

Introduction à l'Intelligence Artificielle (L2 Portail Sciences et Technologies)

Andrea G. B. Tettamanzi
Laboratoire I3S – Équipe SPARKS
`andrea.tettamanzi@univ-cotedazur.fr`



univ-cotedazur.fr

Séance 5

Représentation des connaissances

Plan pour cette séance

- Connaissances et ontologies
- Formalismes de représentation
 - Des réseaux sémantiques aux graphes conceptuels
 - Logiques de description
- Le Web sémantique

Représenter les connaissances

- Pour pouvoir raisonner, un agent intelligent doit représenter (explicitement ou implicitement) ses connaissances
- Comprendre un texte signifie construire une représentation (explicite ou implicite) de son sens
- Le « sens » se base sur les autres connaissances de l'agent
- Pour qu'il y ait de la communication, il faut que les interlocuteurs possèdent, au départ, des connaissances en commun

Ontologie

- En philosophie, c'est la science de l'être, l'étude des premiers principes.
- Aristote : « Il y a une science qui étudie l'être en tant qu'être, et les attributs qui lui appartiennent essentiellement. Elle ne se confond avec aucune de ces sciences particulières, car aucune de ces autres sciences dites particulières ne considère en général l'Être en tant qu'être, mais découpant une certaine partie de l'Être c'est seulement de cette partie qu'elles étudient l'attribut »
- *Une* ontologie = théorie qui tente d'expliquer les concepts qui existent dans le monde et comment ces concepts s'imbriquent et s'organisent pour donner du sens

Ontologies

- Ontologie = structure sémantique intensionnelle encodant les règles implicites contraignant la structure d'un morceau de réalité
- Les logiques de description (LD) fournissent des langages pour exprimer ces ontologies
- Ontologie ou base de connaissances = Tbox (+ Rbox) + Abox
- Une ontologie consiste en
 - Un ensemble de noms de concepts C
 - Un ensemble de noms de relations R
 - Une relation de subsomption \sqsubseteq
 - Un ensemble d'axiomes (définition de concepts/rerelations, etc.)
 - Des faits : $C(a)$, $R(a, b)$

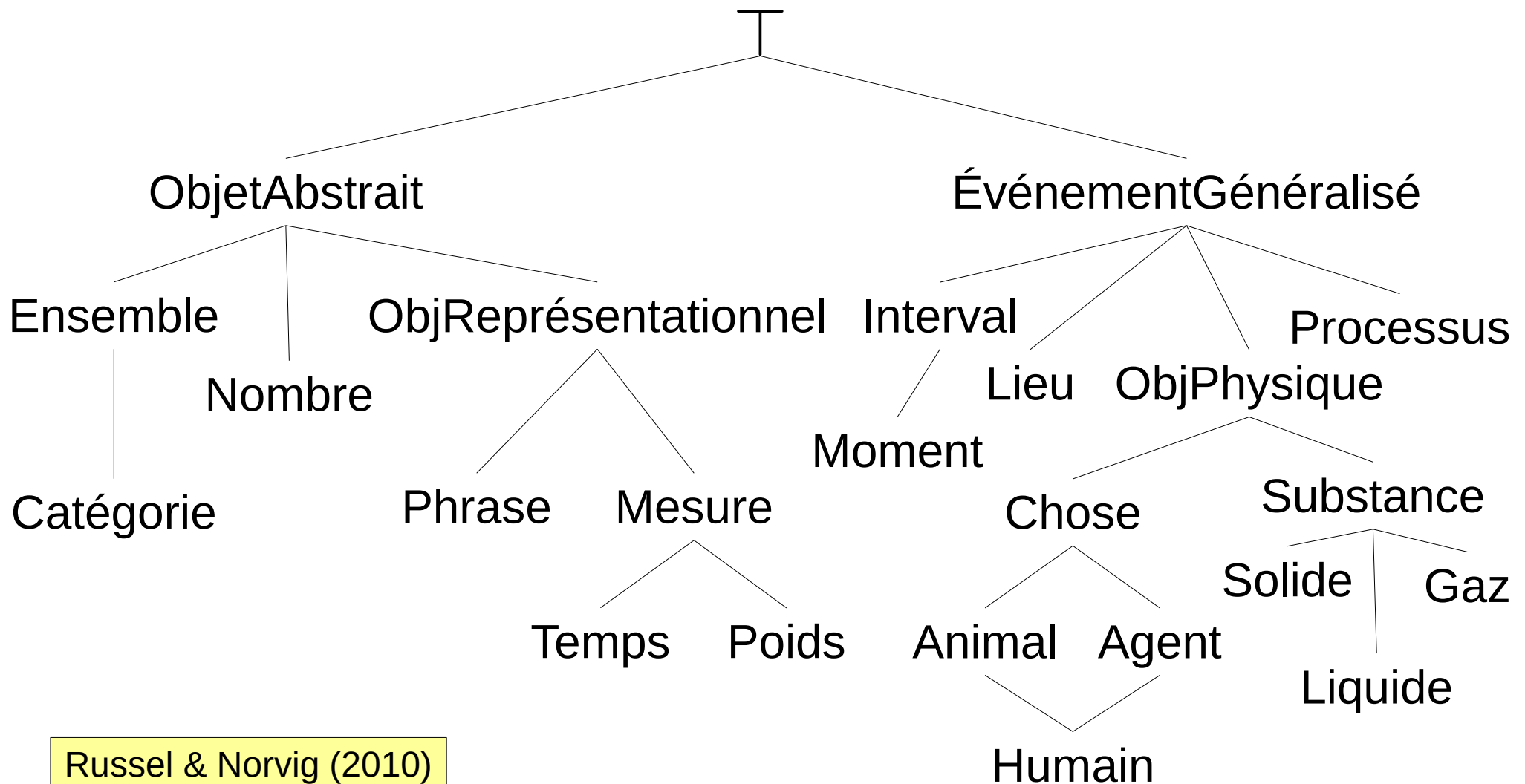
Ontologies

- Définition de Gruber :
« Une ontologie est la spécification d'une conceptualisation. [...] Une conceptualisation est une vue abstraite et simplifiée du monde que l'on veut représenter »
- Autre définition, plus opérationnelle :
« Une ontologie est un *réseau sémantique* qui regroupe un ensemble de concepts décrivant complètement un domaine. Ces concepts sont liés les uns aux autres par des relations taxinomiques (hiérarchisation des concepts) d'une part, et sémantiques d'autre part »

Ontologies

- On peut distinguer trois types d'ontologies :
 - **Globales** (Upper Ontology) : au plus haut niveau d'abstraction et de généralité, issues d'un développement systématique, rigoureux et axiomatique de la logique de toutes les formes et modes d'existence. Construites pour être partagées et transférées d'un contexte à l'autre
 - **De domaine** : limitées à la représentation de concepts dans des domaines donnés (géographie, médecine, écologie, etc.), spécialisant les concepts de l'ontologie globale
 - **Applicatives** : dédiées à un champ d'application précis à l'intérieur d'un domaine, offrent le plus fin niveau de spécificité et décrivent le rôle particulier des entités de l'ontologie de domaine dans ce champ.

Exemple d'ontologie globale



Russel & Norvig (2010)

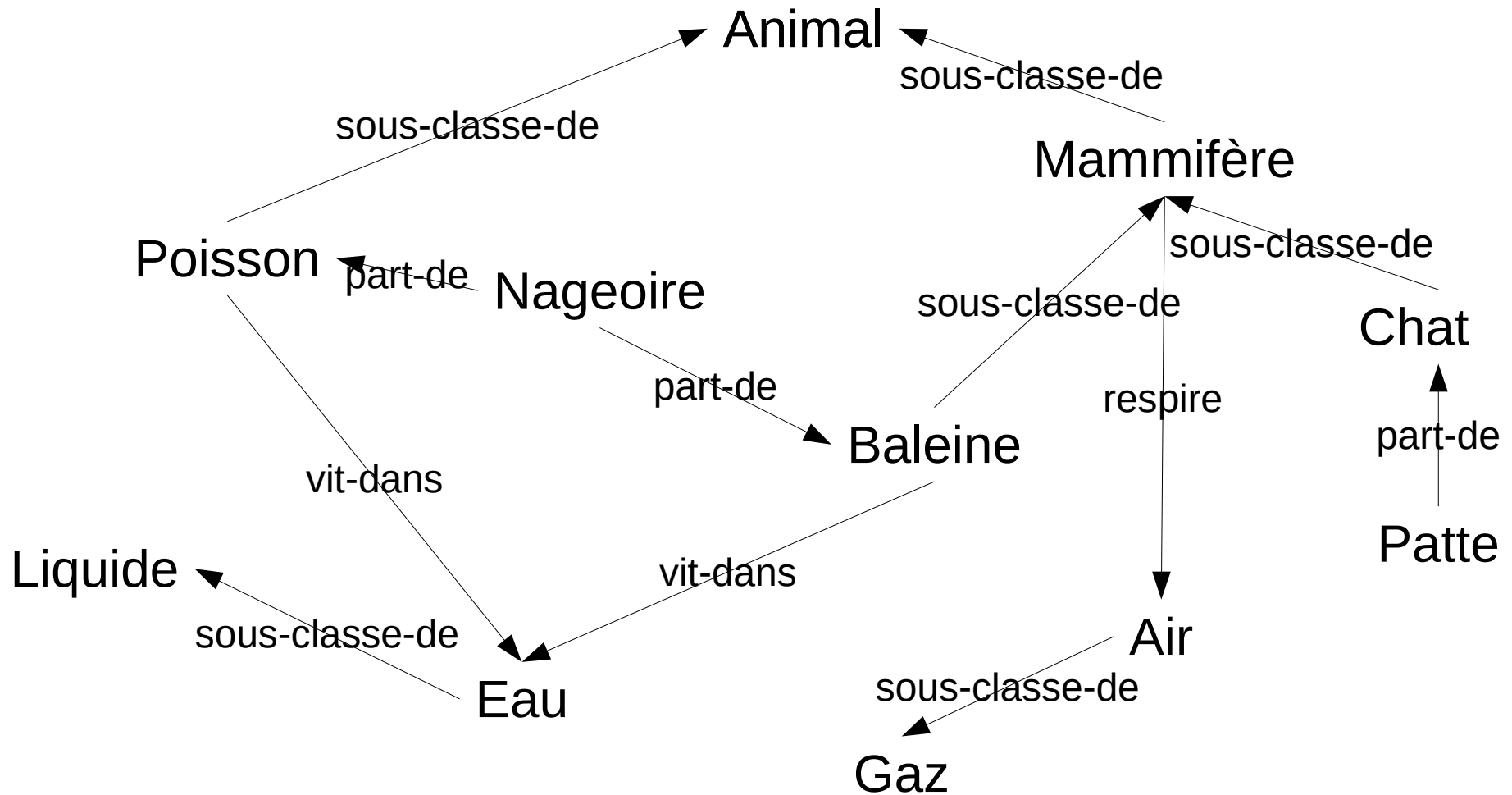
Formalismes de représentation

- On cherche un langage adapté à décrire les connaissances d'un agent intelligent
 - Les langues humaines sont sans doute un outil adapté (l'humanité les utilise déjà depuis des millénaires pour cela!)
 - Cependant, on vient de voir dans la séance précédente que leur traitement par ordinateur est tout sauf que facile !
- On doit alors se tourner vers la logique :
 - Système avec syntaxe et sémantiques formelles
 - Règles d'inférence pour le raisonnement

Réseaux sémantiques

- Un problème de la logique est qu'elle est peu lisible : variables et quantificateurs
- Dans les années 1960, les réseaux sémantiques ont été proposés comme alternative
- Un réseau sémantique est un graphe marqué
 - Nœuds : concepts et objets individuels
 - Arcs : relations entre concepts et/ou objets

Exemple de réseau sémantique



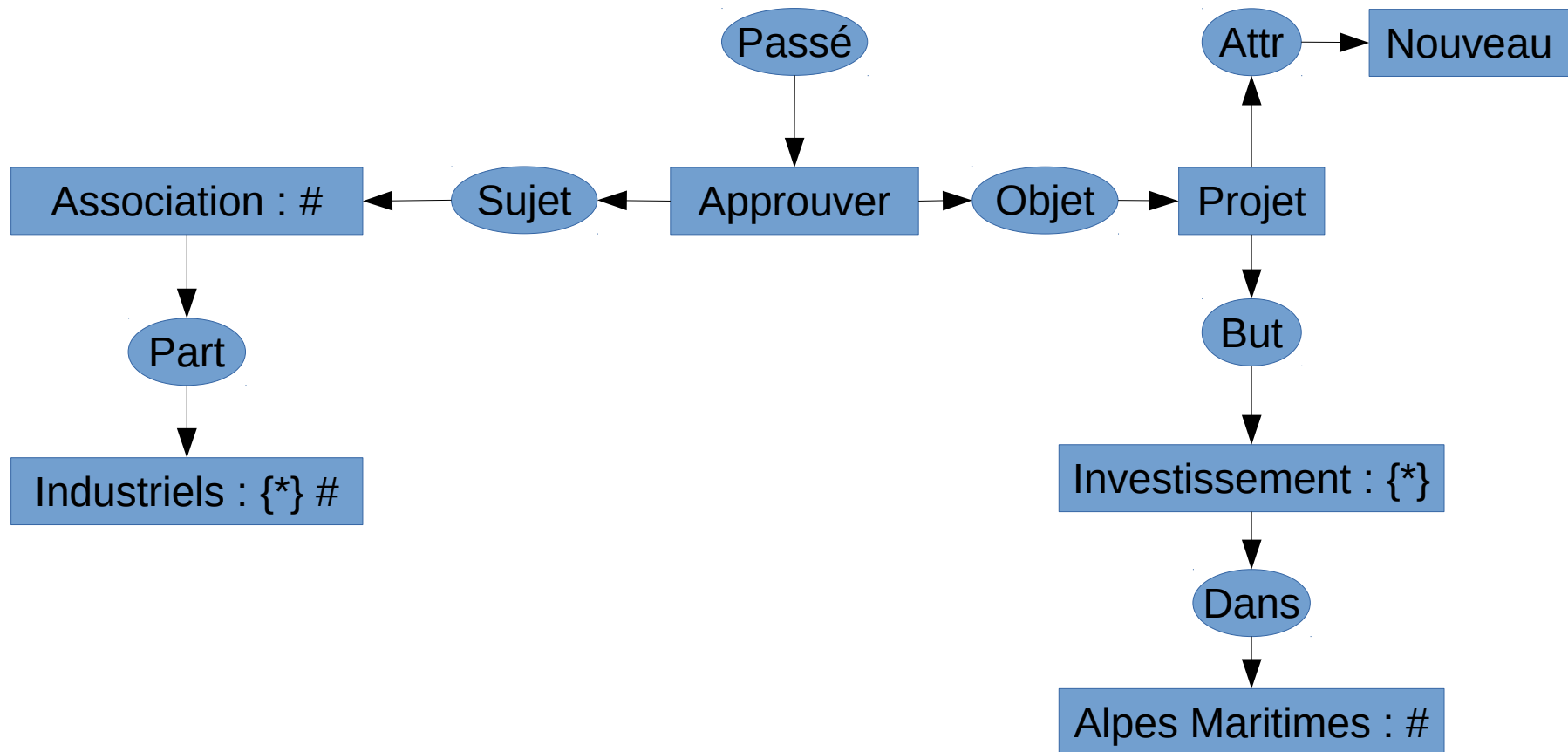
Graphes conceptuels

- Une extension de la notion de réseau sémantique introduite par John Sowa (1984)
- Combinaison avec les graphes existentiels du logicien Charles Sanders Peirce
- Concepts
- Relations conceptuelles

<champ type> : <champ référent>

<nom de la relation>

Exemple de graphe conceptuel



Modalités

- Des affirmations absolues ne suffisent pas à décrire un monde complexe
- Nous devons prendre en compte des attitudes propositionnelles telles que
 - Croire / penser / supposer
 - Savoir
 - Vouloir / désirer / souhaiter
 - Avoir l'intention de
 - Informer

Logiques Modales

- Idée : représenter ces modalités au niveau syntaxique
- Symbole de nécessité (modalité) $\Box P$
- Distinction nette entre $\neg\Box P$ et $\Box\neg P$
- Exemple : logique doxastique (von Wright, Hintikka, Fagin *et al.*) pour décrire les croyances
- Axiomes:
 - K $\Box(P \Rightarrow Q) \Rightarrow (\Box P \Rightarrow \Box Q)$
 - D $\Box P \Rightarrow \neg\Box\neg P$
 - T $\Box P \Rightarrow P$
 - 4 $\Box P \Rightarrow \Box\Box P$
 - 5 $\neg\Box P \Rightarrow \Box\neg\Box P$

Logiques non-monotones

- La logique classique est monotone : si j'ajoute un fait ou un axiome à un ensemble de faits ou d'axiomes, l'ensemble des conclusions que j'en peut tirer ne peut pas diminuer
- À la limite, si j'ajoute un fait qui contredit les connaissances précédentes, j'obtiens une base incohérente, d'où n'importe quelle conséquence peut être tirée
- En IA, on a besoin de logiques dans lesquelles la base de faits inférés peut ne pas croître et même parfois décroître quand on ajoute des nouvelles connaissances
- Toutes ou certaines connaissances dans la base sont provisoire (on parle alors de croyances) et peuvent être retirées
- (Être capable de) changer d'idée est une marque d'intelligence !

Révision de croyances

- Supposons que nous disposions d'une base de connaissances contenant:
 - A : L'or ne peut être taché que par l'aqua regia
 - B : L'acide contenu dans la bouteille est de l'acide sulfurique
 - C : L'acide sulfurique n'est pas aqua regia
 - D : Mon alliance est en or
- Le fait suivant est une conséquence logique de A–D:
 - E: Mon alliance ne sera pas tachée par l'acide contenu dans la bouteille
- Supposons maintenant que, en fait, l'alliance soit effectivement tachée par l'acide : on devrait ajouter $\neg E$ à nos connaissances
- Pourtant, la base deviendrait incohérente : nous devons réviser
- Au lieu de renoncer à toutes nos croyances, nous choisirons lesquelles garder

Logiques par défaut

- Nous sommes capables de raisonner avec des données et des connaissances incomplètes
- La logique par défaut formalise le raisonnement avec des hypothèses « par défaut »
- Exemple : « les oiseaux volent »
 - À savoir : [sauf avis contraire, typiquement] les oiseaux volent
 - Les manchots sont des oiseaux qui ne volent pas
- Formalisation :
 - <Prérequis> : <Justifications> / <Conclusion>
 - Si <Prérequis> est vrai et qu'aucune des <Justifications> n'est contredite par mes connaissances actuelles, je peux en conclure (provisoirement) <Conclusion>

Logiques de description

- Les réseaux sémantiques n'avaient pas de sémantique formelle
- Les graphes conceptuels ont la même puissance que la logique des prédicats du premier ordre ou même plus
 - La conséquence logique est indécidable
- Les logiques de description naissent du besoin de surmonter ces deux problèmes
- Les logiques de description forment une famille de langages de représentation de connaissances qui peuvent être utilisés pour représenter la connaissance terminologique d'un domaine d'application d'une façon structurée et formelle
- Chaque langage est caractérisé par une puissance expressive et une complexité des tâches de raisonnement
- Fragments décidables de la logique du premier ordre

Base de connaissances

- Base de connaissances = TBox + ABox
- Base d'assertions (ABox)
 - Connaissances contingentes
 - Vraies à un instant donné ou dans une situation donnée (qui restent implicites)
- Base terminologique (TBox)
 - Connaissances non-contingentes, ontologie
 - Axiomes \approx contraintes d'intégrité
 - Vraies toujours et dans toute situation

ABox

- Prédicats unaires
 - Expriment des propriétés d'individus
Femme(Anne)
Homme(Charles)
- Prédicats binaires
 - Expriment des relations entre individus
mèreDe(Anne, Charles)
pèreDe(Robert, Charles)

TBox

- Axiomatise les propriétés et les relations
 - terminologie LD : concepts et rôles
 - principalement : inclusion et équivalence de concepts

$\text{Femme} \equiv \text{Personne} \sqcap \text{Féminine}$

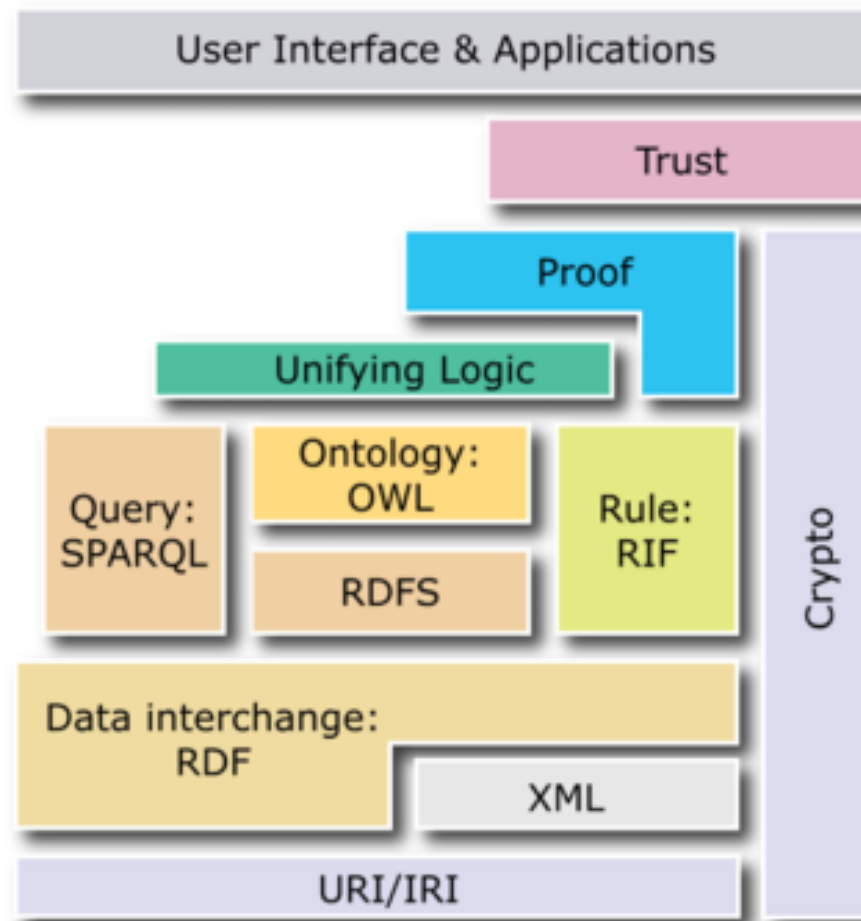
$\exists \text{mèreDe.}(\exists \text{mèreDe.}\top) \sqsubseteq \text{GrandMère}$

$\text{Mammifère} \sqsubseteq \neg \text{Poisson}$

Web sémantique

- Extension du Web standardisée par le World Wide Web Consortium (W3C)
- Ces standards encouragent l'utilisation de formats de données et de protocoles d'échange normés sur le Web, en s'appuyant sur le modèle Resource Description Framework (RDF)
- Le but est de construire une toile de données et connaissances accessibles à des agents intelligents
- Les formalismes de représentation des connaissances, les ontologies, etc., y sont mis en œuvre

La pile des technologies

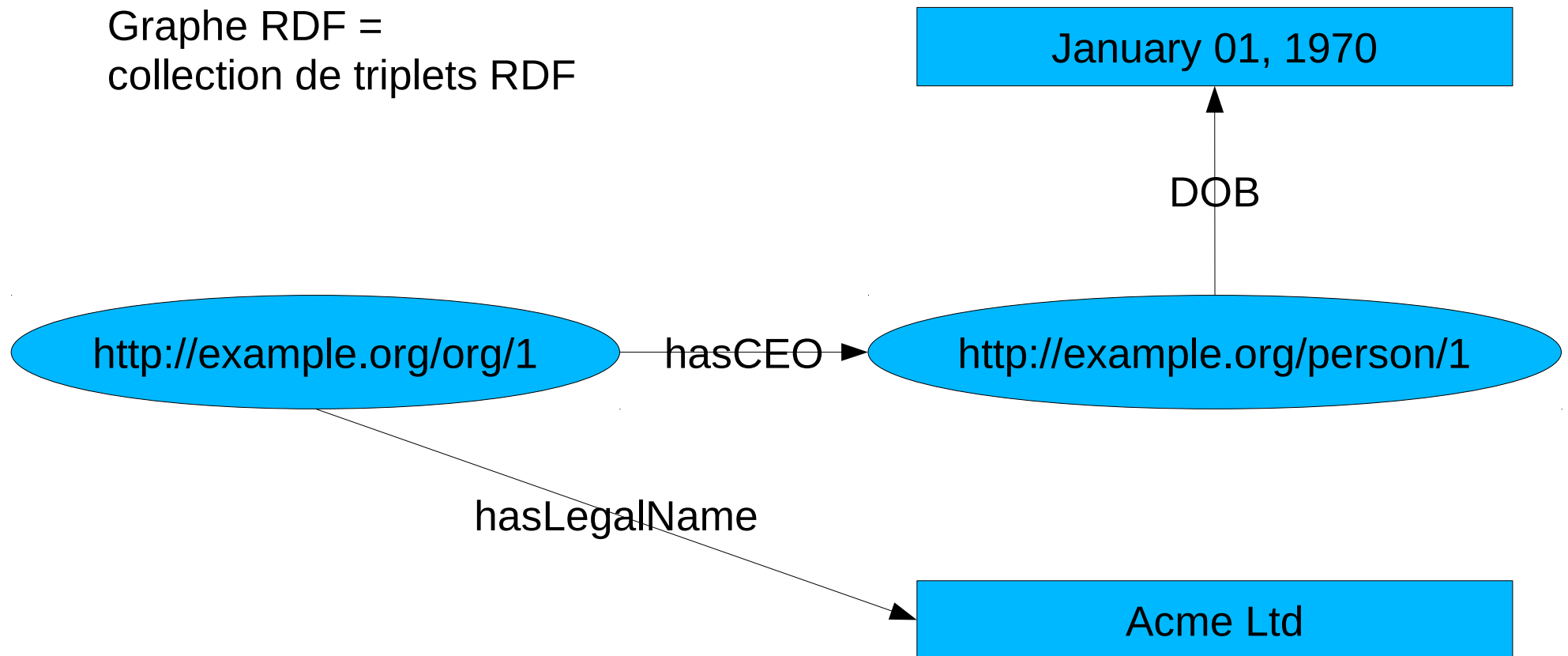


RDF

- Le modèle de données du Web sémantique
 - Publié comme recommandation du W3C en 1999
 - Initialement un modèle pour les métadonnées
 - Utilisé pour représenter tout type de données et connaissances
- Resource Description Framework
 - Ressource : tout ce qui peut avoir un URI, un nœud du graphe
 - Description : attributs et relations des ressources
 - Framework : modèle, langage, syntaxe et sémantique
- Basé sur des triplets de la forme (sujet prédicat objet)
 - Sujet et objet sont des ressources (= nœuds); objet peut être un littéral
 - Prédicat est une étiquette d'arc

Modèle de données RDF

Graphe RDF =
collection de triplets RDF



Syntaxe RDF

- RDF/XML

```
<rdf:RDF xmlns:voc="http://example.org/vocabulary">  
  <voc:RegisteredOrganization rdf:about="http://example.org/org/1">  
    <voc:hasLegalName> "Acme Ltd" </voc:hasLegalName>  
    <voc:hasCEO rdf:resource="http://example.org/person/1"/>  
  </voc:RegisteredOrganization>  
  <voc:Person rdf:about="http://example.org/person/1">  
    <voc:DOB>January 1, 1970</voc:DOB>  
  </voc:Person>  
</rdf:RDF>
```

- Turtle

```
@prefix voc: <http://example.org/vocabulary/>  
<http://example.org/org/1> rdf:type voc:RegisteredOrganization ;  
  voc:hasLegalName "Acme Ltd" ;  
  voc:hasCEO <http://example.org/person/1> .  
<http://example.org/person/1> rdf:type voc:Person ;  
  voc:DOB "January 1, 1970" .
```

SPARQL

- SPARQL = **SPARQL Protocol And RDF Query Language**
- SPARQL è le langage d'interrogation de RDF
 - Basé sur le modèle de données RDF (triplets / graphe)
 - Idée principale : trouver correspondances d'un patron
 - Déclaratif : il décrit des sous-graphes du graphe RDF interrogé
 - Patron de graphe (= graphe RDF avec variables)
- SPARQL est aussi un protocole
 - Transmission de requêtes SPARQL et des résultats
 - SPARQL Endpoint: service Web qui réalise ce protocole
- Standard du W3C depuis Janvier 2008

Types de requêtes SPARQL

- `SELECT ?x ?y ... WHERE patron-de-graphe(?x, ?y, ...)`
 - Renvoie une table des x, y, \dots qui correspondent à la description du graphe contenue dans le patron de graphe.
- `CONSTRUCT {?s ?p ?o} WHERE patron-de-graphe(?s, ?p, ?o)`
 - Trouve tout x, y, \dots satisfaisant les conditions données et les remplace dans les schémas de triplets pour créer, avec eux, un nouveau graphe RDF
- `DESCRIBE ?x WHERE patron-de-graphe(?x)`
 - Trouve toutes les déclarations qui décrivent les ressources qui correspondent au patron de graphe donné
- `ASK WHERE patron-de-graphe(?x, ?y, ...)`
 - Vérifie s'il existe x, y, \dots correspondant à la description

SPARQL: Exemple

- SELECT ?companyName
WHERE {
 ?company voc:hasLegalName ?companyName .
 ?company voc:hasCEO ?ceo .
 ?ceo voc:DOB "January 1, 1970" .
}

- Returns:

companyName

...

"ACME Ltd"

...

Ontologies : OWL

- OWL = Web Ontology Language
- Basé sur les logiques de description
- Utilise RDF comme support

EquivalenClasses(ex:Femme intersectionOf(ex:Personne ex:Féminine))

