables seulement par leurs utilisateurs
 - sans sémantique utilisable par les machines
- Conclusion
 - l'information et les services sur le Web sont aujourd'hui peu exploitables par des machines,
 - mais peut-être de moins en moins exploitables sans l'aide des machines !

Exemples de requêtes complexes

- Trouver des résumés d'articles écrits par des chercheurs que connaît YP, après 2004 portant sur le Web sémantique (ou les extraits intéressants signalés par d'autres chercheurs)
- Trouver des entreprises du secteur informatique faisant des pertes, avec un chiffre d'affaire entre ... et ... et ayant eu des contrats avec d'autres entreprises logicielles dans les derniers 90 jours
- Trouver des locations de vacances familiales dans telle région avec des activités de plein-air, quelques musées et églises touristiques et un festival de jazz dans la période du au...

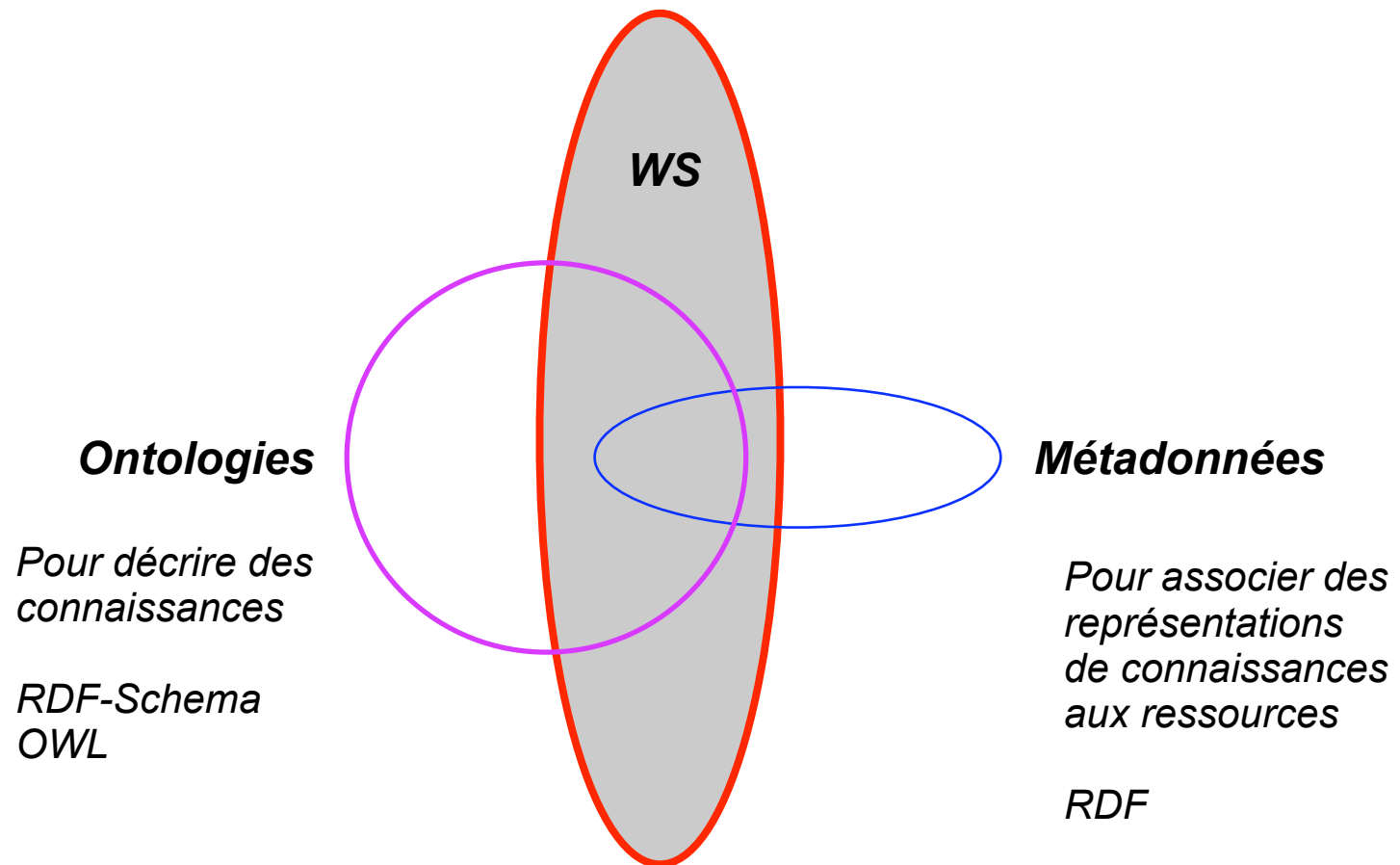
Le Web de demain (W3C, 2000)

- « Un vaste espace d'échanges de ressources entre machines permettant l'exploitation de grands volumes d'informations et de services variés, aidant les utilisateurs en les libérant d'une (bonne) partie de leur travail de recherche, et de combinaison de ces ressources »
- « Le Web sémantique n'est pas un Web séparé, mais une extension du Web actuel dans lequel l'information est munie d'une signification bien définie permettant aux ordinateurs et aux personnes de mieux travailler en coopération » (T. Berners-Lee , J. Hendler et O.Lassila, Scientific American 2001)
- Les ressources sont interprétées et utilisées au moins autant par des logiciels que par des utilisateurs humains au service des utilisateurs
 - Utilisation de méta-données sémantiques

L'approche initiale du Web sémantique

- Mise en place d'une couche « sémantique »
 - permettant la description des ressources et des contenus
 - et la structuration des ressources (liens entre ressources étiquetés) avec un certain niveau de formalisation
 - utilisables par des agents logiciels
- De tels logiciels peuvent utiliser
 - les réseaux de concepts et de relations présentes dans ces descriptions
 - combiner ces significations
- Ces descriptions associées à une ressource lors de sa création ou plus tard
 - Et codées dans des langages standards dédiés
 - RDF, RDF(S), OWL

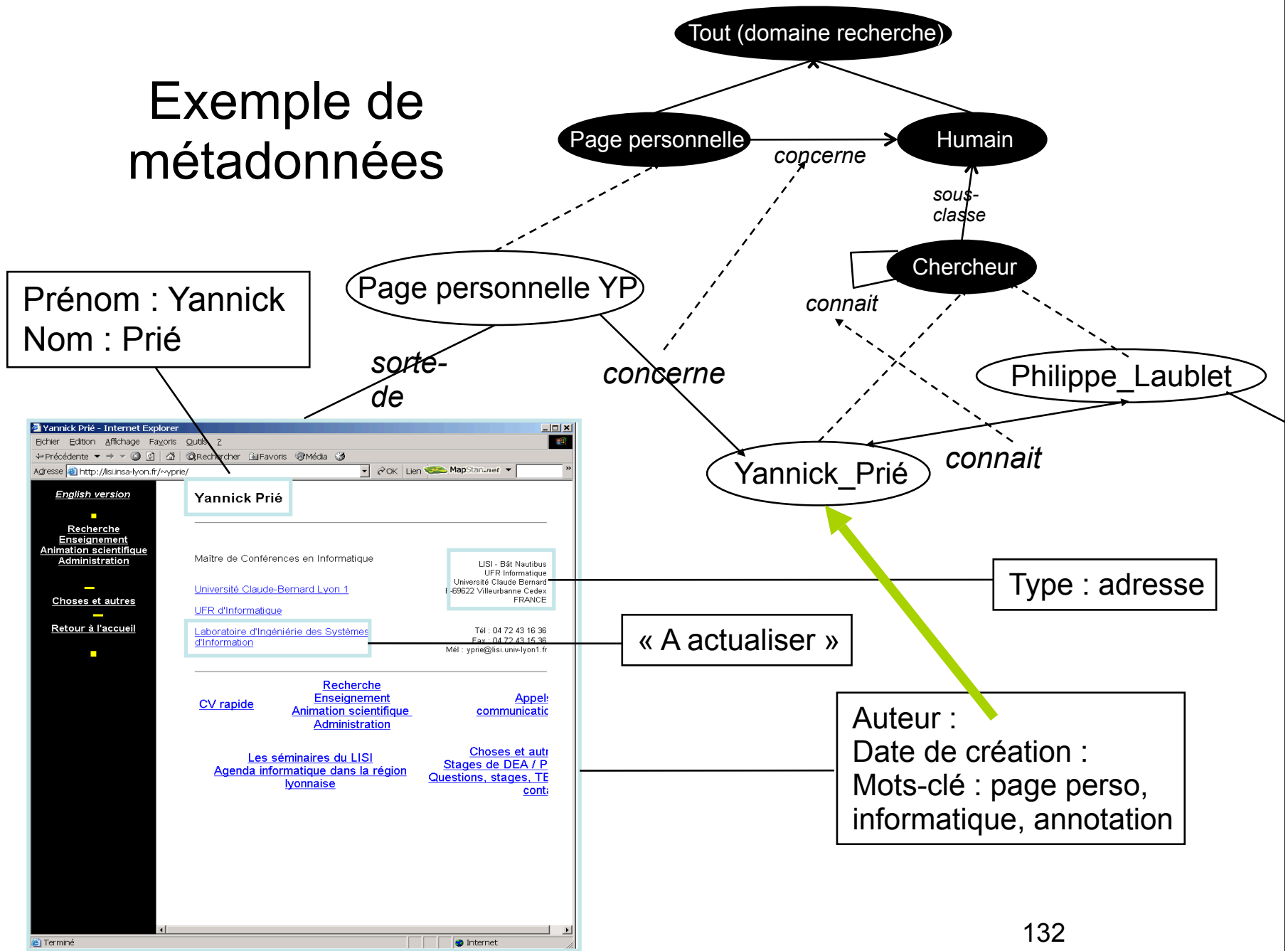
Web sémantique



Métadonnées / annotations

- Une définition
 - « information associée à une ressource du web, permettant d'en favoriser l'utilisation par un agent humain, du fait de son exploitation par un agent logiciel »
- Points importants
 - Séparation de la ressource et de l'annotation
 - Information structurée descriptive de la ressource
 - Traitements plus ou moins évolués par des programmes
 - Utilisateur plus ou moins conscient de l'exploitation des métadonnées (de leur existence)

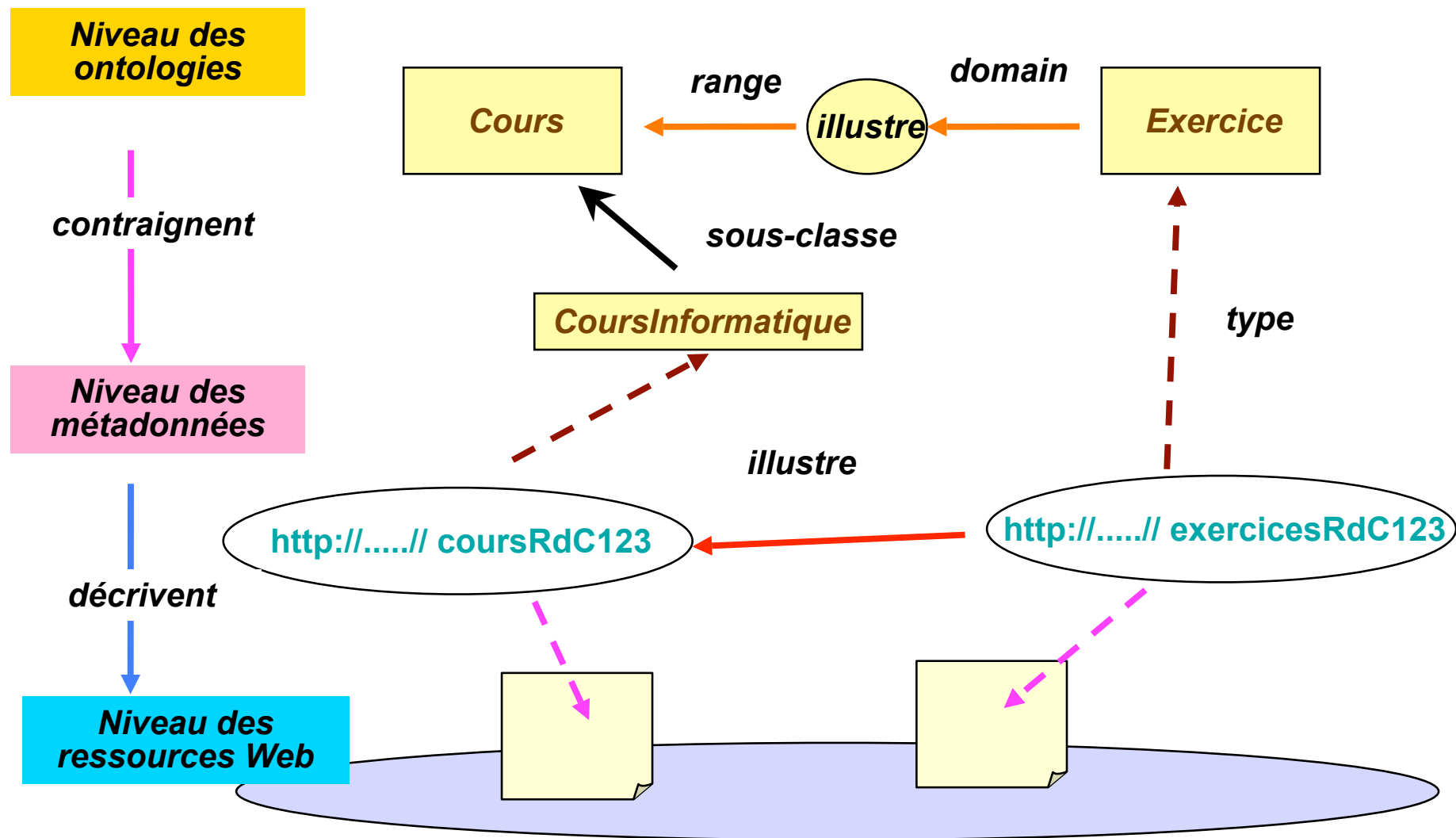
Exemple de métadonnées



Ontologies

- Modèles informatisables répertoriant les concepts et les notions d'un domaine en les formalisant
 - Représentation des connaissances
- Point de vue de l'ingénierie : en vue de leur utilisation dans des réalisations informatiques.
 - De ce point de vue, elles reflètent un certain niveau de connaissance sur un domaine
- Langages de représentation
 - OWL, RDFS, RDF

Ontologies, métadonnées et ressources



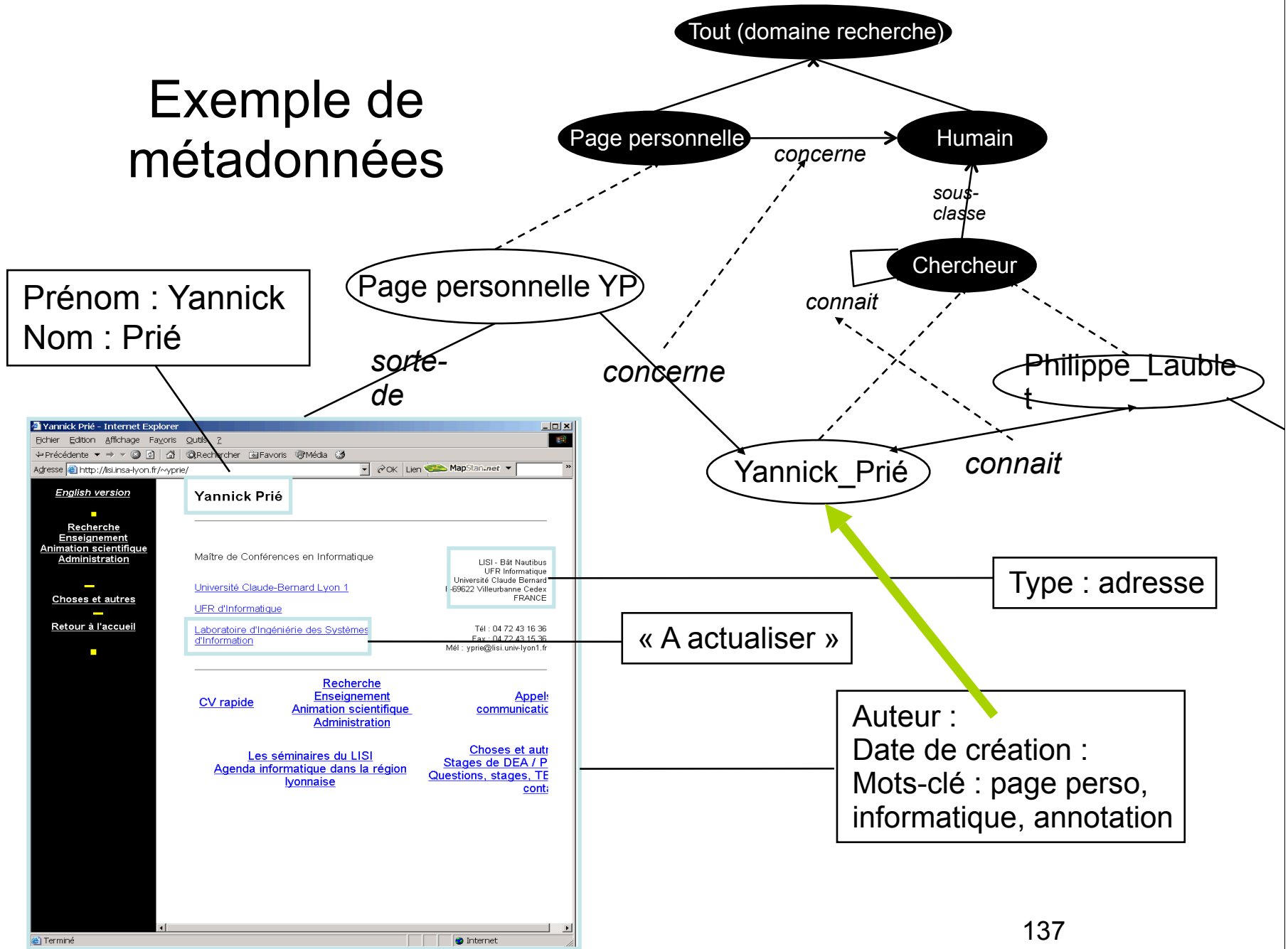
Familles d'application

- Recherche de ressources basée sur les structures relationnelles entre concepts
 - versus recherche basée sur des mots-clés
- Navigation sémantique
 - versus navigation basée sur des liens “anonymes” pour les logiciels
- Génération sémantique de ressources, personnalisation
 - versus une seule “taille” pour tout le monde
- Question - réponse en combinant des résultats
 - versus simple recherche de document
- Services
 - versus langages de descriptions de service, négociation, composition de services ...

Ressources sémantiques pour

- Décrire des ressources (métadonnées documentaires)
 - Conditions de productions, ... qui, quand, où, comment
 - Conditions d'utilisation
- Décrire des documents (métadonnées, annotations ...)
 - Cela parle de quoi
 - C'est structuré comment
 - Cela fait référence à quoi ...
 - ...
- Collaborer entre communautés (de travail, relationnelles,)
 - Notes de travail
 - Commentaires
 - Notes de « réputation »
- Structurer des ressources (liens hypertextes étiquetés) ... avec un certain niveau de formalisation

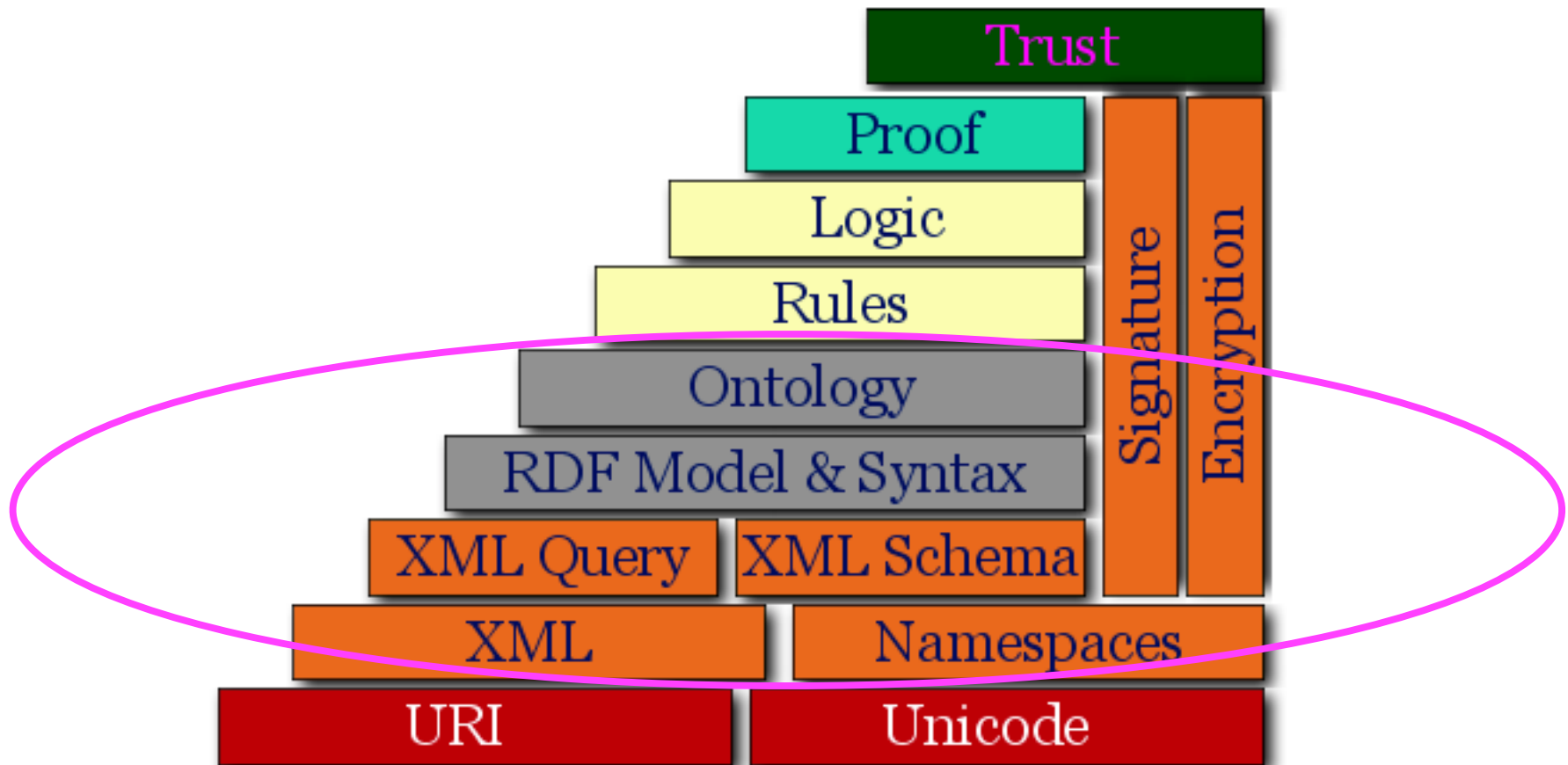
Exemple de métadonnées



Outils et langages

- Langages pour exprimer les descriptions
- Outils permettant
 - De créer ontologies et descriptions
 - éditeurs
 - outils d'annotation de textes, d'images
 - outils semi-automatiques
 - Des récolter et de stocker les descriptions
 - fichiers RDF, OWL
 - moissonneurs, bases de données
 - De les utiliser
 - y mener des recherches
 - langages de requêtes standardisés
 - les présenter
 - navigateurs sémantiques côté client
 - génération sémantique de sites côté serveur
 - De les échanger

La pile des langages (W3C)



Les « couches » du WS (1)

- XML : couche de transport syntaxique
 - transformation, échange, stockage, import/export
- RDF : réseau sémantique de base
 - triplets relationnels...
 - modèle de connaissances : description, intégration,
- RDFS : langage de classes, simple
 - représentation de structures conceptuelles et d'ontologies simples avec description hiérarchique des concepts et des propriétés
- OWL : langage de représentation d'ontologies plus complexes
 - raisonnement en plus

Les « couches » du WS (2)

- (couches plus prospectives...)
- Logique
 - Complète les langages d'ontologies
 - Connaissance déclarative spécifique à l'application
- Preuve
 - Génération de preuves
 - Validation
- Confiance
 - Signatures numériques
 - recommandations, organismes d'évaluation et de certification

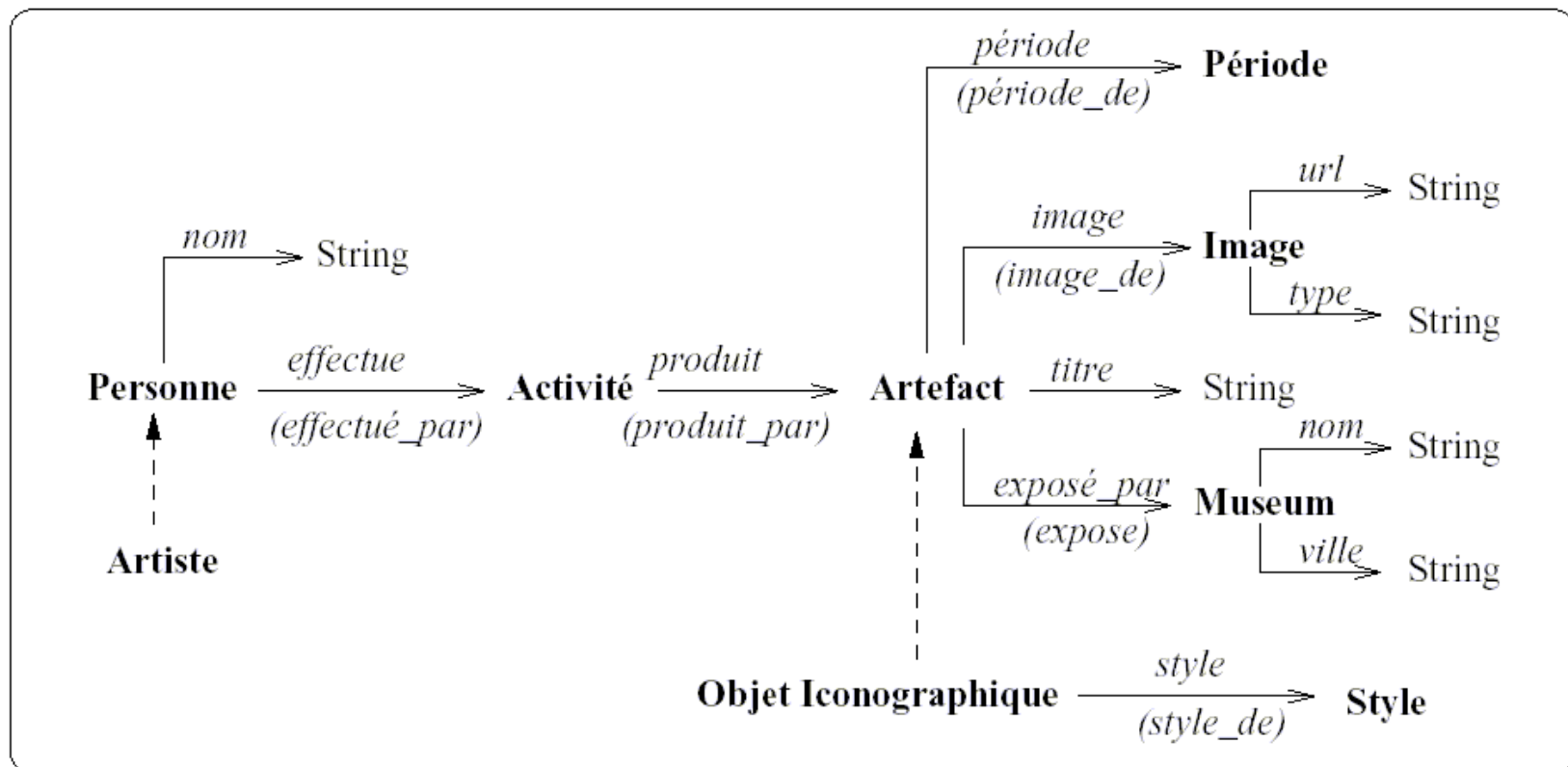
Représentation de connaissances simples - RDF

- La signification est exprimée
 - Sous forme de triplets
 - Comme le sujet, verbe et objet d'une phrase élémentaire
- En RDF (W3C), des assertions sont faites sur une ressource Web (voir plus loin),
 - Elle a des propriétés avec certaines valeurs : une valeur de base ou une autre ressource Web
 - D'autres langages peuvent être utilisés MAIS ... on préfère
 - Un langage standard
 - L'insertion dans les outils du Web : la galaxie XML ...
- Aspect relationnel premier
 - Possibilité d'ajouter des triplets descriptifs sans difficulté
 - Mais des représentations plus complexes seront proposées

Représentation de connaissances - RDFS

- RDF Schema = extension de RDF
- Langage de description des classes ou des types des éléments de RDF
- Décrit donc des classes et les types de propriétés autorisées entre les instances de classes qui elles sont en RDF
- Fournit des hiérarchies de généralisation de propriétés ou de classes

Modèle conceptuel simple exprimable en RDFS



D'après Bernd Amman

Composants typiques d'une ontologie pour le WS

en OWL
ou en RDF(S)

- Concepts ou classes
 - *Enseignant, Cours, CoursInformatique*
 - Relations de généralisation / spécialisation entre concepts
- Relations entre concepts ou « propriétés objets »
 - *X enseigne Y*
 - ObjectProperties en OWL
- Les attributs des concepts ou « propriétés types de données »
 - *X a comme e-mail « philippe.laublelet@paris4.sorbonne.fr »*
 - DataTypesProperties en OWL

Composants typiques d'une ontologie

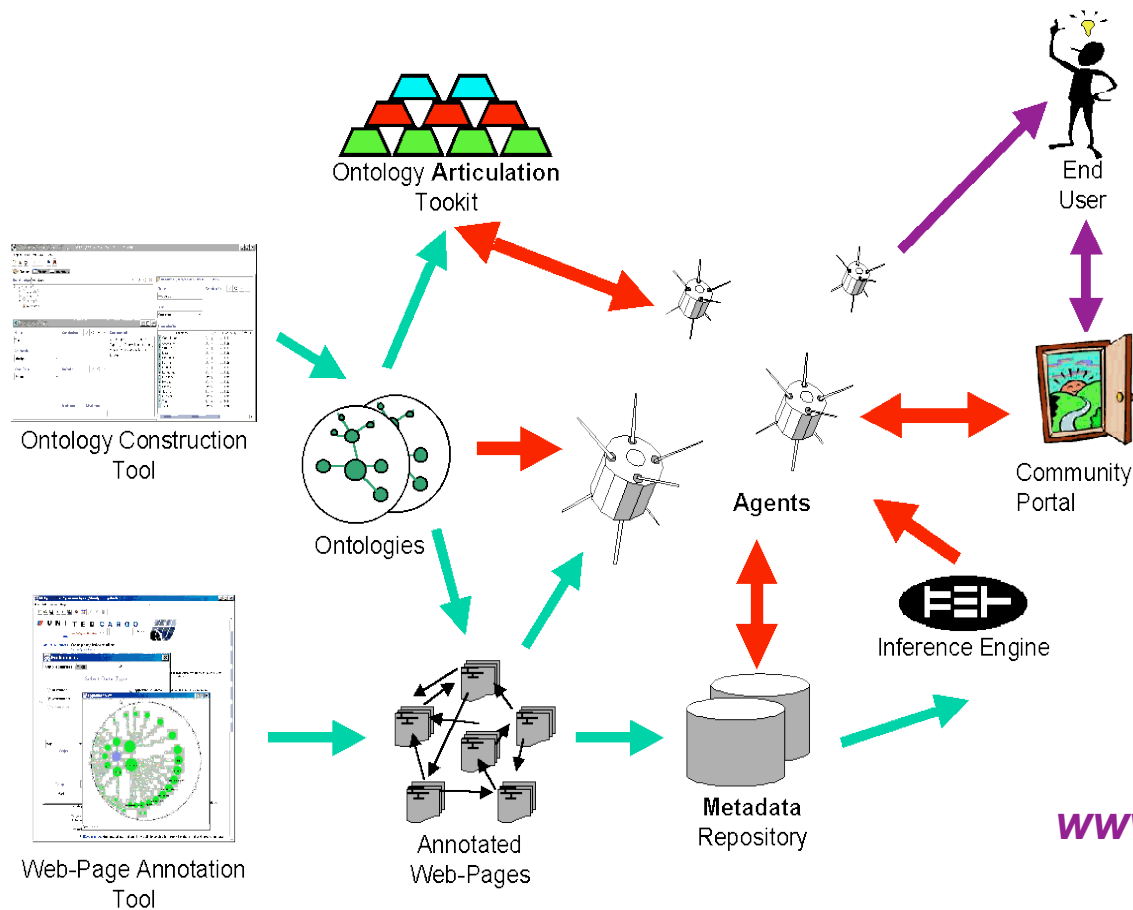
- Contraintes diverses ou axiomes restreignant l'interprétation des éléments
 - *Seuls les membres de l'université peuvent enseigner des cours*
 - *Professeur et Etudiant sont disjoints*
 - *Chaque département doit avoir au moins 10 enseignants*
- Construction de classes complexes
 - Utilisation de connecteurs logiques
 - *EtablissementEnseignement et PoleDeRecherche*
 - Définitions ...
 - *Université dans un grande ville avec au moins un cours de linguistique*
 - *Ces constructions sont utilisées par les mécanismes inférentiels*

en OWL
mais pas en
RDF(S)

Défis du WS pour la représentation des connaissances

- Convergence / standardisation / pragmatisme
 - Apport de « millions d'ingénieurs »
 - Solutions partielles déjà intéressantes
 - Tests instantanés
- Variété / richesse du web
 - Taille du web
 - RDF/OWL : 100 K pages (2005)
 - important mais environ 0,01 % du Web
 - Utilisation parfois de mécanismes inférentiels
 - Variété
 - Beaucoup d'auteurs
 - Qui fait « autorité » ? Questions de confiance
 - Beaucoup d'ontologies / vocabulaires, qualité variée
 - Comment les articuler ?
 - Utilisation imprévisible
 - Décentralisation et évolution permanente du web

Ontologies, métadonnées, agents logiciels et ... utilisateur



www.semanticweb.org

Plan

- Représentation de connaissances
- Ingénierie des connaissances
- Ontologies
- Graphes conceptuels
- Logiques de description
- Introduction au Web sémantique
- **OWL**
- **Conception d'ontologies**

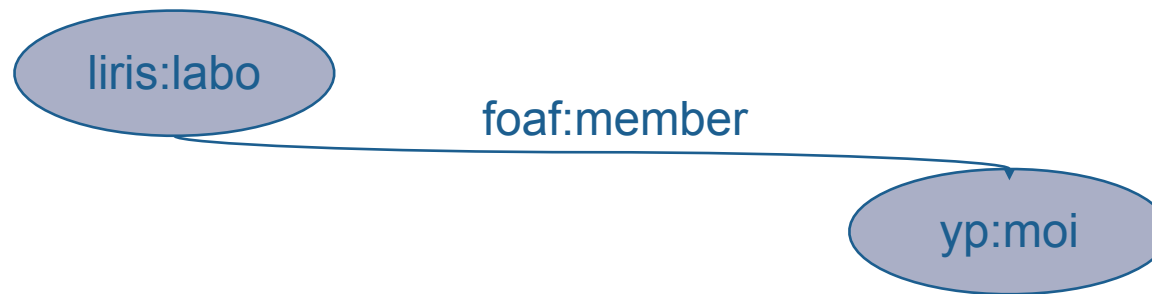
RDF

- Resource Description Framework
 - Recommandation du W3C
 - Description de métadonnées, proche des réseaux sémantiques
 - XML : unification de la syntaxe
 - RDF : unification de la sémantique
- Modèle
 - Termes du langage :
 - URI (\approx URL)
 - littéraux
 - nœuds vierges (*blank*)
 - Structure d'un énoncé : sujet prédicat objet

RDF (triplet)

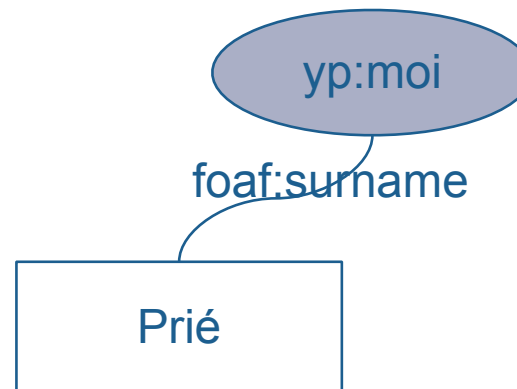


RDF (notation)



foaf: <http://xmlns.com/foaf/0.1/>
liris: <http://liris.cnrs.fr/>
yp: <http://liris.cnrs.fr/yannick.prie/>

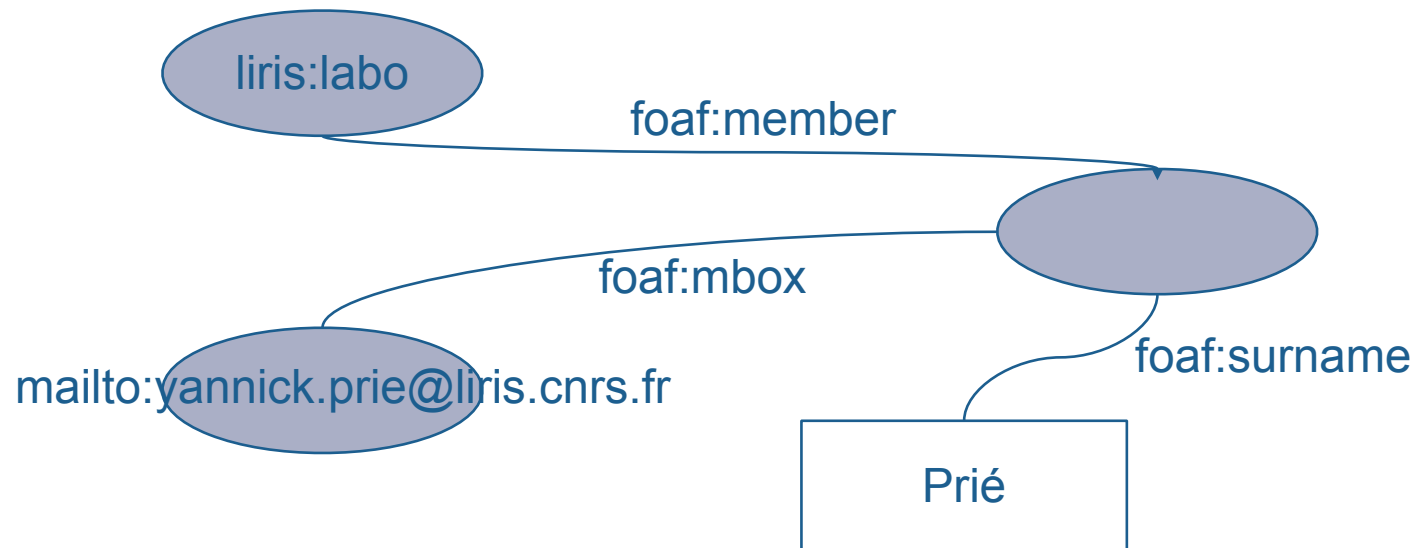
RDF (littéral)



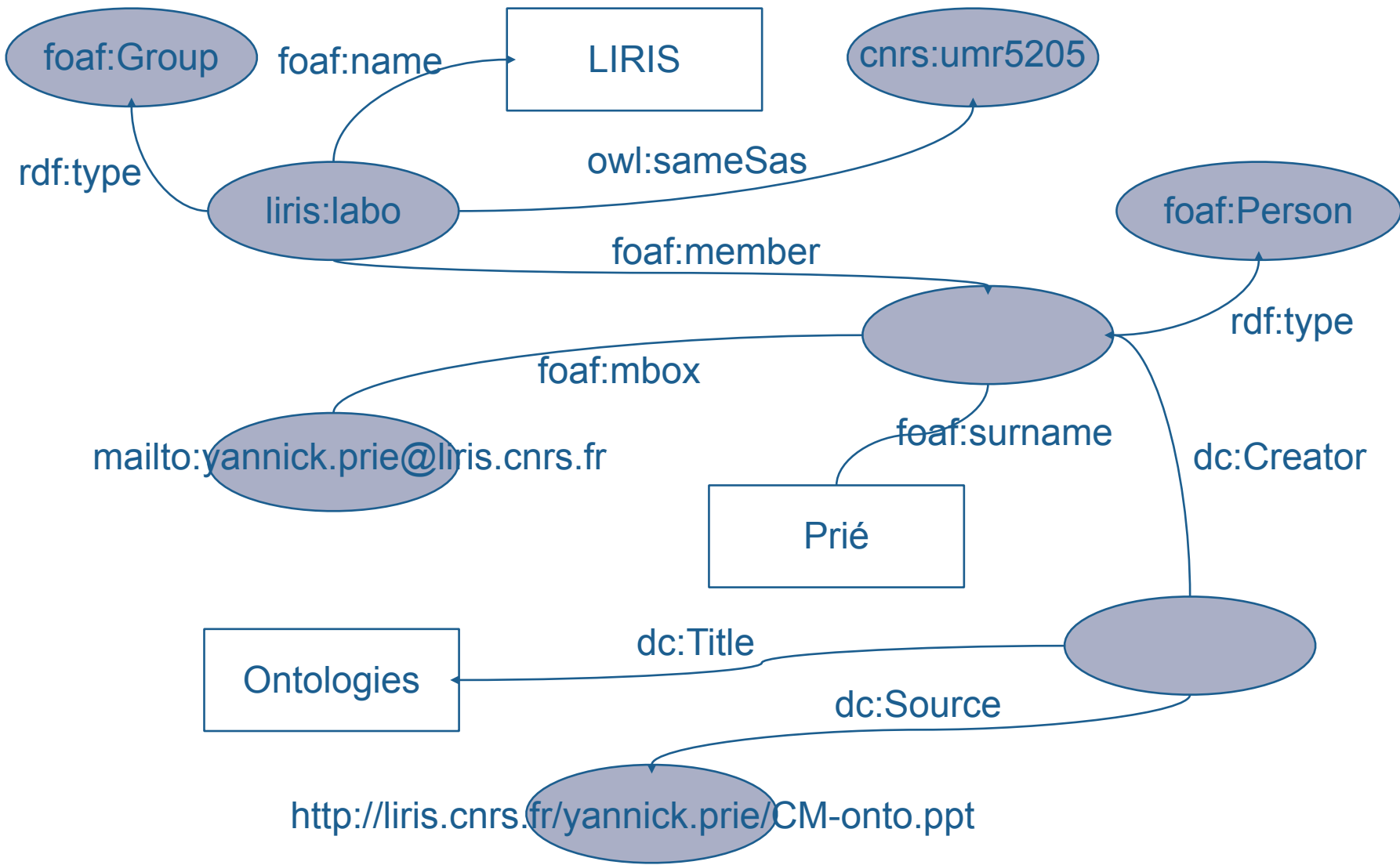
RDF (nœud vierge)



RDF (exemple)



RDF (exemple)



RDF (syntaxe RDF/XML)

```
<foaf:Group rdf:about="&liris;labo">
  <foaf:name>LIRIS</foaf:name>
  <owl:sameAs rdf:resource="&cnrs;umr5205" />
  <foaf:member>
    <foaf:Person rdf:nodeID="yp">
      <foaf:surname>Prié</foaf:surname>
    </foaf:Person>
  </foaf:member>
</foaf>
```

```
<rdf:Description>
  <dc:Title>
    Ontologies
  </dc:Title>
  <dc:Creator rdf:nodeId="yp" />
  <dc:Source rdf:resource=
    "http://liris.cnrs.fr/ens/CM-onto.ppt">
```

RDF (syntaxe Notation3)

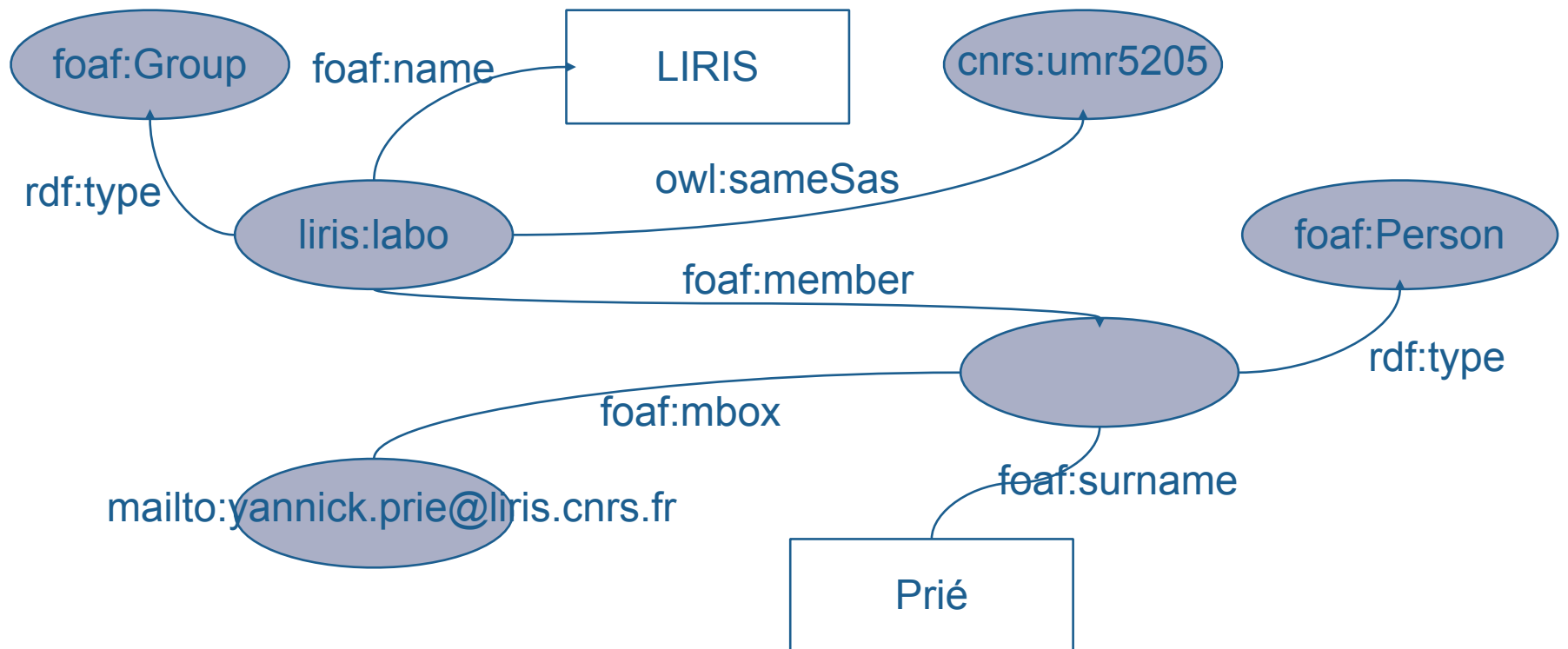
```
liris:labo rdf:type foaf:Group ;
            foaf:name "LIRIS" ;
            owl:sameAs cnrs:umr5205 ;
            foaf:member _:pac .

_:pac rdf:type foaf:Person ;
       foaf:surname "Champin" .

[
  dc>Title "Interopérabilité sémantique..." ;
  dc:Creator _:pac ;
  dc:Source <http://liris.cnrs.fr/mccir-050127.sxi>
].
```

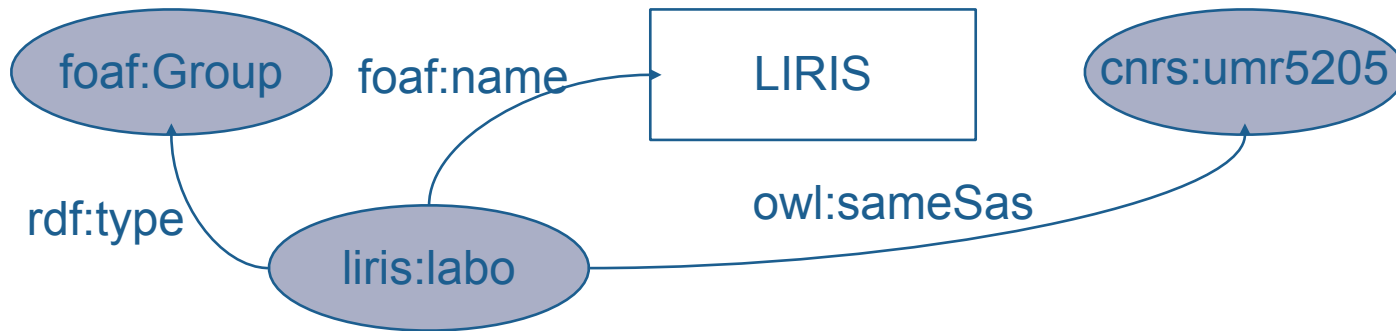
RDF – Sémantique (1)

- Graphe conséquence d'un autre graphe



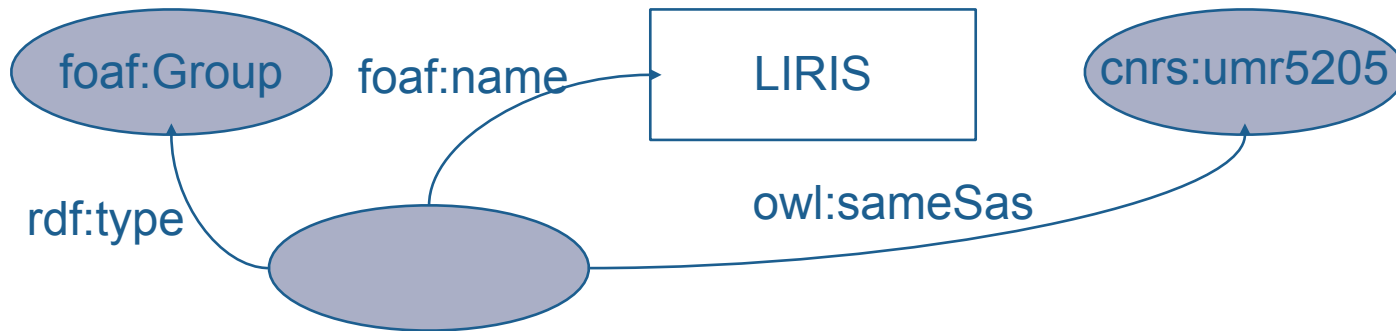
RDF – Sémantique (2)

- Tout sous-graphe est une conséquence :
sémantique **monotone**



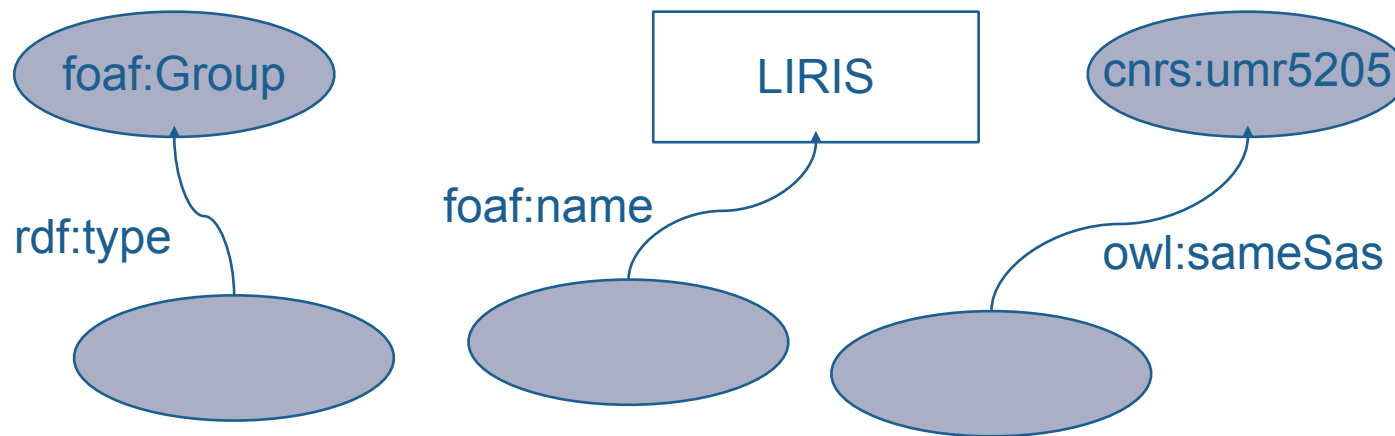
RDF – Sémantique (3)

- Substitution des nœuds vierges



RDF – Sémantique (3)

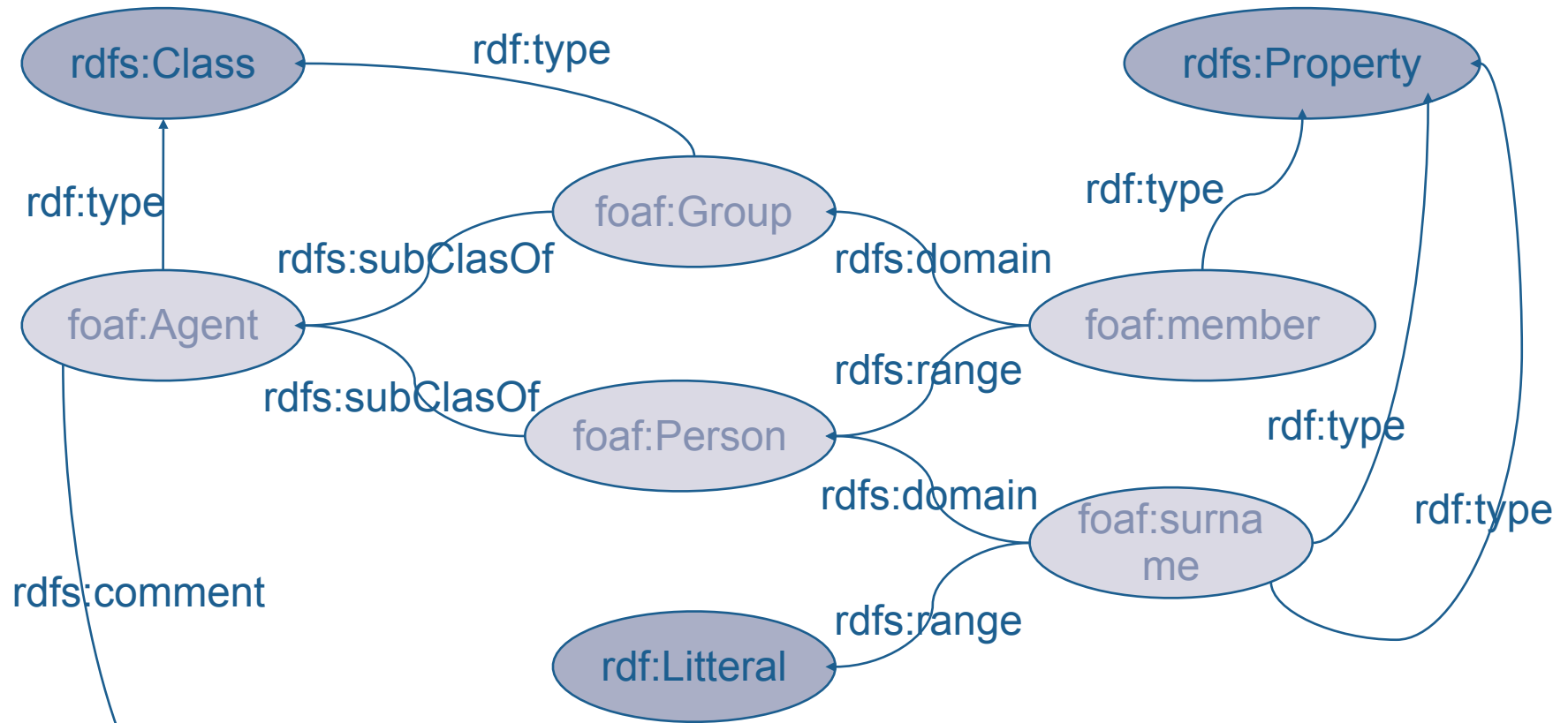
- Éclatement des nœuds vierges



RDF Schema

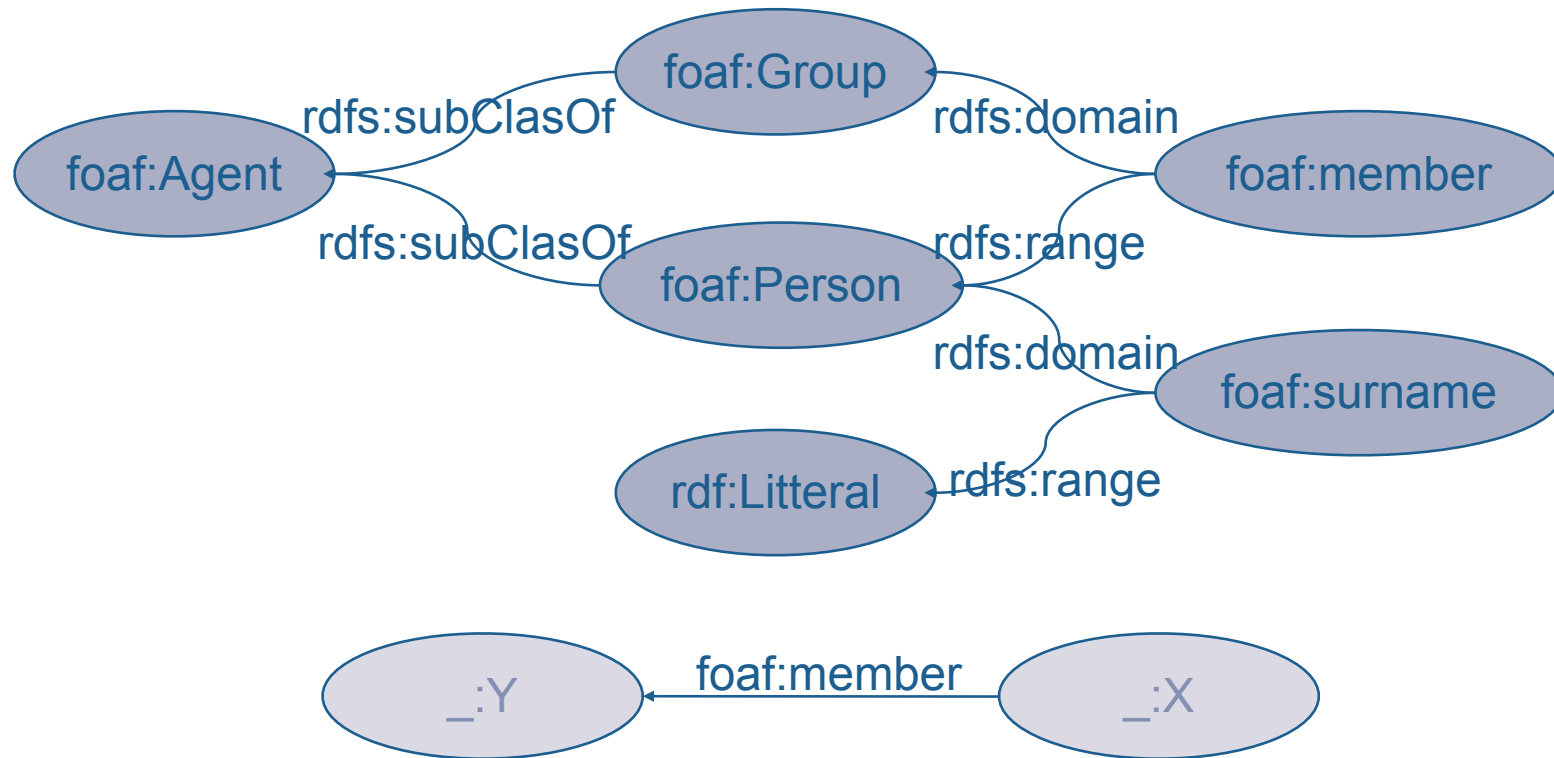
- Vocabulaire pour
 - décrire des classes (concepts),
 - décrire des propriétés (rôles),
 - les structurer en hiérarchie de spécialisation (\sqsubseteq),
 - contraindre le domaine et la portée des propriétés (\forall limité).
 - peupler une telle ontologie (A-Box).
- Sémantique d'une LD très simple

RDF Schema (example)



An agent (eg. person, group, software or physical artifact).

RDF Schema (exercice)



X est un Groupe ? Y est une Personne ? X est un Agent ?

<http://www.w3.org/TR/rdf-mt/>

RDF Schema

- En fait, RDF Schema n'est pas strictement une LD : les familles de termes (classes, propriétés, individus) ne sont **pas** distinctes
 - :Dumbo rdf:type :Éléphant
 - :Éléphant rdfs:subClassOf :Mammifère
 - :Éléphant rdf:type :Espèce
 - :Espèce rdf:type rdfs:Class
- Pas gênant car il n'y a pas de termes complexes, et notamment pas de négation
 - on ne peut pas exprimer de contradiction en RDFS
 - les mécanismes d'inférence restent décidables

OWL – Présentation (1)

- Recommandation du W3C
<http://www.w3.org/2004/OWL/>
- Expressivité des LD en RDF
 - réutilise le vocabulaire de RDF Schema
 - logique très expressive (famille SH)
 - domaines concrets (littéraux RDF)
 - implémentations performantes disponibles
 - FaCT <http://www.cs.man.ac.uk/~horrocks/FaCT/>
 - Racer <http://www.racer-systems.com/>
 - Pellet <http://www.mindswap.org/2003/pellet/>

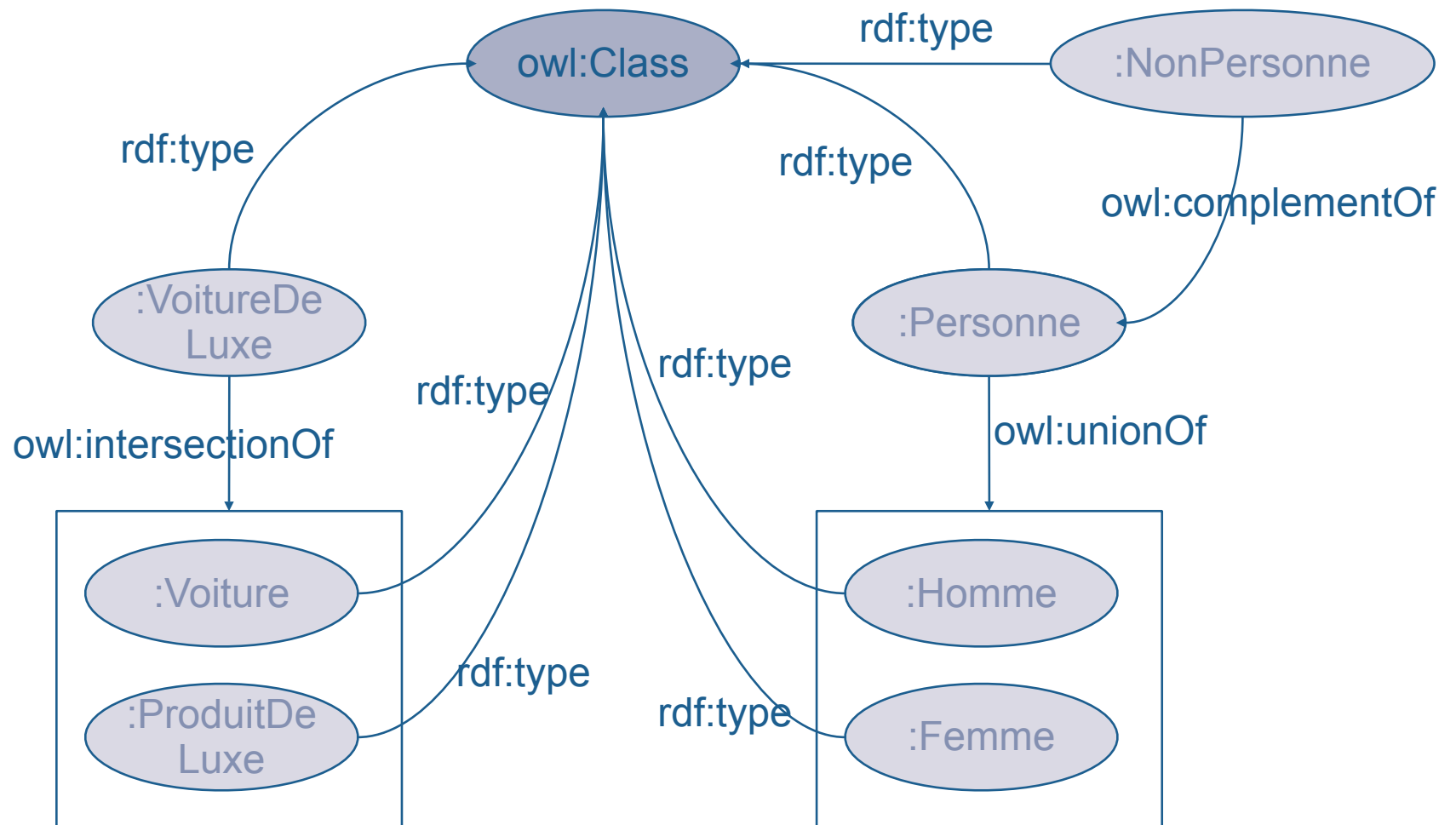
OWL – Présentation (2)

- Différences avec les LD
 - utilise les URIs comme termes
 - différence de méta-terminologie (RDF Schema)
 - concept → classe
 - rôle → propriété
 - les axiomes s'expriment en RDF
 - non atomiques (triplets)
 - sémantique d'un triplet isolé ?

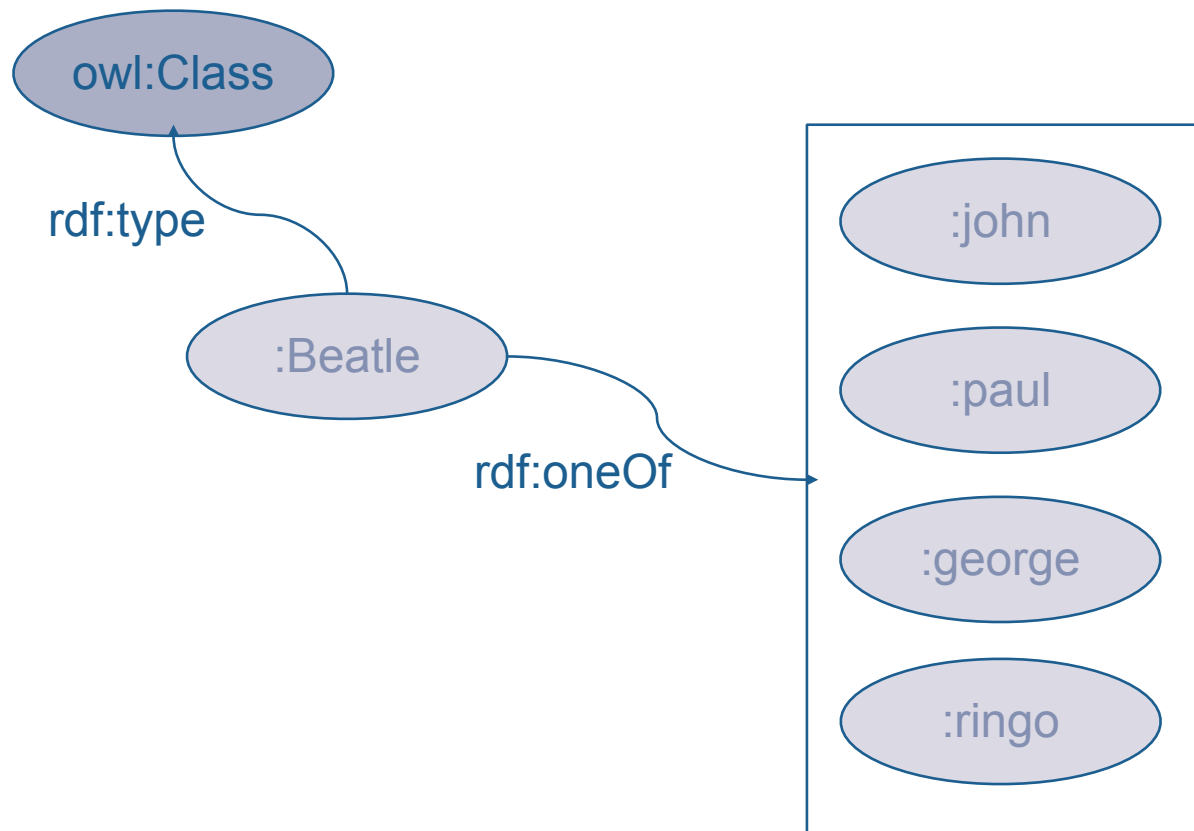
OWL

- Hiérarchies de Classes
- Hiérarchies de Propriétés
 - Types de données simples
 - « Propriétés objets (relations entre classes) »
- Descriptions de classes
 - Peuvent être utilisées au lieu de classes nommées
 - Énumérations
 - une EstimationDeCrise est soit normal, soit dommagesMatériels, soit grave, soit trèsGrave
 - Restrictions
 - une Ville est un LieuGéo qui aCommeHabitants au moins 3000
 - Enoncés logiques
 - quelquechose qui soit RouteSecours et pas RouteTerrestre

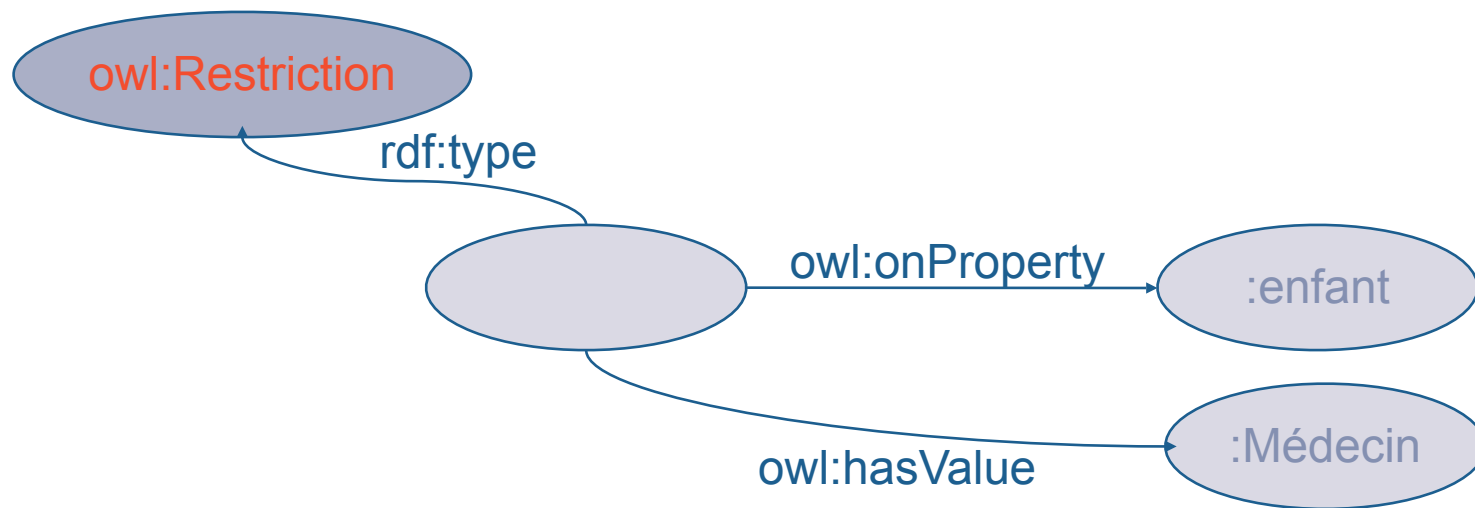
Classes OWL (op. ensemblistes)



Classes OWL (extension)

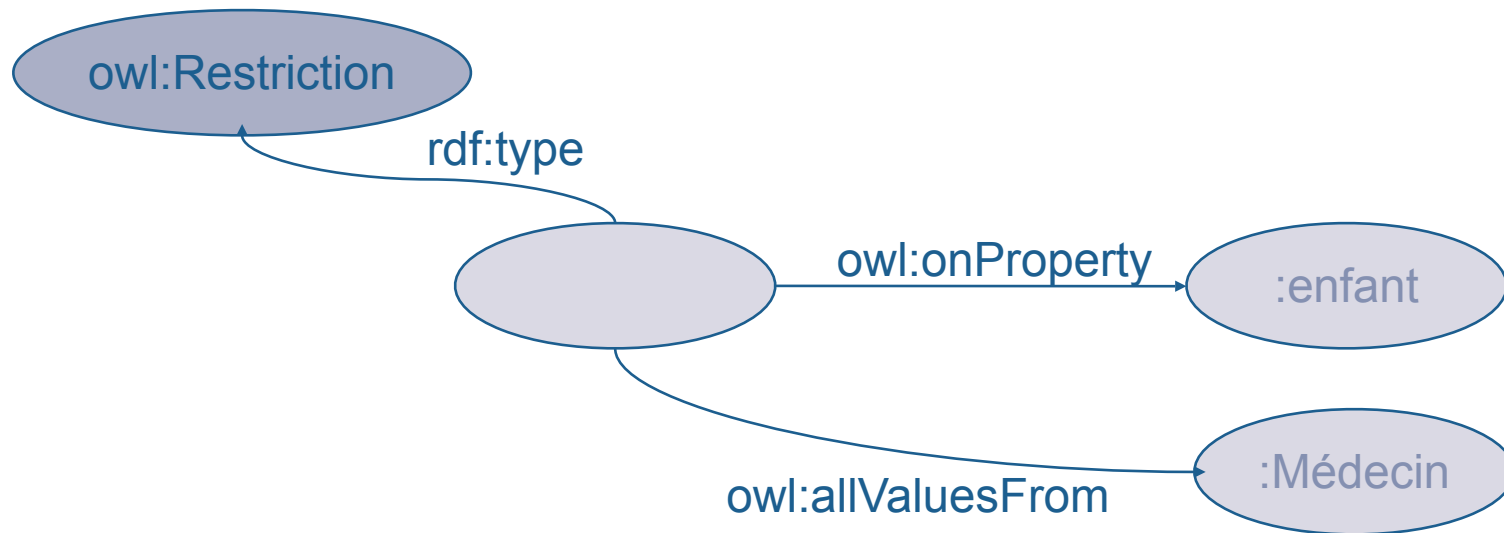


Classes OWL (restrictions)



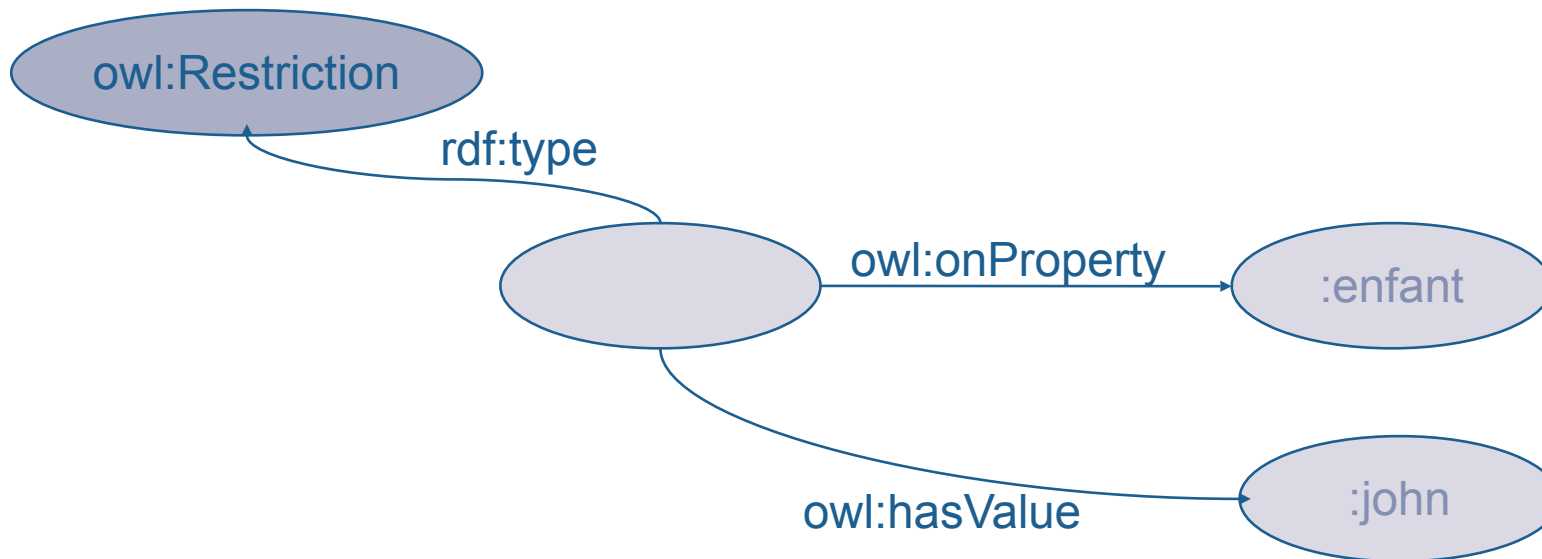
\exists enfant.Médecin

Classes OWL (restrictions)



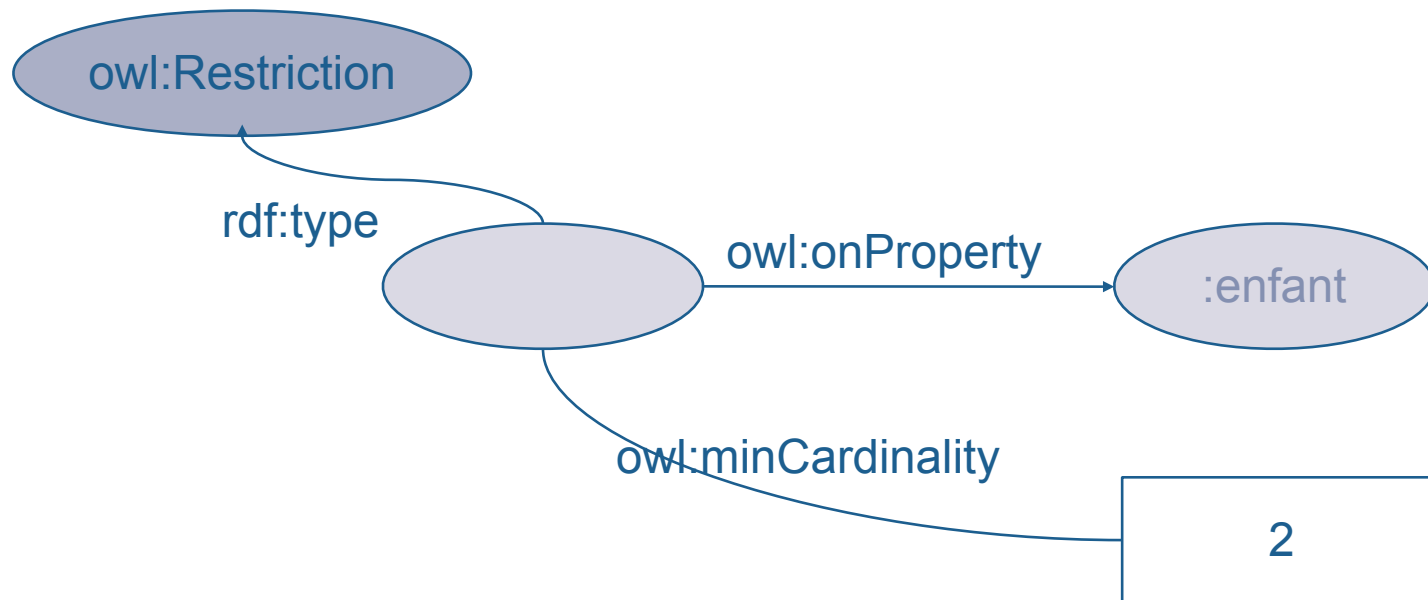
\forall enfant.Médecin

Classes OWL (restrictions)



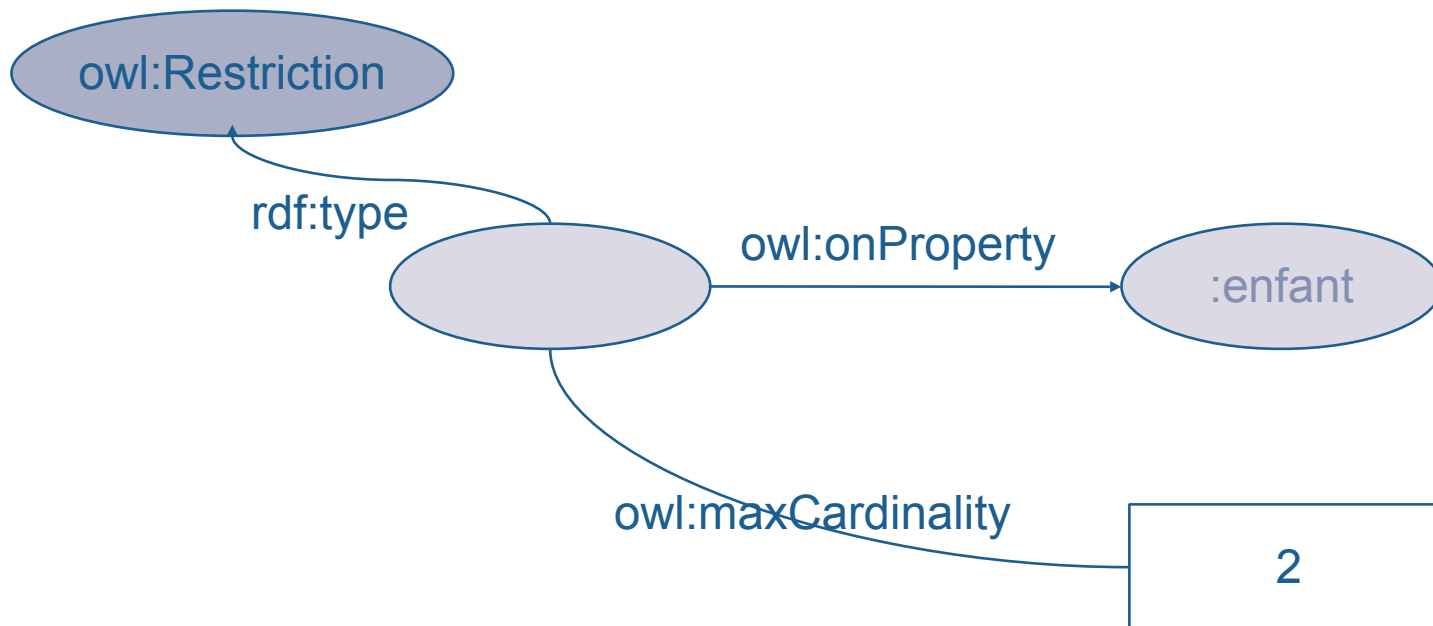
\exists enfant.{john}

Classes OWL (restrictions)



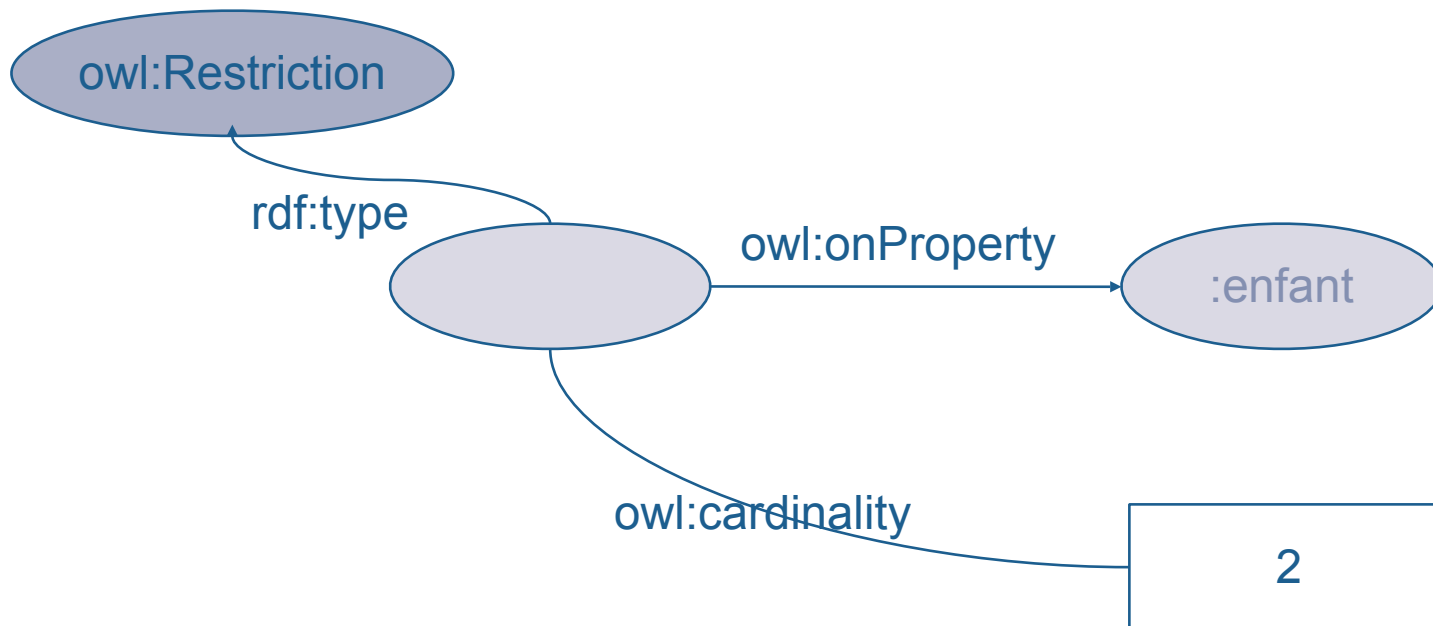
(≥ 2 enfant)

Classes OWL (restrictions)



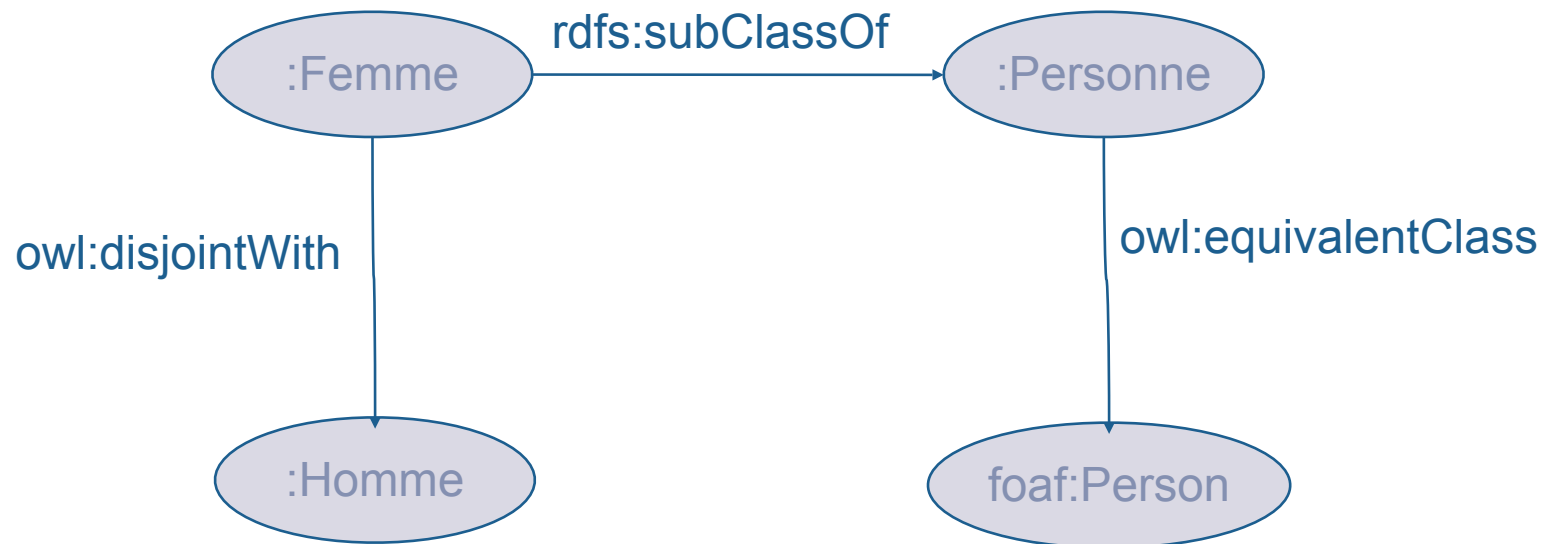
(≤ 2 enfant)

Classes OWL (restrictions)



(= 2 enfant)

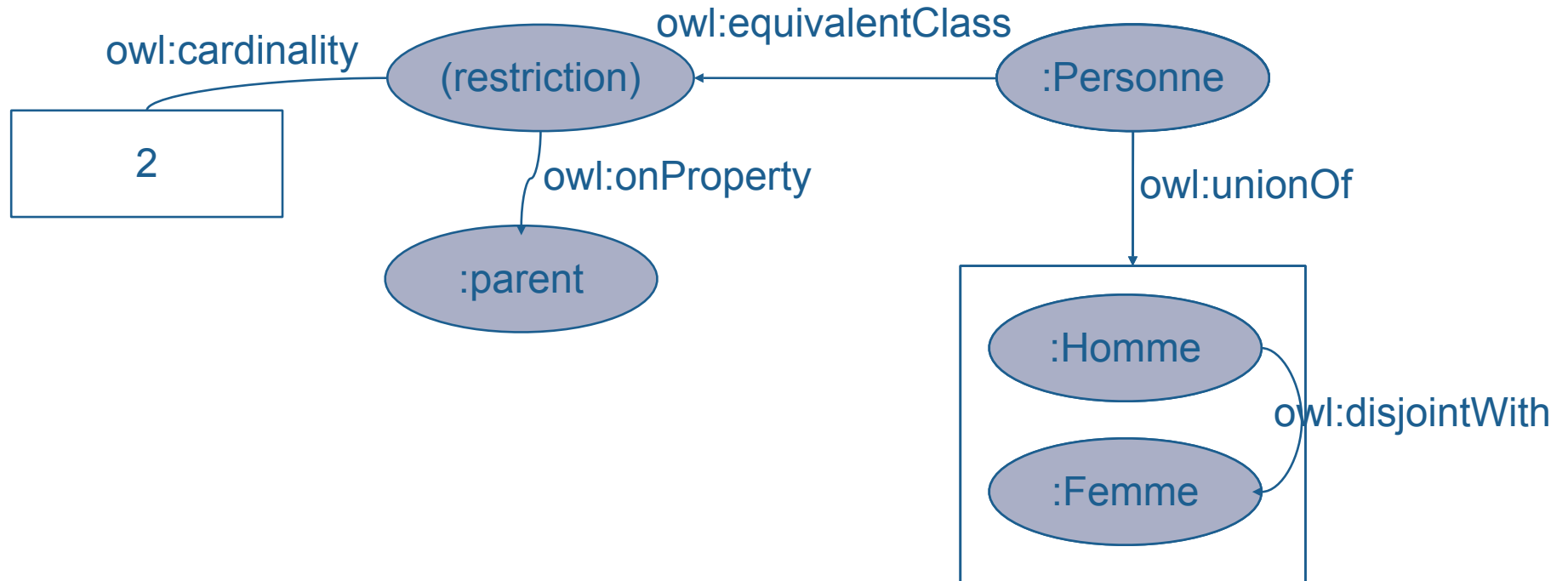
Relations entre classes OWL



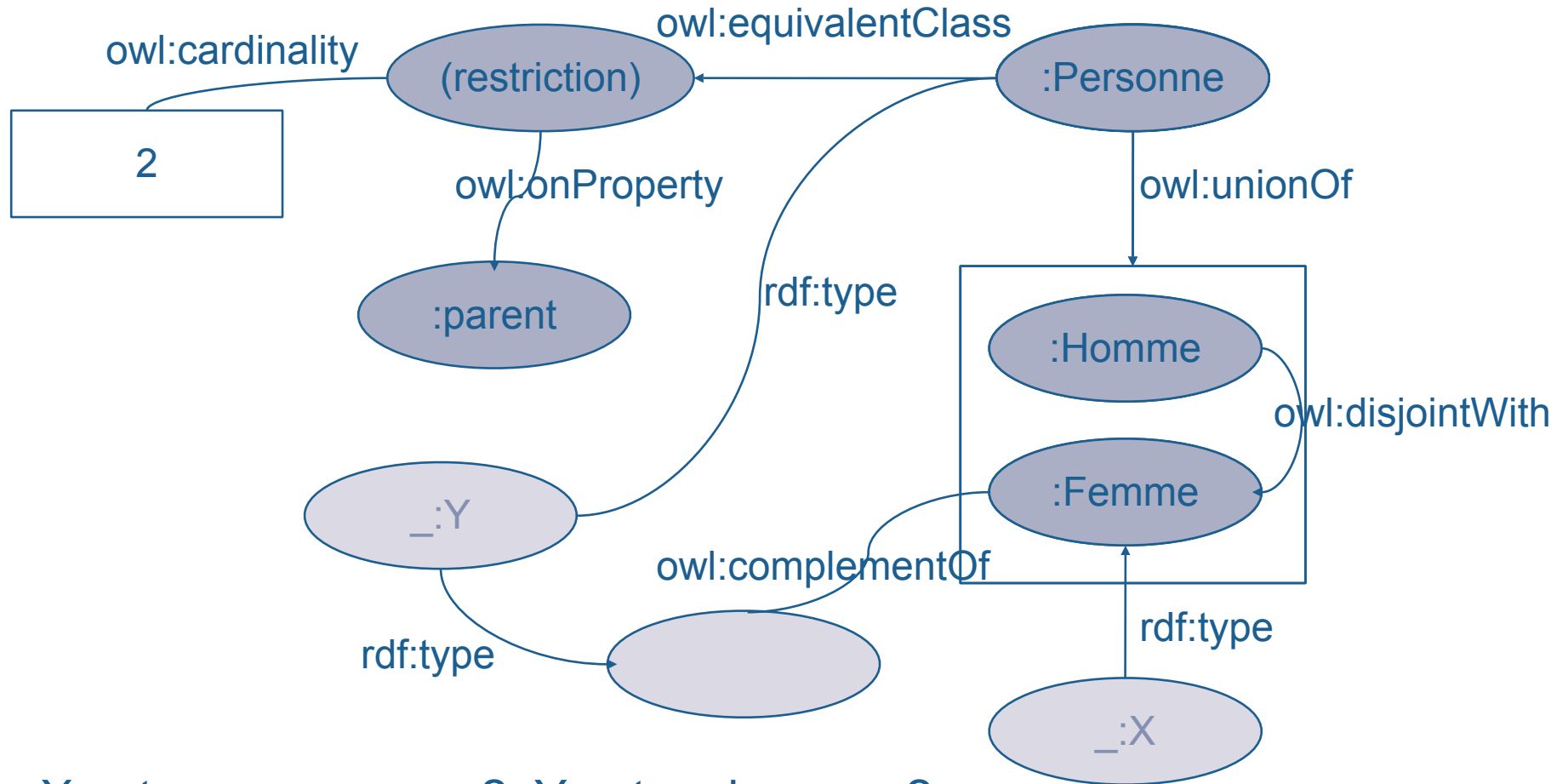
Propriétés

- Relation entre propriétés
 - Inverse : owl:inverseOf
 - Equivalence : owl:equivalentProperty
- Propriétés spéciales
 - Transitivité : owl:TransitiveProperty
 - Symétrie : owl:SymmetricProperty
 - Fonctionnelle : owl:FunctionalProperty
 - un seul individu en relation par cette propriété
 - Inverse fonctionnelle : owl:InverseFunctionalProperty

Classes OWL (exercices)

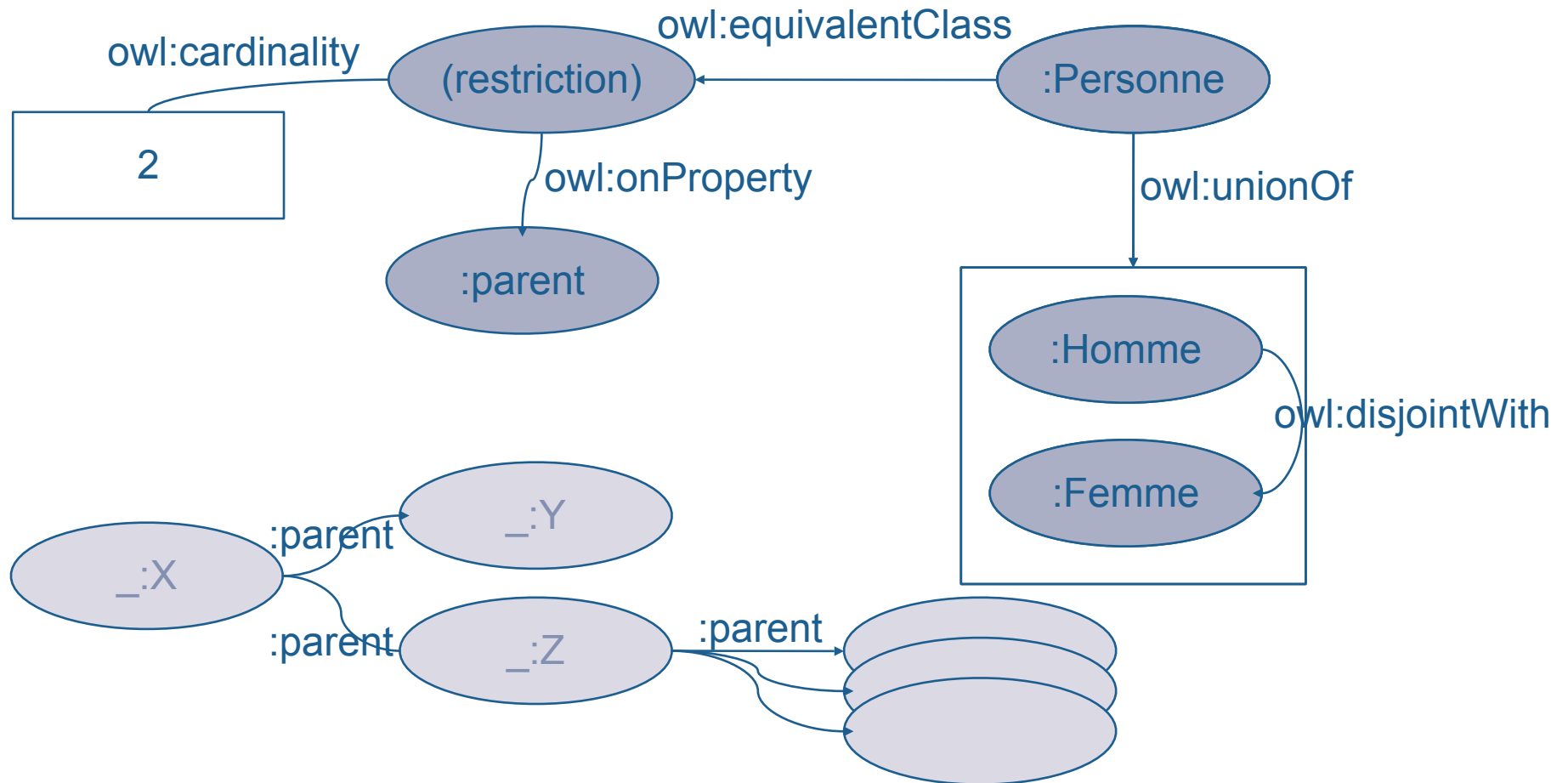


Classes OWL (exercices)



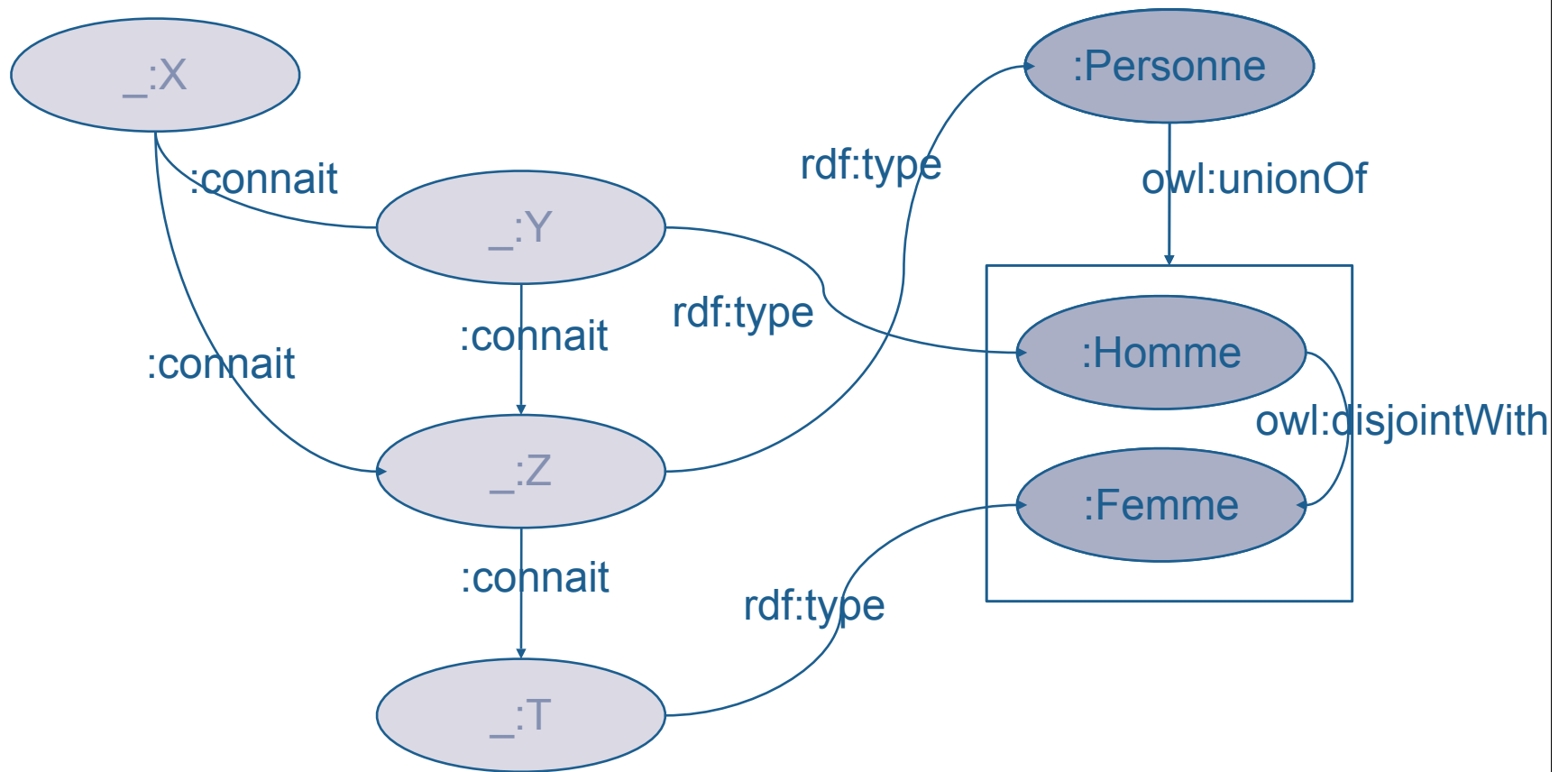
X est une personne ? Y est un homme ?

Classes OWL (exercices)



X est une personne ? Y est une personne ? Z est une personne ?

Classes OWL (exercices)



X connaît un homme qui connaît une femme ?

Syntaxe XML / OWL

```
<owl:Class rdf:ID= "Latitude">
  <rdfs:subClassOf rdf:resource="#MesureGeo" />
</owl:Class>
<owl:ObjectProperty rdf:ID="aPourEstimation">
  <rdf:type rdf:resource="#owl:FunctionalProperty" />
  <rdfs:domain rdf:resource="#LieuGeo"/>
  <rdfs:range rdf:resource="#EstimationDeCrise"/>
</owl:ObjectProperty>
<owl:Class rdf:ID="VilleNormale">
  <owl:intersectionOf rdf:parseType="Collection">
    <owl:Class rdf:about="#Ville" />
    <owl:Restriction>
      <owl:onProperty rdf:resource="#aPourEstimation" />
      <owl:hasValue rdf:resource="#normal" />
    </owl:Restriction>
  </owl:intersectionOf>
</owl:Class>
```


OWL – Modularité (1)

- Utilisation d'URIs comme identificateurs
 - pas de collision de noms entre ontologies
 - permet de faire référence à des termes d'une autre ontologie
 - exemple : foaf utilise le vocabulaire de wordnet, wot utilise le vocabulaire de foaf
 - ATTENTION : pas de sémantique définie
 - possibilité d'**importer** une ontologie
 - inclusion des axiomes de l'ontologie importée

Différentes versions d'OWL

- OWL Lite
 - Hiérarchie de classification
 - Contraintes simples
 - Outils inférentiels plus simples et efficaces
- OWL DL
 - Correspond à certaines logiques de descriptions
 - Complétude
 - Décidabilité
- OWL Full
 - Une classe peut être considérée comme instance d'une classe
 - Complètement compatible avec RDFS
- Les implémentations OWL
 - <http://www.w3.org/2001/sw/WebOnt/impls>

OWL Lite

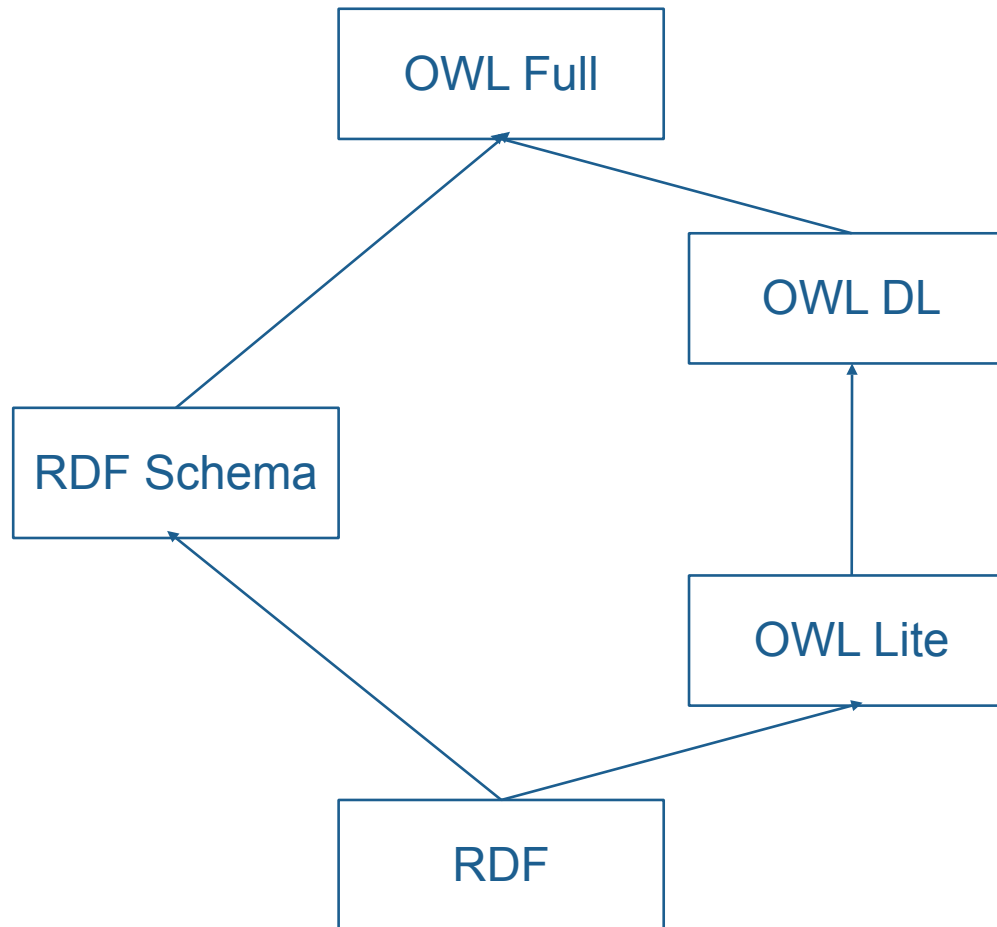
- RDF Schema Features:
 - [Class](#)
 - [rdf:Property](#)
 - [rdfs:subClassOf](#)
 - [rdfs:subPropertyOf](#)
 - [rdfs:domain](#)
 - [rdfs:range](#)
 - [Individual](#)
- (In)Equality:
 - [equivalentClass](#)
 - [equivalentProperty](#)
 - [sameAs](#)
 - [differentFrom](#)
 - [allDifferent](#)
- Property Characteristics:
 - [inverseOf](#)
 - [TransitiveProperty](#)
 - [SymmetricProperty](#)
 - [FunctionalProperty](#)
 - [InverseFunctionalProperty](#)
- Property Type Restrictions:
 - [allValuesFrom](#)
 - [someValuesFrom](#)
- Restricted Cardinality:
 - [minCardinality](#) (only 0 or 1)
 - [maxCardinality](#) (only 0 or 1)
 - [cardinality](#) (only 0 or 1)
- Header Information:
 - [ontology](#)
 - [imports](#)
- Class Intersection:
 - [intersectionOf](#)
- Versioning:
 - [versionInfo](#)
 - [priorVersion](#)
 - [backwardCompatibleWith](#)
 - [incompatibleWith](#)
 - [DeprecatedClass](#)
 - [DeprecatedProperty](#)
- Annotation Properties:
 - [rdfs:label](#)
 - [rdfs:comment](#)
 - [rdfs:seeAlso](#)
 - [rdfs:isDefinedBy](#)
- Datatypes
 - [DatatypeProperty](#)

OWL DL and FULL

- Class Axioms:
 - [oneOf, dataRange](#)
 - [disjointWith](#)
 - [equivalentClass](#)
(applied to class expressions)
 - [rdfs:subClassOf](#)
(applied to class expressions)
- Boolean Combinations of Class Expressions:
 - [unionOf](#)
 - [intersectionOf](#)
 - [complementOf](#)
- Arbitrary Cardinality:
 - [minCardinality](#)
 - [maxCardinality](#)
 - [cardinality](#)
- Filler Information:
 - [hasValue](#)

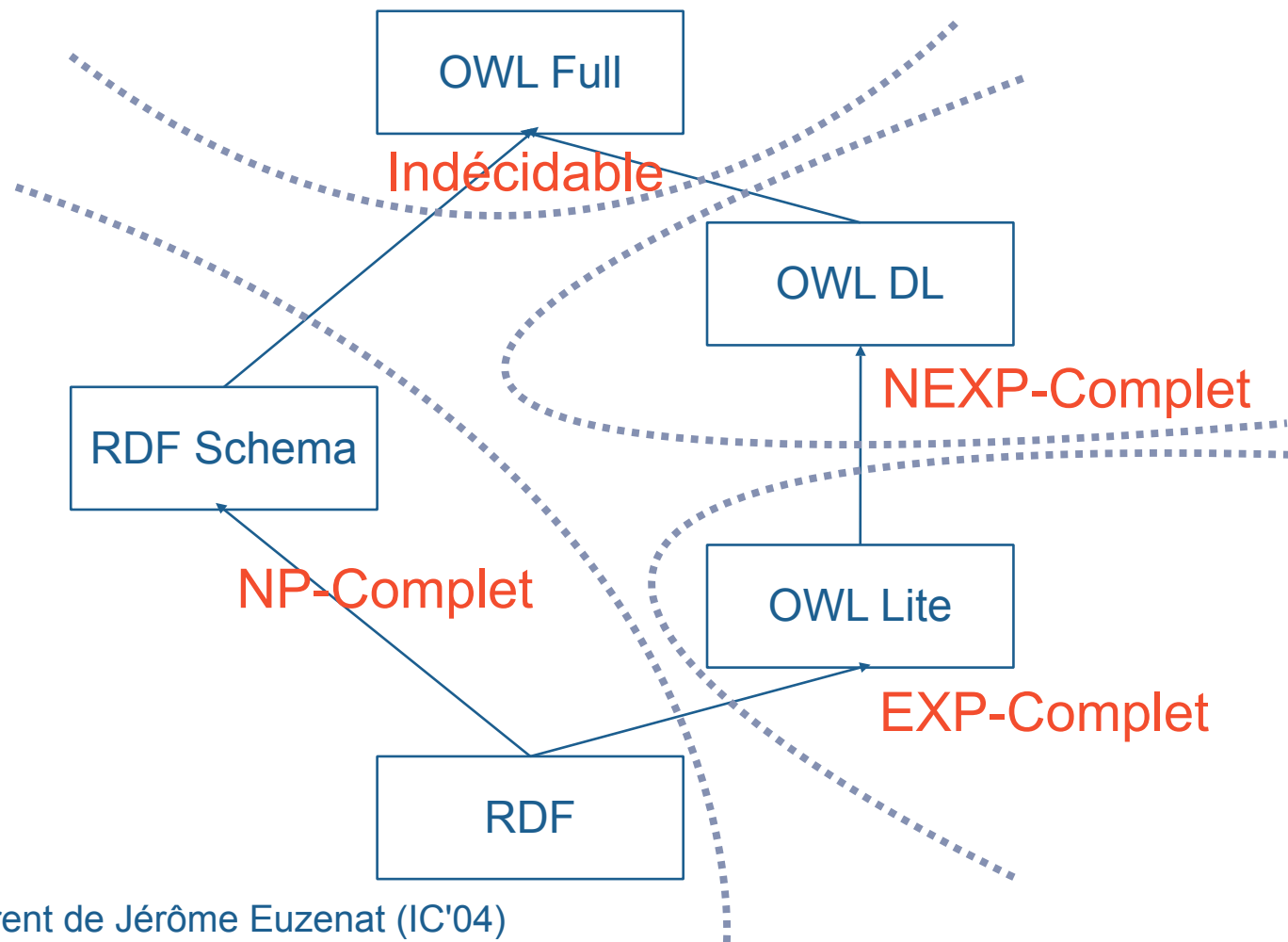
Quelques restrictions pour OWL DL par rapport à OWL FULL

OWL – Modularité (2)



D'après un transparent de Jérôme Euzenat (IC'04)

OWL – Modularité (3)



D'après un transparent de Jérôme Euzenat (IC'04)

OWL 1.1

- En cours de spécification
 - Restrictions de cardinalité qualifiées
 - Composition limitée de rôles
 - ...
- Moteurs d'inférences en cours d'implémentation
 - Pellet

Outils liés à OWL

- Editeurs
 - Swoop, Protege, KAON, OntoEdit, Doe
- Raisonneurs
 - Pellet, Fact++, Racer (pro)
- Bases de triplets RDF
 - Jena, Sesame...

Plan

- Représentation de connaissances
- Ingénierie des connaissances
- Ontologies
- Graphes conceptuels
- Logiques de description
- Introduction au Web sémantique
- OWL
- **Conception d'ontologies**

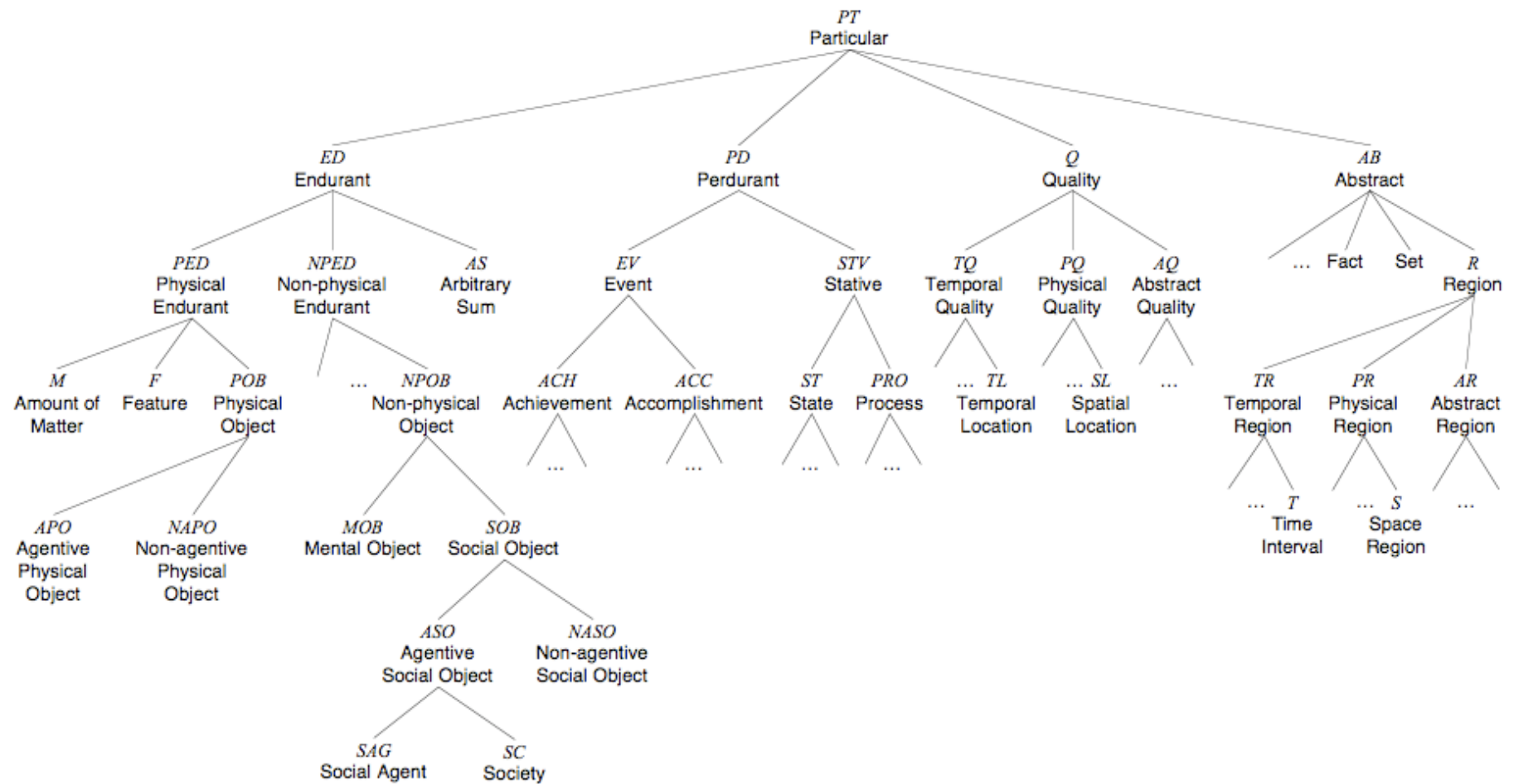
Construire des ontologies (1)

- A partir de corpus documentaire
 - outils de TAL: Traitement Automatique des Langues
 - Extraction de candidats-termes, candidats-relations, etc.
 - dans des domaines s'y prêtant bien, avec des méthodologies qui demandent encore à être affinées à la main
 - méthodologie du groupe TIA
 - construction de modèles terminologiques à partir de textes
 - qui servent de base à la construction d'ontologies formelles

Construire des ontologies (2)

- A la main
 - Travail avec experts : entretiens, observation
 - A partir de rien ou en réutilisant
 - ontologies dans des domaines proches
 - *ontologies de haut-niveau* suffisamment génériques
 - SUMO, DOLCE
 - Méthodologies de modélisation conceptuelle
 - e.g. pour spécialiser un concept, pour choisir entre propriété et concept, pour « nettoyer » une ontologie...
 - Outil de validation de la cohérence formelle
 - Bonnes pratiques
 - voir ISWC–Tutorial–Best–practice–2005.ppt (de <http://www.co-ode.org/resources/tutorials/iswc2004/>)

DOLCE : haut-niveau



Objectifs DOLCE

WONDERWEB Foundational Ontologies Library are to serve as:

- starting point for building new ontologies. One of the most important and critical questions when starting a new ontology is determining what things there are in the domain to be modeled. Adopting a high level view provides an enormous jump start in answering this question;
- a reference point for easy and rigorous comparisons among different ontological approaches;
- a common framework for analyzing, harmonizing and integrating existing ontologies and metadata standards (by manually mapping existing categories into the categories assumed by some module(s) in the library).

In addition, we intend the library to be:

- minimal – as opposed to other comprehensive ontology efforts, we intend the library to be as general as possible, including only the most reusable and widely applicable upper-level categories;
- rigorous – where possible, the ontologies in the libraries will be characterized by means of rich axiomatisations, and the formal consequences (theorems) of such characterizations will be explored in detail;
- extensively researched – each module in the library will be added only after careful evaluation by experts and consultation with canonical works. The basis for ontological choices will be documented and referenced.

Utiliser une ontologie

- En tant qu'inscription d'une modélisation conceptuelle, au sein d'une application, au sein d'un contexte d'usage
- Evolution
 - les usages évoluent
 - l'application change, l'ontologie doit évoluer
 - interconnection de systèmes
 - utiliser les ontologies d'autres systèmes
 - interopérabilité
 - syntaxique
 - sémantique

Visualiser et interagir avec des ontologies

The screenshot displays a software interface for ontology management, divided into two main panels: **SUBCLASS EXPLORER** and **CLASS EDITOR**.

SUBCLASS EXPLORER: This panel shows the project name "pizza.owl" and an "Asserted Hierarchy" tree. The hierarchy starts with "owl:Thing" and branches into "DomainConcept", "Country", "Ice Cream", "Pizza", "PizzaBase", and "PizzaTopping". Under "PizzaTopping", there are several sub-classes, including "VegetableTopping" and "PepperTopping". "GreenPepperTopping" is currently selected.

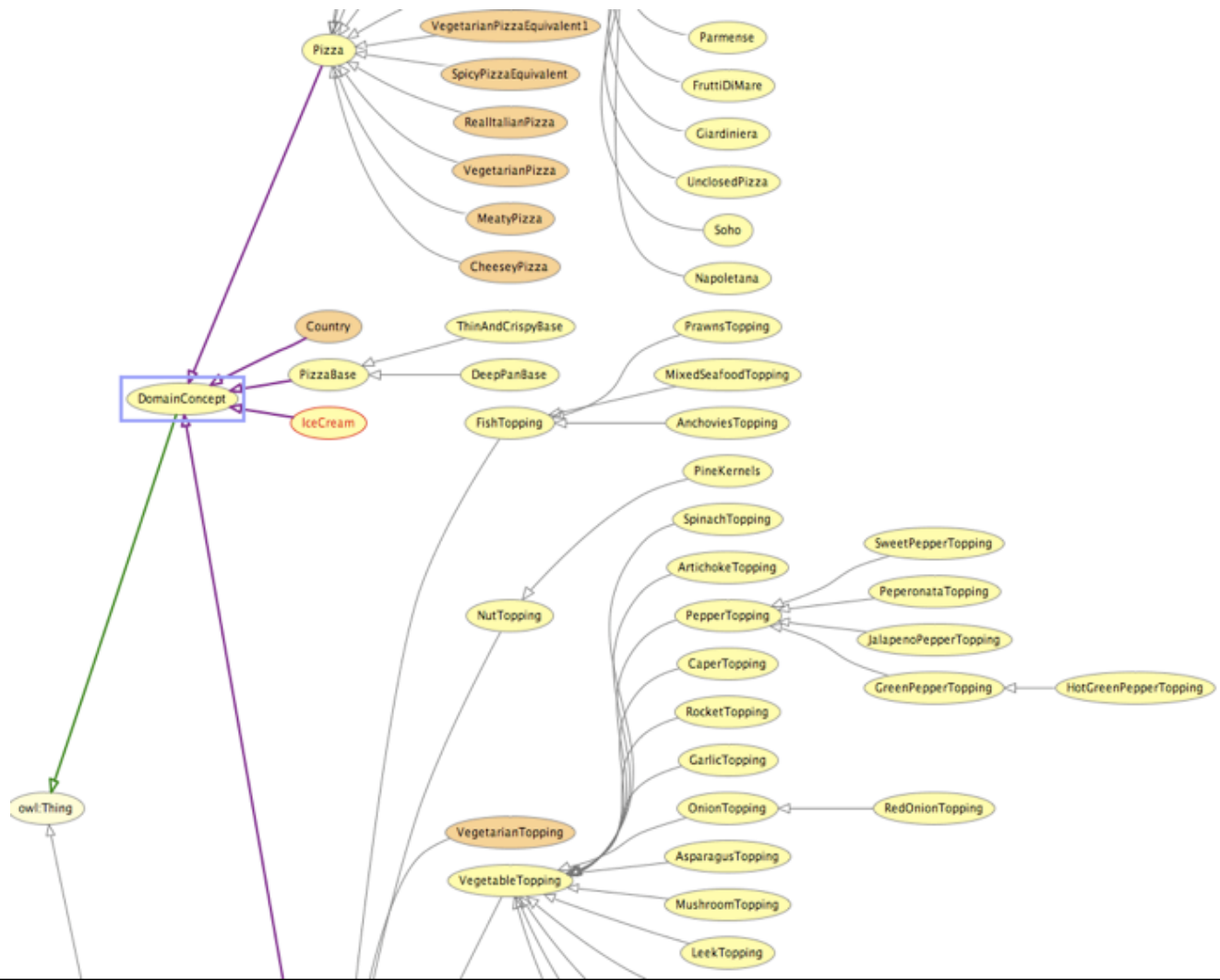
CLASS EDITOR: This panel is titled "For Class: GreenPepperTopping (instance of owl:Class)". It features a table with the following data:

Property	Value	Lang
rdfs:comment		
rdfs:label	CoberturaDePimentaoVerde	pt

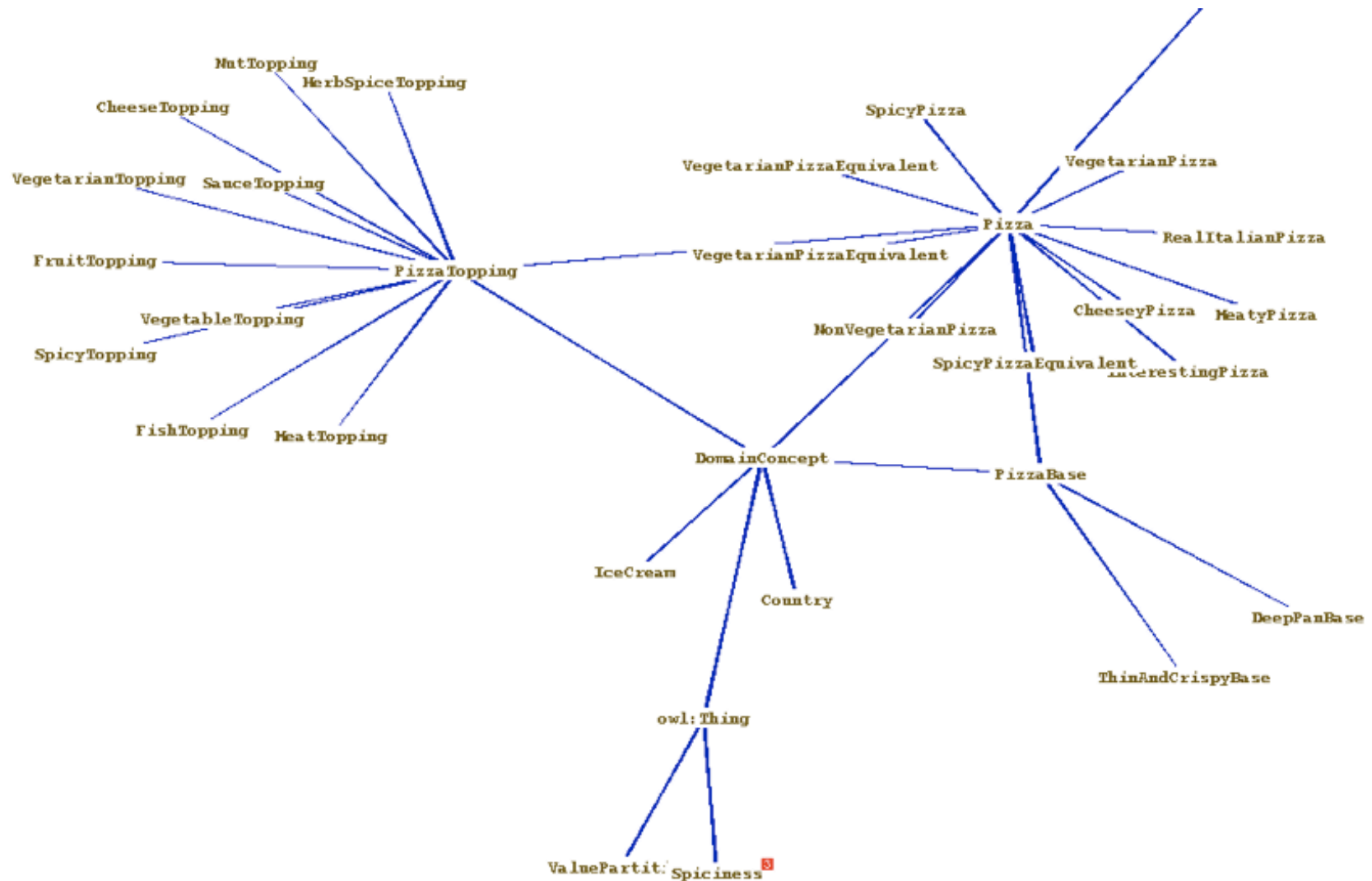
Below the table, there are sections for "Asserted Conditions" and "Disjoints". The "Asserted Conditions" section shows "PepperTopping" with a "NECESSARY & SUFFICIENT" condition. The "Disjoints" section lists "PeperonataTopping", "JalapenoPepperTopping", and "SweetPepperTopping".

At the bottom of the interface, there are radio buttons for "Logic View" (selected) and "Properties View".

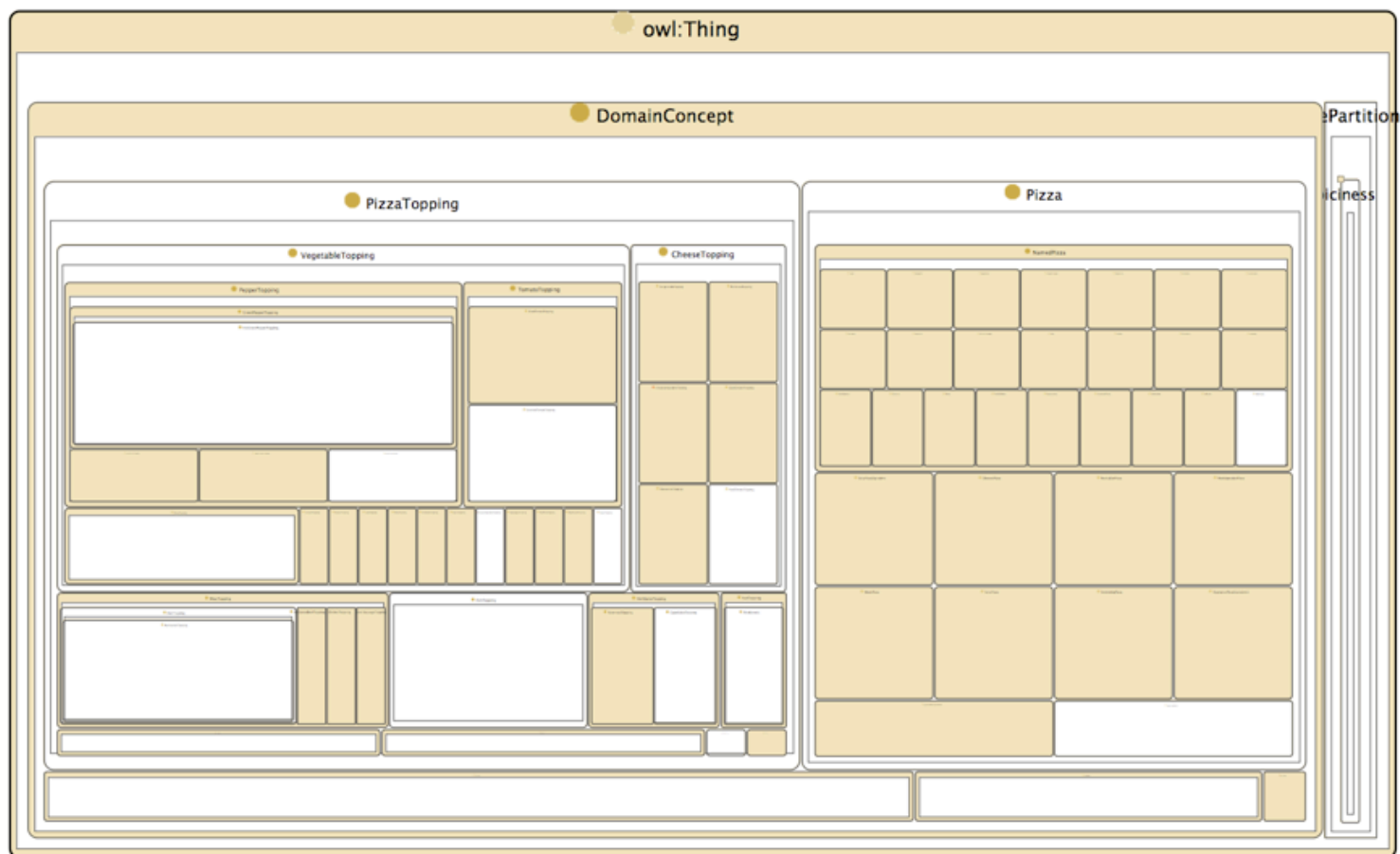
Visualiser et interagir avec des ontologies



Visualiser et interagir avec des ontologies



Visualiser et interagir avec des ontologies



Plan

- Représentation de connaissances
- Ingénierie des connaissances
- Ontologies
- Graphes conceptuels
- Logiques de description
- Introduction au Web sémantique
- OWL
- Conception d'ontologies
- **Bilan**

Bilan

- Des formalismes de représentation de connaissances
 - graphes conceptuels
 - logiques de description
- Le Web sémantique comme extension du Web offrant la possibilité d'exprimer des connaissances pour les machines (et les humains qui les utilisent)
 - RDF(S)
 - OWL
- Tendances
 - "semi-formalisation"
- Travaux pratiques
 - prise en main de Protege-OWL
 - réalisation d'une ontologie OWL

Remerciements - emprunts

- La partie sur LD / OWL a été réalisé en reprenant nombre de transparents de Pierre-Antoine Champin (Université Lyon 1)
- Quelques transparents sont repris de Philippe Laublet (Université Paris 5), Alain Mille (Université Lyon 1), Chantal Reynaud (Université Paris Sud)
- Site du Gracq

Annexe

- différentes sortes d'ontologies
(Chantal Reynaud)

Des descriptions variées

- Distinction selon **la structure** de la conceptualisation
 - Ontologies **linguistiques ou terminologiques** : lexiques de termes
 - Ontologies **d'information**: cf. spécification de la structure des enregistrements d'une BDR
 - Ontologies de **modélisation** des connaissances
- Distinction selon **le sujet** de la conceptualisation
 - Ontologies de **représentation** des connaissances- Ex : Frame Ontology
 - Ontologies **générales, de haut niveau (Top-Level), Meta-ontologies**
 - Ontologies de **domaine**
 - Ontologies d'**applications**

Ontologies linguistiques

Elles spécifient les termes qui sont utilisés pour représenter la connaissance d'un domaine.

Ex : **WordNet**, une ontologie linguistique qui couvre la plupart des mots anglais ordinaires

- WordNet est composée d'ensembles de synonymes (synsets), des termes regroupés en classes d'équivalence sémantiques, chaque terme appartenant à une catégorie lexicale donnée (nom, verbe, adverbe, adjectif).
- Chaque ensemble de synonymes représente un sens particulier d'un mot anglais. Un mot peut appartenir à plusieurs ensembles et plusieurs catégories lexicales.
- Les ensembles de synonymes sont associés par des relations sémantiques : hyperonymie- hyponymie (is-a), antonymie (relation entre ensembles de mots qui, par leur sens, s'opposent), etc.

Relations sémantiques représentées dans WordNet

Relation sémantique	Catégorie lexicale	Exemples
Synonymie	N, V, adj, adv	horse - knight / remember - reward happy - euphoric/ rapidly - speedily
Antonymie	adj, adv	wet - dry / powerfull - powerless
Généralisation Spécialisation	N	car - motor vehicle
Composant Composé	N	oxygen - air / car - air bag
Troponymie	V	march - walk
Opposition séquentielle	V	divorce - marry

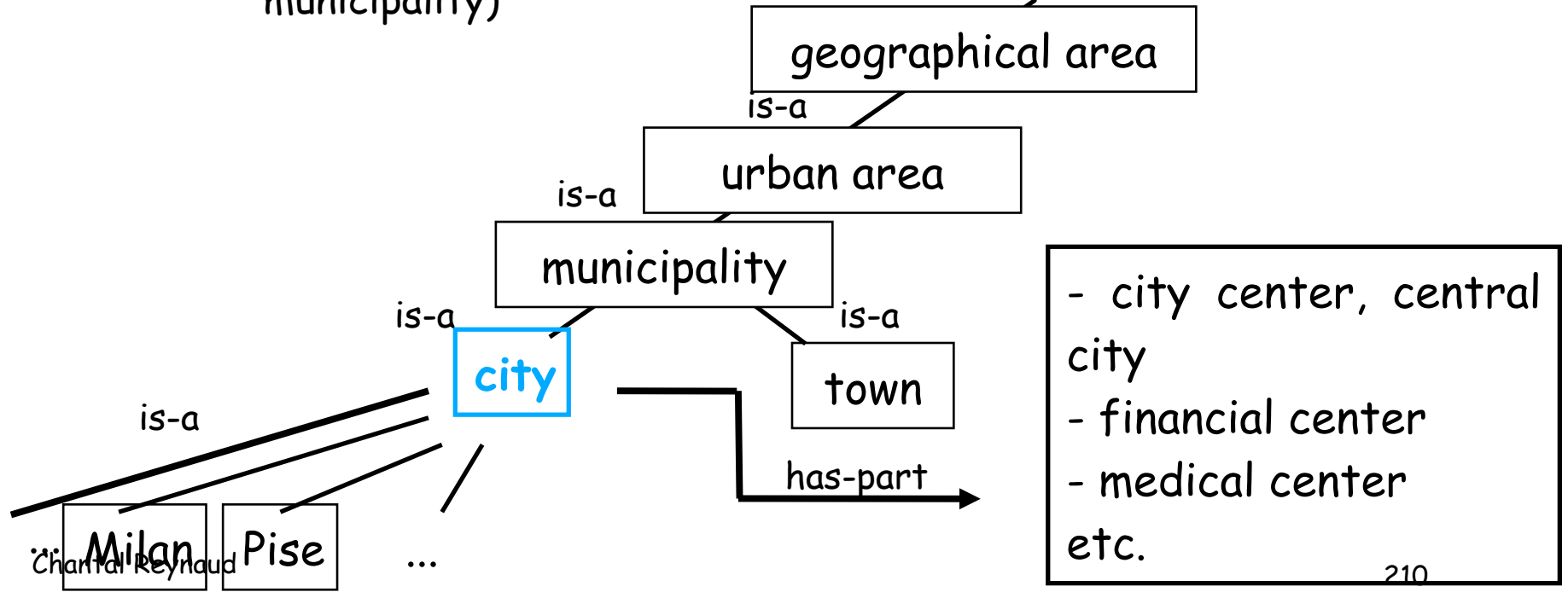
City

Ensemble de synonymes (Synset)

Définition

3
sens

- 1. city, metropolis, urban center -- (a large and densely populated urban area; may include several independant administrative districts; etc)
- 2. city -- (an incorporated administrative district established by state charter)
- 3. city, metropolis -- (people living in a large densely populated municipality)



Ontologies d'information

Elles spécifient la structure des enregistrements des bases de données.

Ex : le niveau 1 du modèle PEN & PAD, un cadre pour modéliser les enregistrements de données médicales de patients (Univ. de Manchester)

Ce que les médecins ont entendu, vu ou fait sans distinguer les symptômes, les signes cliniques, les tests de laboratoire ou les traitements.

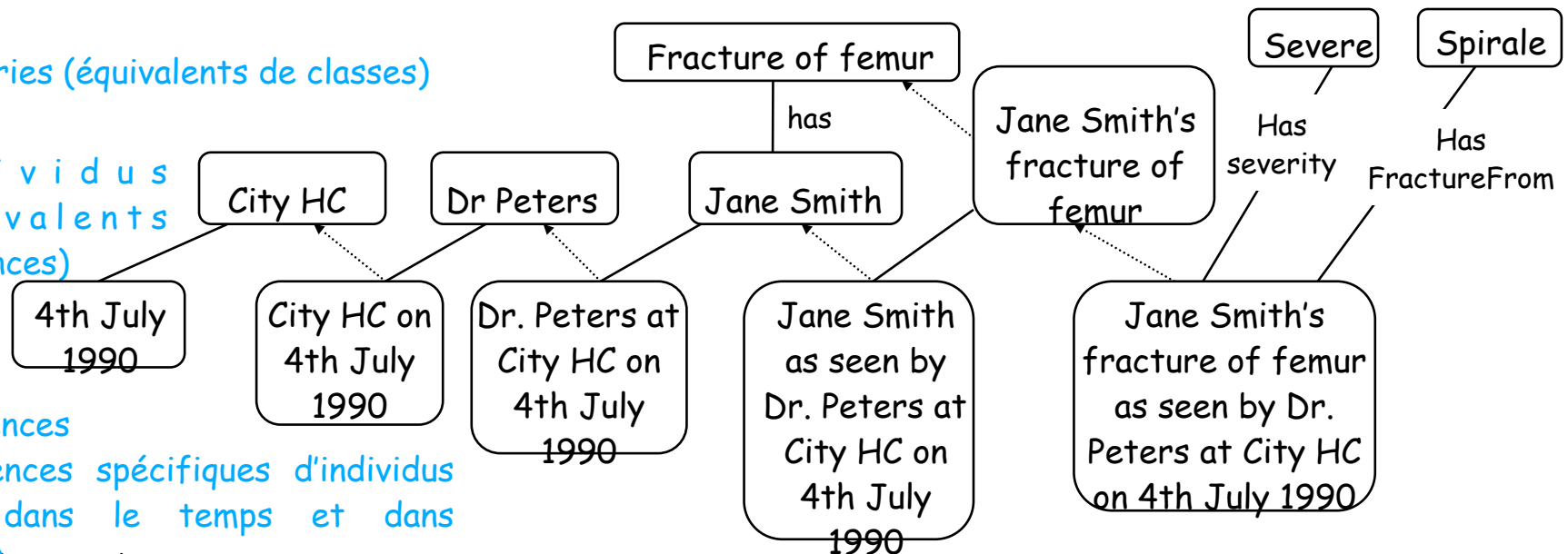
Ex : « When seen by Dr. Peters on 4 July 1990, Jane Smith had a severe spiral fracture of the femur »

Catégories (équivalents de classes)

Individus (équivalents d'instances)

Occurrences

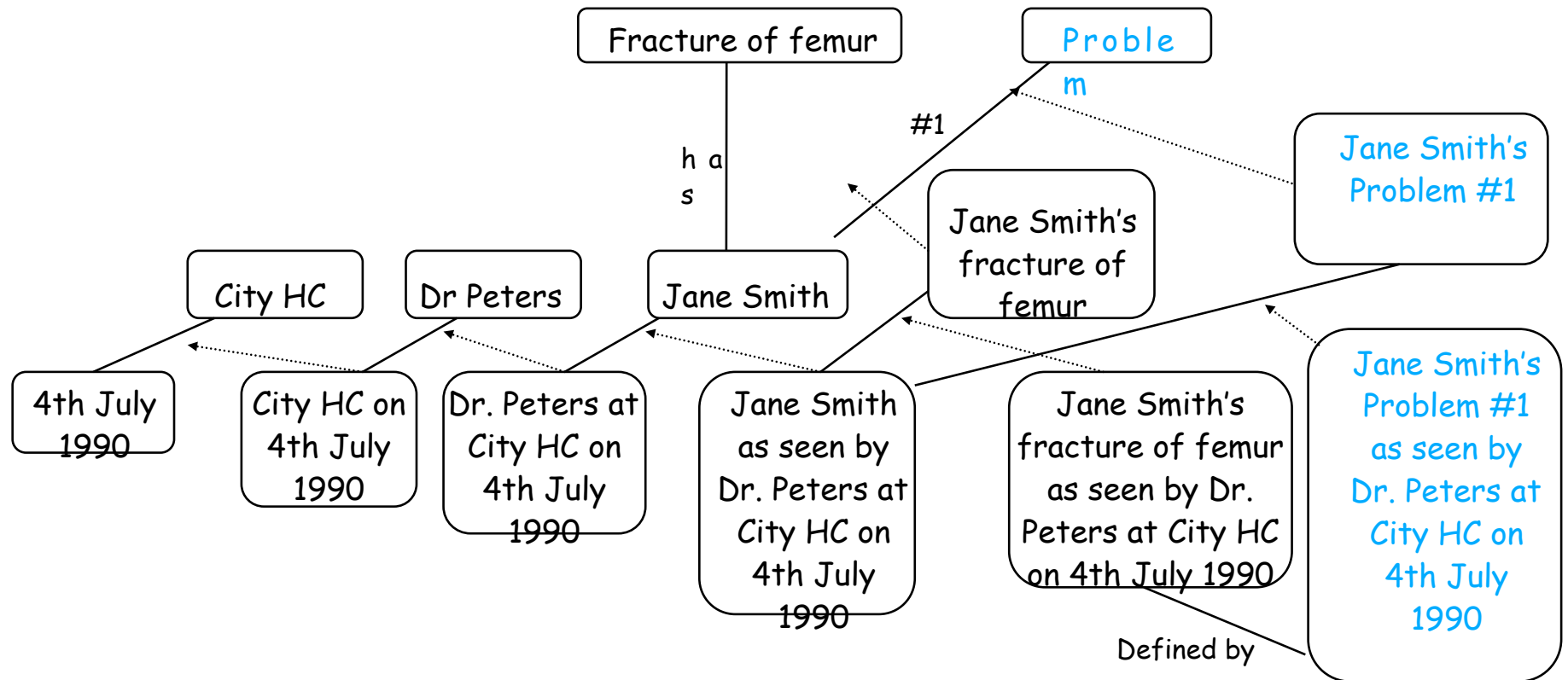
(occurrences spécifiques d'individus situés dans le temps et dans l'espace)



Ontologies de modélisation

Elles spécifient des conceptualisations de la connaissance en ayant souvent recours à des structures de représentation des connaissances plus riches

Ex : le niveau 2 du modèle PEN & PAD, une ontologie médicale.



Ontologies de représentation

Définition d'un ensemble de **primitives de représentation** : classe, relation, fonction.

Ex : **The Frame Ontology** (Gruber)

Définition des primitives de représentation utilisées dans les langages à base de frames ou dans les langages objets

Extrait du vocabulaire défini

Class relation (?relation)	Relation onto (?relation ?range-class)
Class function (?function)	Class n-ary-relation (?relation)
Class class (?class)	Class unary-relation (?relation)
Relation instance-of (?individual ?class)	Class binary-relation (?relation)
Function all-instances (?class) :-> ?set-of-instances	Class unary-function (?function)
Function one-of (@instances) :-> ?class	Relation single-valued (?binary-relation)
Relation subclass-of (?child-class ?parent-class)	Function inverse (?binary-relation) : -> ? relation
Relation superclass-of (?parent-class ? Child-class)	Function projection (?relation ?column) :-> ? class
Relation direct-instance-of (?individual ?class)	Function composition (?relation1 ? relation2) :-> ?binary-relation
Relation direct-subclass-of (?child-class ?parent-class)	Etc.
Function arity (?relation) : -> ?n	
Function exact-domain (?relation) :-> ?domain-relation	
Class exact-range (?relation) : -> ?class	

Ontologies générales

Définition du vocabulaire relatif aux choses, aux événements, au temps, à l'espace, à la causalité, au comportement, à la fonction, etc.

Ex : the *CYC knowledge base* (Lenat-Guha)

Une représentation formalisée d'un vaste ensemble de connaissances humaines fondamentales : des faits, des règles de bon sens, des heuristiques de raisonnement sur les événements et les objets de la vie de tous les jours.

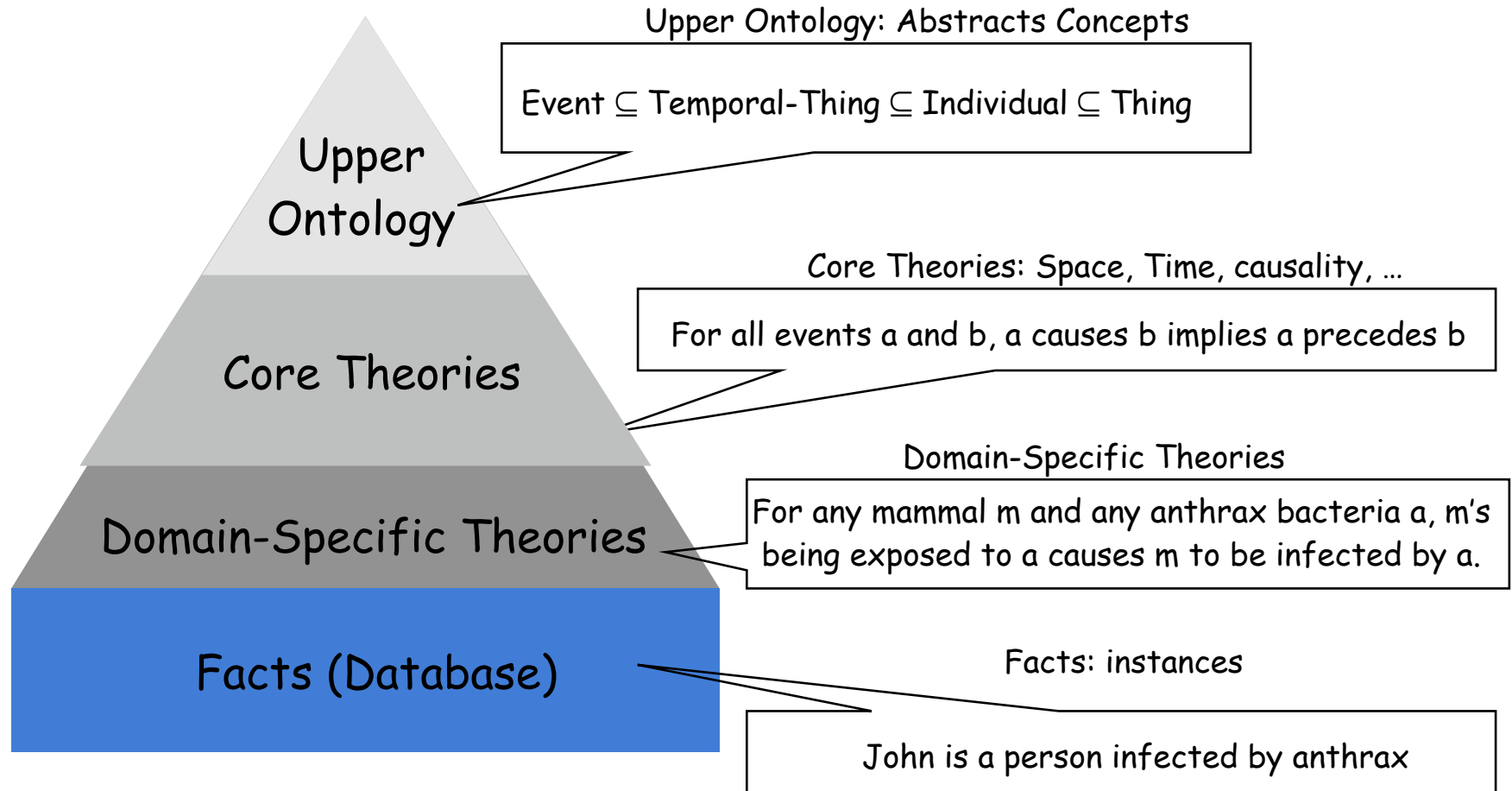
La base de connaissance *CYC* se compose d'un ensemble de termes (200 000) et d'assertions liant ces termes (plusieurs douzaines par terme) regroupés en micro-théories. Cette BC est sans cesse complétée par des termes non atomiques créés automatiquement via des fonctions ainsi que par des assertions déduites par le système d'inférence associé.

Termes

Quelques exemples : Thing, Intangible thing, Individual, Relations, Sets, Collections, Paths, Logic, Math, Space, Spatial paths, Borders geometry, Movement, Artifacts, etc.

Knowledge base layers

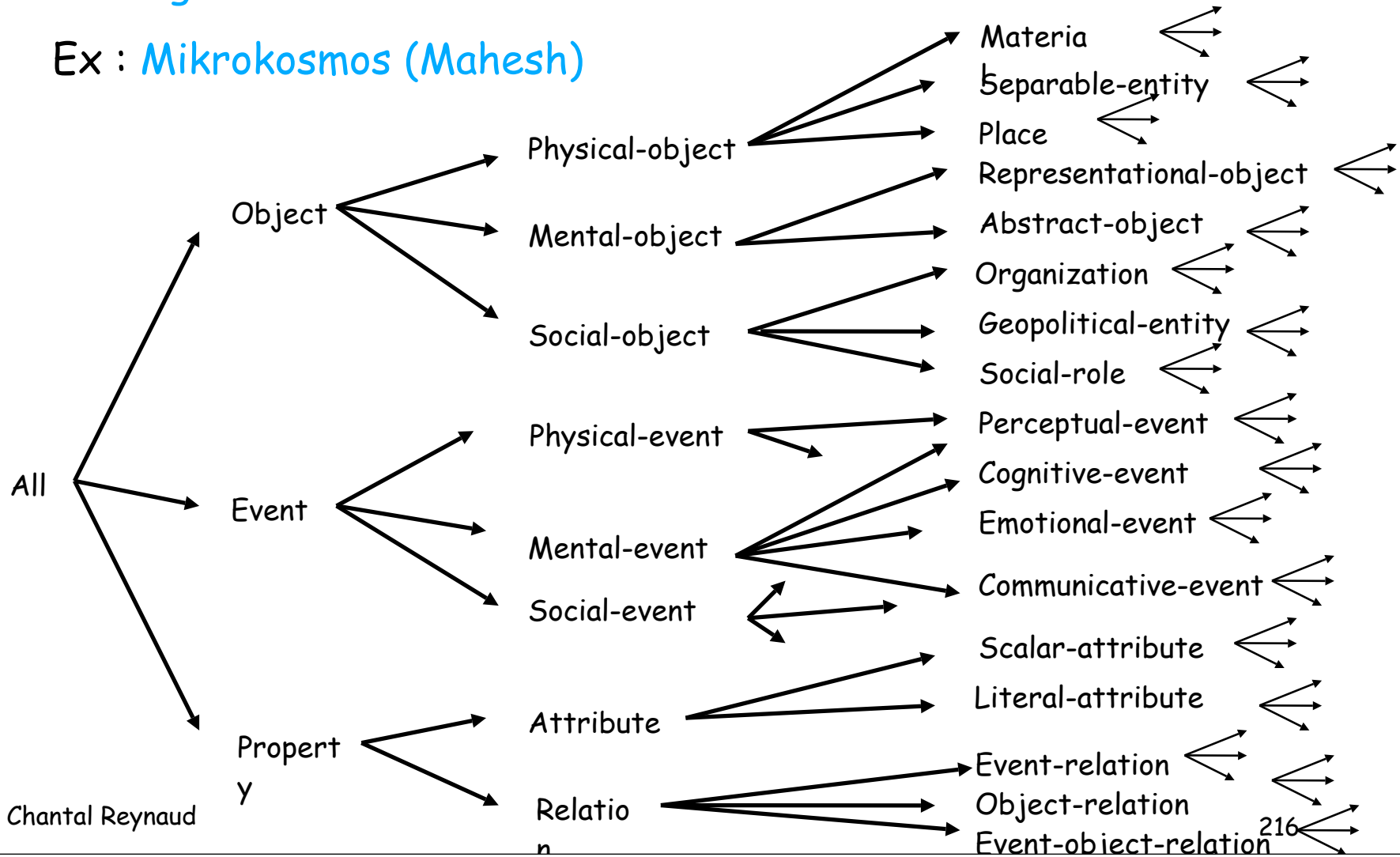
(The Cyc Knowledge Base)



Ontologies « Top-Level »

Définition de notions en relation avec tout concept de toute ontologie

Ex : Mikrokosmos (Mahesh)



Meta-Ontologies ou Ontologies génériques

Définitions de notions utilisables quel que soit le domaine d'application

Ex : Extrait de *The Mereology ontology* (Borst)

- 1 define-theory mereology
- 2 Define-class m-individual (x)
m-individual (x) \leftrightarrow equal (x,x)
- 3 Define-relation proper-part-of (x,y)
proper-part-of (x,y) \rightarrow not proper-part-of (y,x)
proper-part-of (x,y) and proper-part-of (u,z) \rightarrow proper-part-of (x,z)
- 4 Define-relation direct-part-of (x,y)
direct-part-of (x,y) \leftrightarrow proper-part-of (x,y) and
not exists z: proper-part-of (z,y) and proper-part-of (x,z)
- 5 Define-relation disjoint (x,y)
disjoint (x,y) \leftrightarrow not equal (x,y) or
exists z: proper-part-of (z,x) and proper-part-of (z,y)

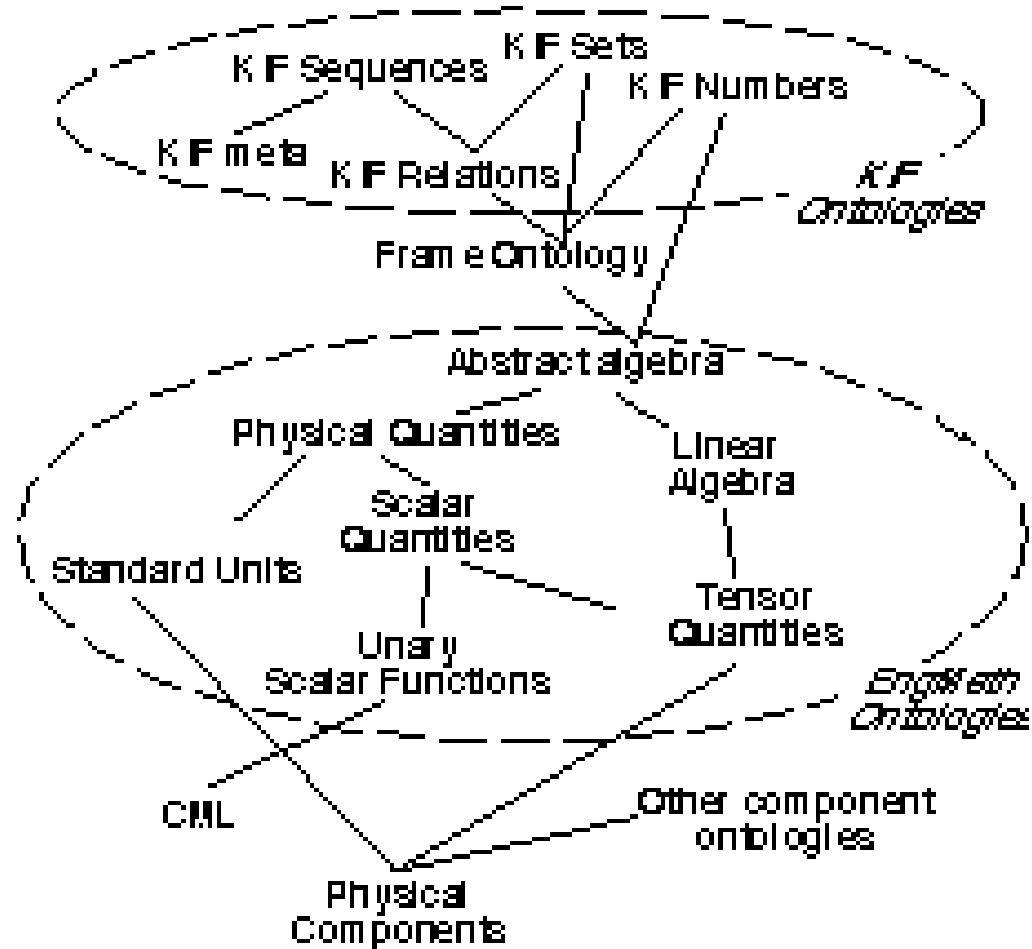
Ontologies de domaine

Réutilisable à l'intérieur d'un domaine donné.

Exemple : [EngMath Ontology](#) (Gruber - Olsen), une ontologie pour la modélisation mathématique en ingénierie

Elle inclut les bases conceptuelles relatives aux scalaires, aux vecteurs, aux dimensions physiques, aux unités de mesure, aux fonctions portant sur des quantités, aux quantités sans dimension.

Elle a été conçue pour être partagée. C'est une base pour la construction d'autres ontologies en ingénierie.

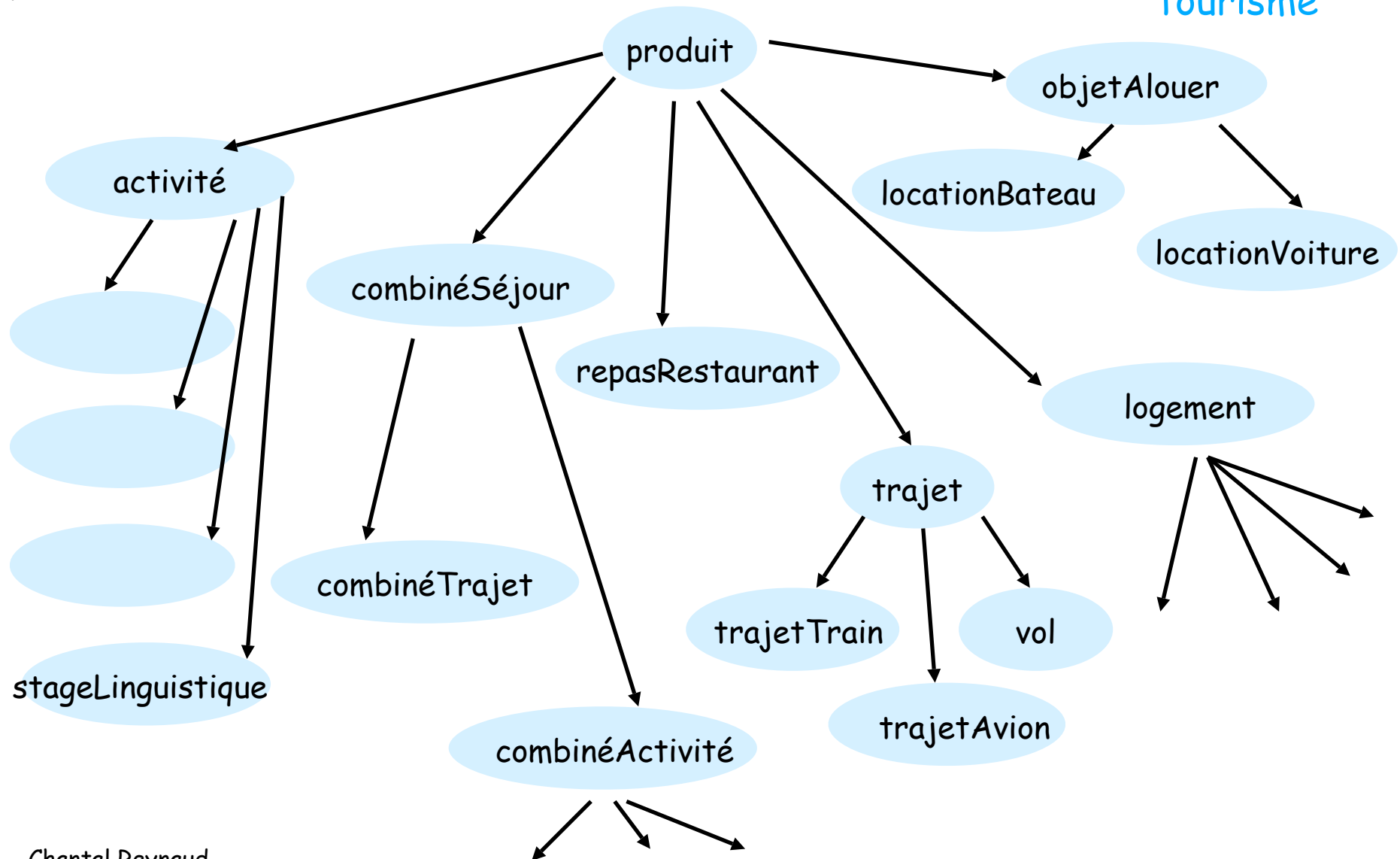


Elles contiennent la connaissance spécifique à une application particulière

Ontologies d'application

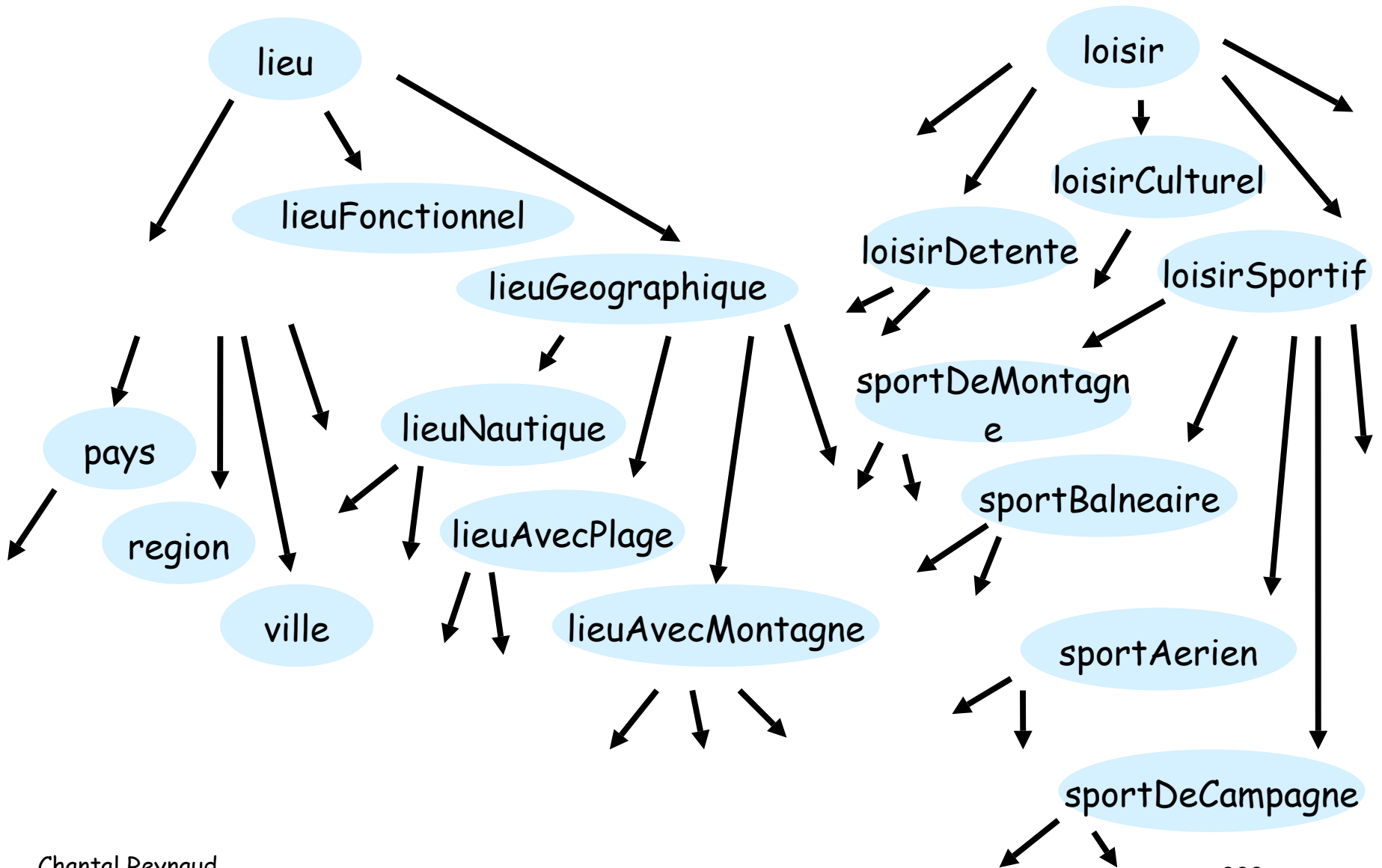
PICSEL1 : Une ontologie principale

Domaine des produits de tourisme



Ontologies d'application

PICSEL1 : des hiérarchies secondaires



Ontologie du médiateur PICSEL

Définitions de concepts

Descriptions terminologiques

produit := (=1 prixAssocié) and (= 1 dateDébutAssocié) and
(≥0 serviceProduitAssocié) and (ALL serviceProduitAssocié.service) and
(≥0 prestationProduitAssocié) and (ALL prestationProduitAssocié.prestation).

activité := produit and
(=1 duréeActivitéAssocié) and
(= 1 natureActivitéAssocié) and (ALL natureActivitéAssocié.loisir) and
(=1 lieuActivitéAssocié) and (ALL lieuActivitéAssocié.lieu).

activitéSportive := activité and
(=1 natureActivitéAssocié) and (ALL natureActivitéAssocié.loisirSportif).

loisirSportif \subseteq loisir.

Inclusion de concepts

Ontologie du médiateur PICSEL

Datalog

- Pour représenter des **connaissances factuelles** : $p(a_1, \dots, a_n)$

Ex : france("paris")

- Pour représenter des **connaissances déductibles** :

$$p_1(x_1, \dots, x_n) \wedge \dots \wedge p_k(y_1, \dots, y_m) \Rightarrow q(z_1, \dots, z_j)$$

Ex : séjour(S,L,R) \wedge nbreNuitsCombineSejourAssocie(S, nombreDeJours)
 \wedge inf(nombreDeJours, quatre) \Rightarrow courtSejour(S)

- Cas particulier : la représentation de **contraintes d'intégrité**

Ex : numDepartementLieuResidence(R, N) \wedge NOT nbre(N) $\Rightarrow \perp$

Références variées

- http://www.philippe-fournier-viger.com/description_logics/introduction_logiques_de_description.html