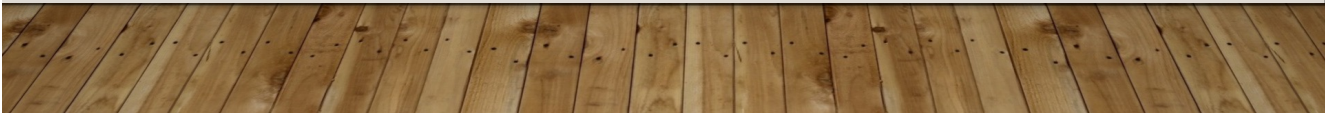


## 70 LANGAGES DE REPRÉSENTATION DE CONNAISSANCES

---

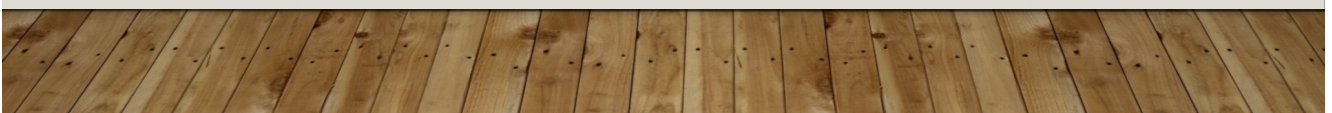
- Les langages de représentation de connaissances se caractérisent par
  - leur syntaxe, c'est-à-dire la manière dont on écrit les phrases ou formules, et
  - la sémantique, c'est-à-dire la manière dont sont interprétées ces formules.
- Plusieurs langages
  - KIF
  - XML
  - RDF et RDFS



## 71 KIF (KNOWLEDGE INTERCHANGE FORMAT)

---

- KIF a été proposé en 1992 par M. R. Genesereth et R. E. Fikes.
- KIF a été conçu pour permettre à l'origine des échanges d'information entre des systèmes disparates (créés par différents programmeurs à différents moments dans des langages différents etc.).
- KIF n'a pas été conçu pour l'interaction avec les humains quoique ces auteurs prétendent que c'est possible.
- KIF est basé sur la Logique du Premier Ordre (LPO) et ressemble beaucoup à Prolog mais avec une syntaxe totalement parenthésée (à la Lisp).



## 72 KIF

- En utilisant KIF comme langage de contenu, un agent peut exprimer:
  - Des propriétés sur les choses
    - Exemples: Farid est jeune; Fatah est cool ; Lucy est une chienne
  - Des relations entre les choses
    - Exemples: Alice et Kenza sont amies ; Lila est la fille de Kenza ;
  - Des propriétés générales sur des ensembles de choses :
    - Exemples: Chaque être humain a une mère ;
- Propriétés générales de KIF:
  - a une sémantique de type déclaratif (sans interprétation)
  - exprime totalement les expressions de (LPO)
  - peut représenter des méta connaissances

## 73 KIF

- Expressions en KIF

- Structures de données

On représente l'information liée à trois salariés sous la forme d'un 3-uple de préfixe **salary**: < **salary** Identificateur, Département, Salaire > :

- (salary 015-46-8700 widgets 72000)
    - (salary 016-44-9832 printing 50000)
    - (salary 017-22-1234 sales 12398000)

On peut représenter des informations plus complexes comme la température de la variable **mI**:

(= (temperature mI) (scalar 80 Celsius))

## 74 KIF

- Définition de concepts nouveaux
  - définir des concepts en fonction de concepts précédemment définis.
  - Par exemple, un célibataire est un homme qui n'est pas marié:

**(defrelation celibataire (?x) := (and (homme ?x) (not (marie ?x))))**

Où:

- **?x** représente une variable
- **homme** et **marie** sont des concepts définis précédemment
- (**and**, **not** comme = et := sont prédéfinis en KIF)

## 75 KIF

- définir une nouvelle relation en fonction des relations déjà existantes comme «<= » en fonctions des relations prédéfinies < et =

**(defrelation <= (?x ?y) := (or (< ?x ?y) (= ?x ?y)))**

- Règles
  - On peut représenter des relations entre des ensembles.
  - Par exemple, toute chose **?x** qui est une personne est automatiquement un mammifère :  
**(defrelation person (?x) => (mammal ?x))**
  - Ou encore pour dire que  $x^n$  | n pair est un nombre positif:  
**(=> (and (real-number ?x) (even-number ?n)  
(> expt ?x ?n) 0))**

## 76 KIF

- Méta connaissances
  - On utilise les symboles «'» et «,» pour quoter des connaissances dans des connaissances.
  - Si **joe** veut savoir tous les items de type **salary** on écrira :  
`(interested joe '(salary, ?x , ?y , ?z))`
  - Si on avait écrit `(salary ?x ?y ?z)` cela voulait dire que **joe** est intéressé par l'expression `(salary ? x ? y ?z)` en elle-même et non en les trois instances qu'elle représente (plus haut).
- Programmes
  - On peut aussi représenter du code (Lisp ou Scheme-like)  
`(prong (new-line t)  
          (print "hello world")  
          (new-line t))`

## 77 XML

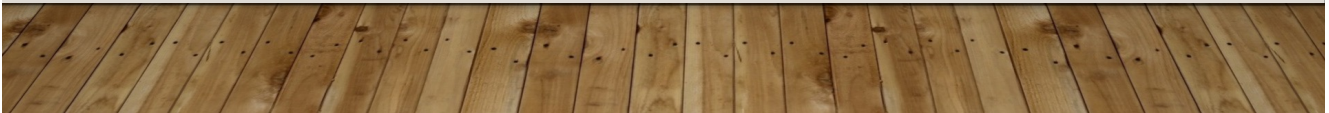
- Qu'est ce que XML?
  - **XML** est l'abréviation d'e**X**tensible Markup **L**anguage.
  - C'est un langage de **balisage** structuré destiné pour:
    - la description,
    - le stockage, et
    - le transfert de données.
  - XML est **indépendante** de toute plateforme.
  - XML est une recommandation W3C.



## 78 XML

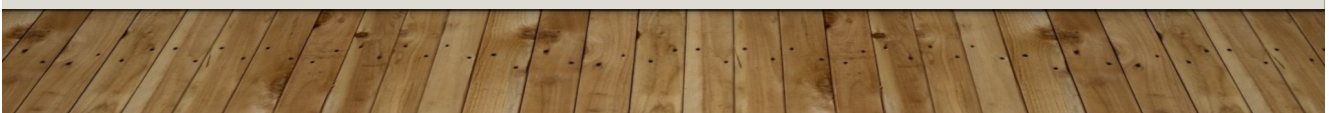
- Différence entre **XML** et **HTML**
  - XML décrit la **structure** de données alors que HTML permet **l'affichage** de données.
  - Les balises HTML sont **prédéfinies** (limitées) alors que les balises XML sont **extensibles** selon le besoin de l'utilisateur.
  - XML ne remplace pas HTML mais ils se **complètent**

Dans la plus part des application web, XML est utilisé pour le stockage et transfert de données alors que HTML est utilisé pour l'affichage.



## 79 XML

- **Balises définissables**
  - L'apport de XML par rapport à HTML c'est la possibilité d'utiliser n'importe quel symbole comme balise :  
**<myTag>**  
ceci est une d'extension d'entité qui est un **myTag**  
**</myTag>**
  - Il faut S'entendre sur :
    - le sens de **myTag** ce qui est fait de manière définitoire par des consortium de standardisation,
    - l'imbrication des balises entre-elles ce qui revient à définir une grammaire.



## 80 XML

- **Exemple:**

Code XML (Atelier1_Exemple1.xml)	Code HTML (Atelier1_Exemple1.html)
<pre>&lt;BIBLIOTHEQUE&gt; &lt;LIVRE&gt; &lt;TITRE&gt;titre livre 1&lt;/TITRE&gt; &lt;AUTEUR&gt;auteur 1&lt;/AUTEUR&gt; &lt;EDITEUR&gt;editeur 1&lt;/EDITEUR&gt; &lt;/LIVRE&gt; &lt;LIVRE&gt; &lt;TITRE&gt;titre livre 2&lt;/TITRE&gt; &lt;AUTEUR&gt;auteur 2&lt;/AUTEUR&gt; &lt;EDITEUR&gt;editeur 2&lt;/EDITEUR&gt; &lt;/LIVRE&gt; &lt;LIVRE&gt; &lt;/LIVRE&gt; &lt;/BIBLIOTHEQUE&gt;</pre>	<pre>&lt;p&gt; &lt;font color="red"&gt; titre livre 1&lt;/font&gt;&lt;br&gt; &lt;b&gt;auteur 1&lt;/b&gt;&lt;br&gt; &lt;u&gt;editeur 1&lt;/u&gt; &lt;/p&gt;  &lt;p&gt; &lt;font color="red"&gt; titre livre 1&lt;/font&gt;&lt;br&gt; &lt;b&gt;auteur 1&lt;/b&gt;&lt;br&gt; &lt;u&gt;editeur 1&lt;/u&gt; &lt;/p&gt;</pre>

## 81 XML

- **Grammaire = DTD**

Un document XML possède deux niveaux :

- **syntaxe** de base pour vérifier
  - l'imbrication correcte des balises sans tenir compte des symboles, et
  - que deux balises ne se chevauchent pas
  - Exemple: à tout **<foo>** correspond un **</foo>**
- **sémantique** (mais qui n'en est pas):
  - tient compte des imbrications réciproques des balises
  - Exemple : une balise **<adresse>** doit obligatoirement contenir dans cet ordre exactement une balise **<rue>**, une balise **<numero>** une balise **<code>** et une balise **<ville>**.

## 82 XML

```
<adresse>
  <rue> Turing </rue>
  <numéro> 2 </numéro>
  <code> 91000 </code>
  <ville> Orsay </ville>
</adresse>
```

- Cette grammaire est décrite dans un **fichier séparé** appelé DTD (Data Type Definition).
- **Exemple de DTD : MathML**

Voici comment Mathematica exporte la formule « $x^2$ » au format MathML, qui est une **DTD** définie par le consortium W3C.

```
Export["test.xml", x^2, "MathML"]
```

Produit le fichier texte «test.xml» suivant:

## 83 XML

```
<?xml version="1.0"?>
<!DOCTYPE math PUBLIC "-//W3C//DTD MathML2.0//EN"
"http://www.w3.org/TR/MathML2/dtd/mathml2.dtd">
<math xmlns="http://www.w3.org/1998/Math/MathML">
  <semantics>

    <msup>
      <mi> x </mi>
      <mn> 2 </mn>
    </msup>

    <annotation-xml encoding="MathML-Content">
      <apply>
        <power/>
        <ci> x </ci>
        <cn type="integer"> 2 </cn>
      </apply>
    </annotation-xml>

  </semantics>
</math>
```

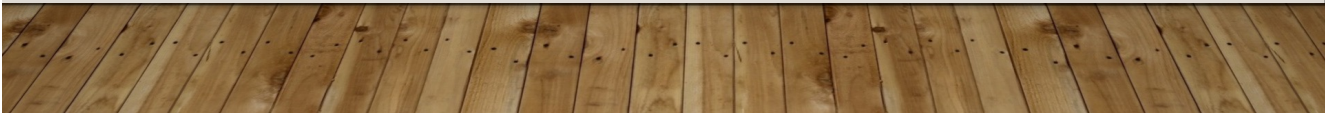
Affichage bidimensionnel

Propriétés mathématiques

## 84 RDF

---

- **Notion de méta données**
  - Les informations de l'Internet, en particulier dans le Web, sont «machine-readable» (compréhensible au niveau lexical –syntaxique) mais pas «machine-understandable»
  - Il faut d'autres informations (méta-informations) émanent de la part de celui qui a produit les informations pour qu'un outil logiciel indépendant puisse traiter ces informations.
- **RDF**
  - W3C a alors élaborer une couche supplémentaire au dessus de XML appelée RDF (ResourceDescription Framework) afin de traiter des méta informations.
  - RDF utilise la syntaxe XML.



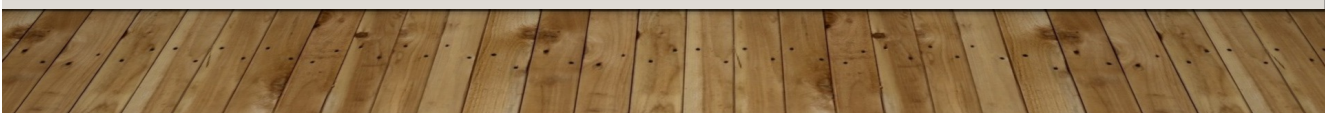
## 85 RDF

---

- **Entités de base de RDF**

En RDF, il y a trois sortes d'entités :

1. Les ressources
2. Les propriétés
3. Les statements (ou assertions)





## 86 RDF

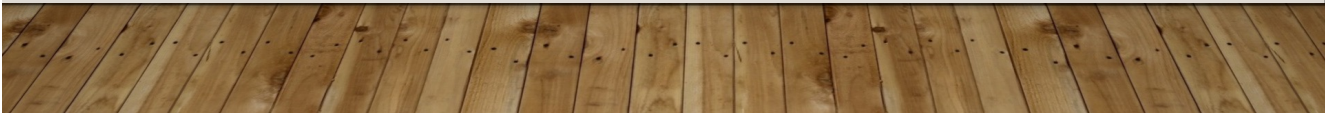
---

- **Ressources**

Les éléments atomiques au sujet desquelles RDF donne des descriptions (les données) sont appelées des ressources :

- Une page web «<http://www.limsi.fr/Individu/jps>»
- Une partie d'une page Web : une ancre
- Un objet accessible via le web: un livre ...

Les ressources sont toujours référées via un pointeur de type URI (Uniform Resource Identifier).



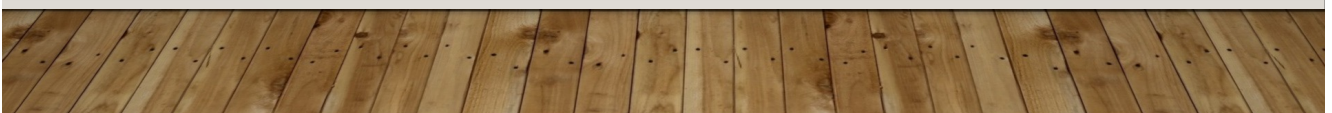
## 87 RDF

---

- **Propriétés**

Les attributs (une caractéristique, un aspect, une propriété, ...) d'une ressource. Chaque propriété a:

- Un sens spécifique
- Un domaine de valeurs autorisé
- Le type de ressources auquel elle s'applique
- Des relations avec d'autres propriétés.



## 88 RDF

- **Statements (ou assertions)**

Un fichier RDF est un ensemble de statements qui sont des assertions de propriétés au sujet de ressources. Un statement est composé:

- **sujet:** Une ressource (URI)
- **prédicat:** Une propriété (son symbole)
- **objet:** La valeur de cette propriété pour la ressource

L'objet, i.e. la valeur d'une propriété, peut être

- Soit un pointeur vers une autre ressource (une URI)
- Soit un littéral : valeur primitive en XML.

## 89 RDF

- **Exemples RDF graphiques**

### Exemple 1

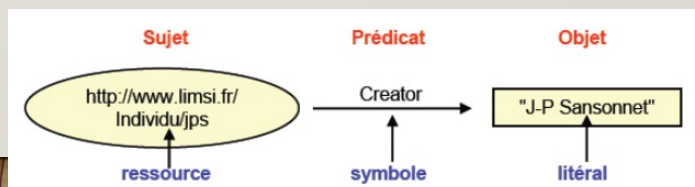
Soit la phrase descriptive suivante : «J-PSansonnet est le créateur de la page <http://www.limsi.fr/Individu/jps>»

Elle peut se décomposer en trois parties :

- Sujet (ressource) : <http://www.limsi.fr/Individu/jps>
- Prédicat (propriété) : Creator
- Objet (littéral) : J-PSansonnet

En RDF, on représente graphiquement un statement en utilisant les notations graphiques suivantes :

- Sujet (ressource) : un cercle
- Prédicat (propriété) : une flèche
- Objet (littéral) : un rectangle.



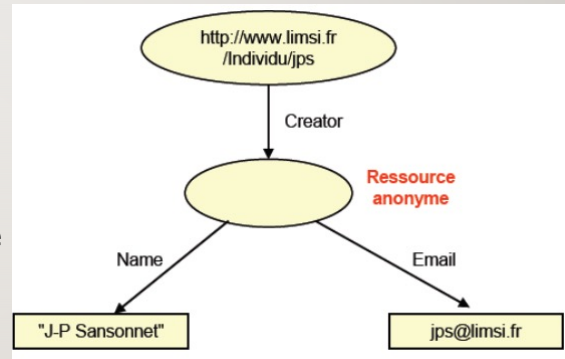
## 90 RDF

### • Exemple 2

Soit la phrase descriptive suivante :

«L'individu dont le nom est J-PSansonnet et dont le mail est [jps@limsi.fr](mailto:jps@limsi.fr) est le créateur de la page <http://www.limsi.fr/Individu/jps>»

- L'objet n'est pas atomique mais une entité structurée.
- En RDF, cette entité est représentée par une ressource (un cercle) mais elle est anonyme c'est-à-dire sans texte à l'intérieur :

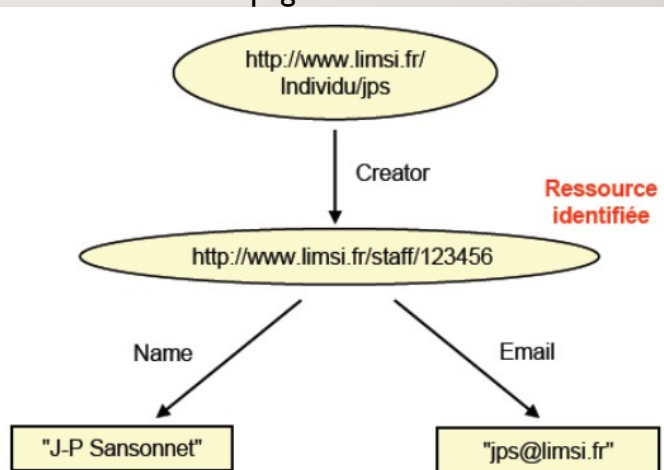


## 91 RDF

### • Exemple 3

Soit la phrase descriptive suivante : «L'individu qui est identifié dans notre base du personnel sous le n° 123456 a pour nom J-PSansonnet et pour mail [jps@limsi.fr](mailto:jps@limsi.fr) ; il est le créateur de la page <http://www.limsi.fr/Individu/jps>»

Il est possible d'identifier la ressource anonyme de l'exemple 2 en pointant sur le fichier du personnel qui est situé à : «<http://www.limsi.fr/staff/123456> ».



## 92 RDF

- **Syntaxe de base de RDF**

- [1] RDF ::= ['<rdf:RDF>'] description\* ['</rdf:RDF>']
  - [2] description ::= '<rdf:Description' idAboutAttr? '>' propertyElt\* '</rdf:Description>'
  - [3] idAboutAttr ::= idAttr | aboutAttr
  - [4] aboutAttr ::= 'about=' URI-reference''
  - [5] idAttr ::= 'ID=' IDsymbol''
  - [6] propertyElt ::= '<' propName '>' value '</' propName '>|' '<' propName resourceAttr '/>'
  - [7] propName ::= QName
  - [8] value ::= description | string
  - [9] resourceAttr ::= 'resource=' URI-reference''
  - [10] QName ::= [ NSprefix ':' ] name
  - [11] URI-reference ::= string, interpreted per [URI]
  - [12] IDsymbol ::= (any legal XML namesymbol)
  - [13] name ::= (any legal XML namesymbol)
  - [14] NSprefix ::= (any legal XML namespace prefix)
  - [15] string ::= (any XML text, with "<", ">", and "&" escaped)
- Un élément RDF est un simple wrapper qui délimite dans un document XML un contenu qui est du RDF.

## 93 RDF

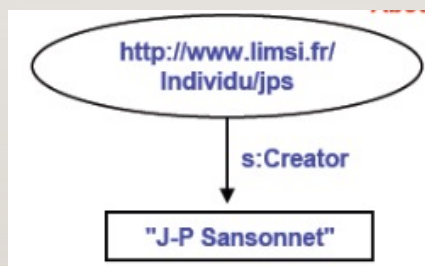
- **Exemple I**

La phrase descriptive «J-PSansonnet est le créateur de la page

<http://www.limsi.fr/Individu/jps>»

s'exprime en suivant la syntaxe textuelle de RDF :

```
<rdf:RDF>
  <rdf:Description about="http://www.limsi.fr/Individu/jps">
    <s:Creator>J-PSansonnet</s:Creator>
  </rdf:Description>
</rdf:RDF>
```





## 94 RDF

Normalement cette description est incluse dans un fichier XML qui déclare les préfixes comme **rdf:RDF** et surtout les namespaces utilisés comme **:rdf** et **:s**.

Le document XML complet est :

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:s="http://description.org/schema/">

  <rdf:Description about="http://www.limsi.fr/Individu/jps">
    <s:Creator>J-PSansonnet</s:Creator>
  </rdf:Description>

</rdf:RDF>
```

Déclaration des domaines  
de tags utilisés

## 95 RDF

### • Exemple 3

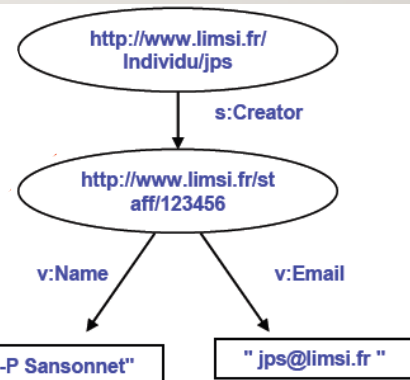
La phrase descriptive «l'individu qui est identifié dans notre base du personnel sous le n° 123456 a pour nom J-PSansonnet et pour mail [jps@limsi.fr](mailto:jps@limsi.fr) ; il est le créateur de la page <http://www.limsi.fr/Individu/jps> »

S'exprime en suivant la syntaxe textuelle de RDF :

```
<?xml version="1.0"?>
<rdf:RDF>
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:s="http://description.org/schema/"
  xmlns:v="http://description.org/schema/.."
  <rdf:Description about="http://www.limsi.fr/Individu/jps">
    <s:Creator rdf:resource="http://www.limsi.fr/staff/123456"/>
  </rdf:Description>

  <rdf:Description about="http://www.limsi.fr/staff/123456">
    <v:Name>J-PSansonnet</v:Name>
    <v:Email>jps@limsi.fr</v:Email>
  </rdf:Description>

</rdf:RDF>
```



## 96 RDF

- **Types de containers RDF**

Souvent, il est nécessaire de référer non pas à une entité isolée mais à un agrégat d'entités. RDF définit trois types d'agrégats :

- **Bag**: un ensemble **non ordonné** de ressources ou de littéraux. Il peut y avoir des répliques dans un bag. (RDF ne propose pas le concept d'ensemble au sens mathématique du terme où les éléments ne sont pas répliqués).
- **Séquence**: une liste **ordonnée** de ressources ou de littéraux. Il peut y avoir des répliques dans un bag.
- **Alternative**: un ensemble **non ordonné** de ressources ou de littéraux qui représentent des possibilités diverses pour une même valeur de propriété. Il ne peut pas y avoir de réplique.

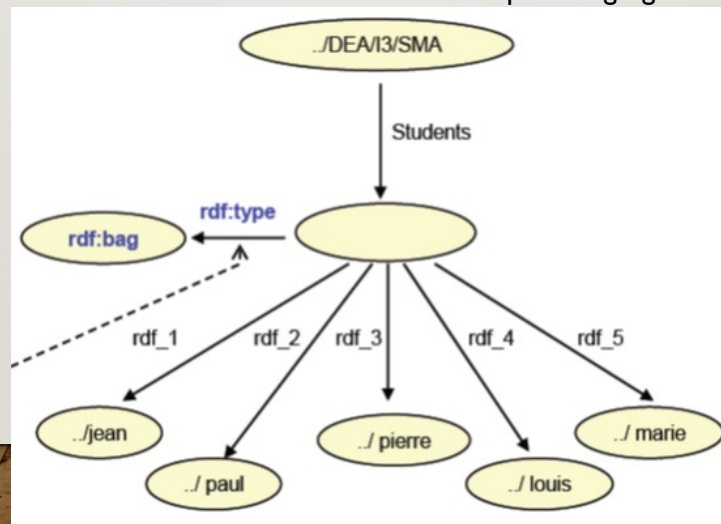
Ces types d'agrégats sont déclarés dans une propriété supplémentaire qui est une relation **rdf:type**

- **Fonction d'accès** : on peut référer un élément dans un agrégat RDF en utilisant la syntaxe "\_1", "\_2", "\_3", etc.

## 97 RDF

- **Exemple**

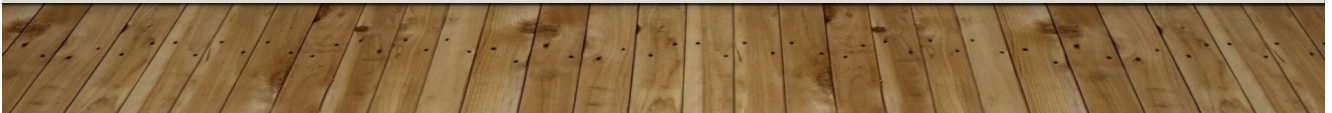
La phrase «Les étudiants du cours du cours DEA/I3/SMA sont jean paul pierre louis et marie» est modélisée en RDF par un agrégat de type bag:



## 98 CONCLUSION SUR LA REPRÉSENTATION DE LA CONNAISSANCE

---

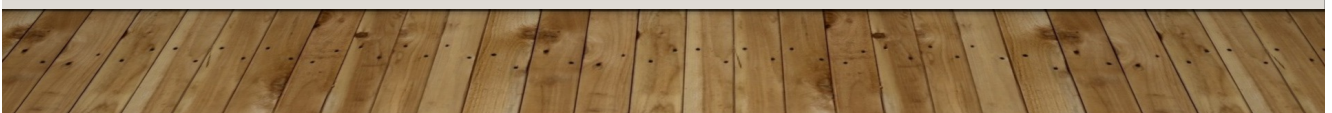
- Richesse expressive  $\Leftrightarrow$  complexité
- Lecture toujours difficile
- Mécanismes de « calcul » (projection, subsumption, etc.) ne permettent pas des expressions de requêtes toujours « simples » (il faut reformuler).
- Ne règle pas le « frame » problem...



## 99 LES SYSTÈMES DE REPRÉSENTATION DES CONNAISSANCES

---

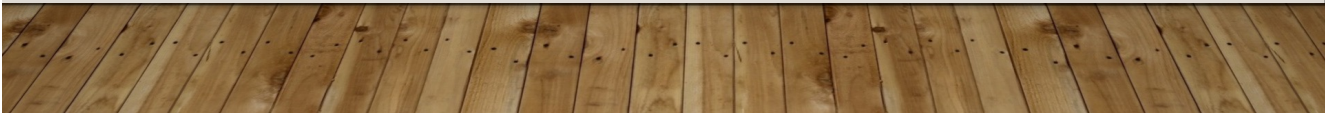
- **KL-One**
- **CLASSIC**
- **LOOM**



## I00 KL-ONE

---

- KL-One est un langage de représentation des connaissances dont la conception a débuté au début des années 70.
- Les propriétés générales de KL-One sont :
  - Spécialisé dans la représentation des concepts.
  - Se situe à la frontière entre les réseaux sémantiques et les frames, donc proche de la langue naturelle avec la notion de sémantique lexicale.
  - Propose uniquement comme outils de raisonnement la classification de concepts.
  - Muni d'une sémantique de type LPO.



## I01 KL-ONE

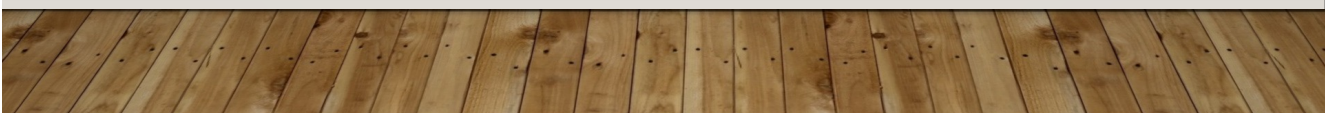
---

- Les concepts de KL-One sont munis:
  - d'attributs, appelés rôles, qui expriment les relations entre concepts ;
  - des contraintes sur le type, et
  - la cardinalité de la valeur du type.

Les contraintes et la cardinalité sont associées à tout rôle, à la manière des facets d'un langage de frames.
- Cardinalité d'un rôle

La cardinalité d'un rôle ou encore son «arité» est une couple d'entiers (min, max) spécifiant :

  - min : le plus petit nombre d'arguments que prend la relation/rôle
  - max : respectivement le plus grand (ce peut être  $\infty$ ).



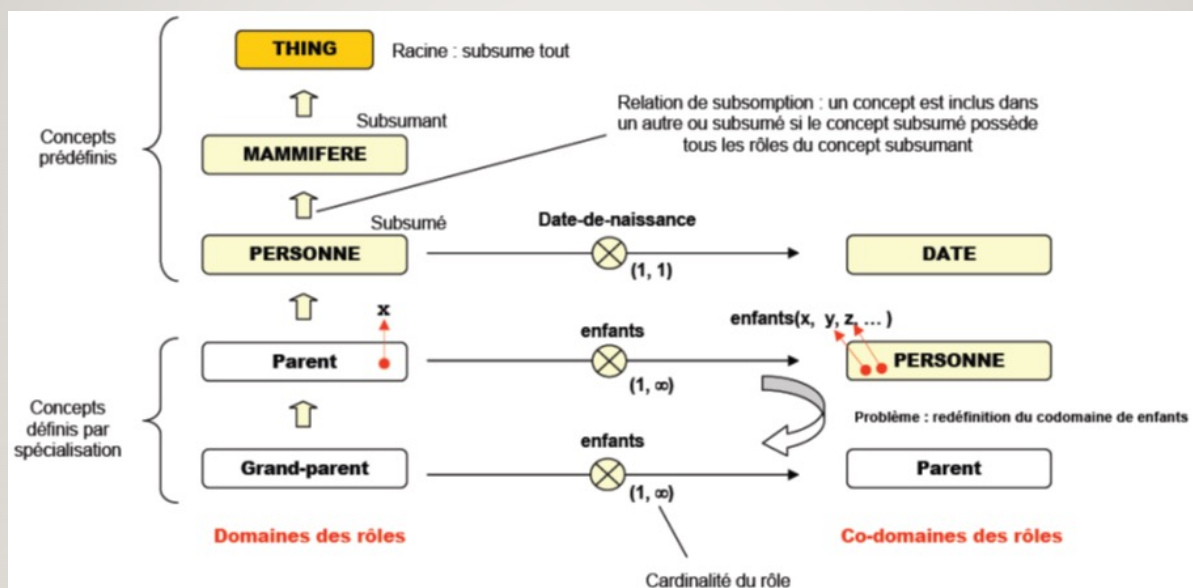


## 102 KL-ONE

### • Exemple KL-One

- On définit le concept **Parent** par :  $\langle x \in \text{Parent} \Leftrightarrow x \in \text{PERSONNE} \wedge \exists y \in \text{PERSONNE} \mid \text{enfants}(x, y, \dots) \rangle$
- Autrement dit, il y a une **PERSONNE** **y** qui apparaît comme un des arguments de la relation **enfants(x, arg1, arg2, ...)**
- Un concept est défini par spécialisation c'est-à-dire qu'il hérite de tous les rôles du concept dont il est tiré: tout **x**  $\in \text{Parent}$  possède automatiquement un rôle **date-de-naissance** même si cela n'est pas explicité dans le schéma :

## 103 KL-ONE

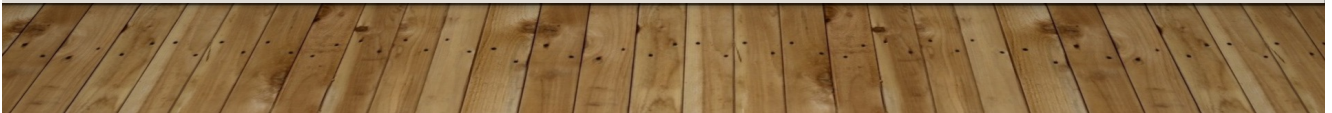


## I04 KL-ONE

---

- **Deux sortes de concepts**

- **Prédéfinis:** donnés par le système et considérés comme génériques, i.e. admis comme tels par «tout le monde ». Ils sont notés tout en MAJUSCULES.
- **Définis:** construits par l'utilisateur de KL-One à partir des types prédéfinis. Ils sont notés avec une Majuscule.
- Les rôles sont en minuscule.



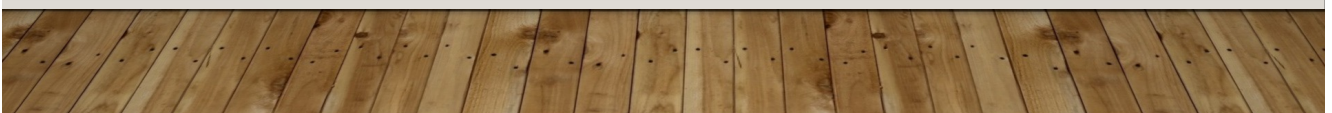
## I05 KL-ONE

---

- **Notion de spécification**

Deux modes de spécification sont proposés par KL-One:

- Les concepts **prédéfinis** sont **incomplètement spécifiés**.
  - Ils expriment des conditions d'appartenance nécessaires mais pas suffisantes
  - Exemple: si un individu est une PERSONNE alors il a une date de naissance mais l'inverse n'est pas vrai; un ANIMAL possède une date de naissance mais n'est pas une PERSONNE.
- Les concepts **définis** sont considérés comme **complètement spécifiés**:
  - Ils expriment des conditions d'appartenance nécessaires et suffisantes
  - Exemple: si un individu possède et vérifie tous les rôles d'un concept défini alors il appartient au concept et vice versa.



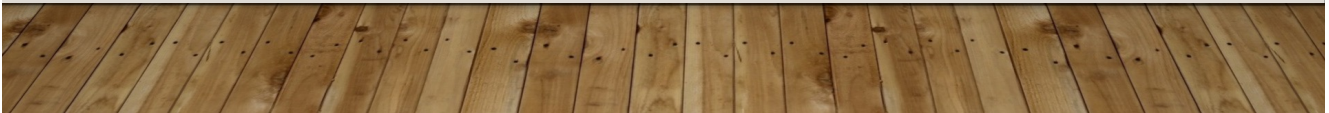
## I06 CLASSIC

Le système opérationnel CLASSIC est l'implémentation la plus connue de langage de DL.

Sa syntaxe est la suivante :

Concept  $\rightarrow$  THING | Conceptname

- | And(Concept+)
- | All(RoleName, Concept)
- | AtLeast(Integer, RoleName)
- | AtMost(Integer, RoleName)
- | Fils(RoleName, Individualname+)
- | SameAs(attribut, attribut)
- | OneOf(Individualname\*)
- | Test (functionarg\*)
- **OneOf**: permet de définir un type énuméré
- **SameAs**: permet de dire que la valeur d'un attribut est la même qu'un autre (un enfant à le même nom que son père).
- **Fils**: associe une valeur élémentaire à un rôle.
- **Test**: fonction externe qui renvoie des valeurs prédicatives à trois états : T, Nil ou ?.



## I07 CLASSIC

### • Règles

- L'originalité du langage CLASSIC est d'associer des règles aux concepts.
- Si un individu est instancié (c'est-à-dire classé) comme un concept C alors la règle R(C) est appliquée :

Musicien-debutant  $\equiv$  (AND Musicien

(ATMOST 1 instrument)

(ATMOST 1 instrument)

(ALL pratique Faible) )

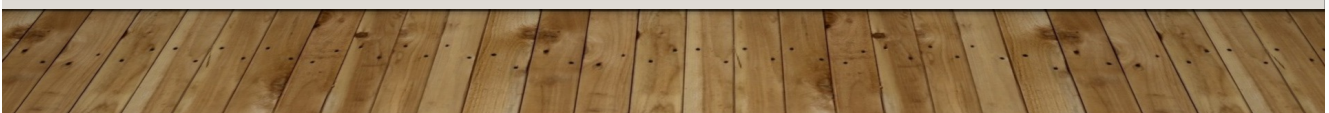
(RULE

instrument-musicien-debutant

Musicien-debutant

(ALL instrument instrument-etude) )

Si un individu **#joe** est classé comme **Musicien-debutant** alors il est automatiquement subsumé par l'expression **(ALL instrument instrument-etude)**.





## I08 CLASSIC

---

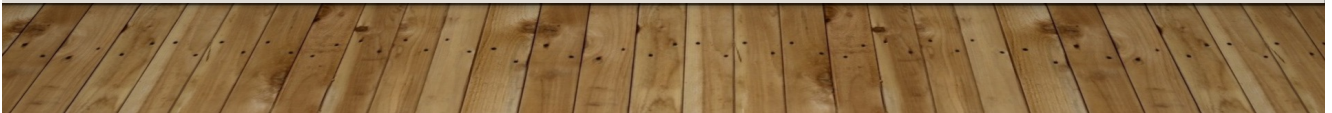
- **Requêtes**

- Toute expression peut s'interpréter comme une requête.
- Par exemple, l'extension de l'expression conceptuelle

**E  $\equiv$  (ATLEAST 2 enfant)**

fournit l'ensemble des individus qui satisfont E.

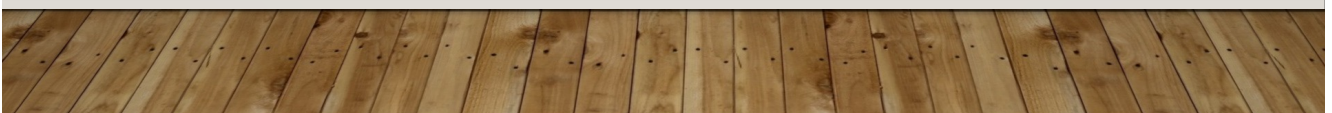
- Plus généralement, une requête est représentée par un concept Q et
- satisfaire la requête Q consiste à classer Q dans la hiérarchie de concepts, i.e. à trouver le plus petit concept qui subsume Q.



## I09 LOOM

---

- LOOM est une plate-forme pour la représentation des connaissances.
- LOOM est un langage à base de frames dans la tradition de KL-ONE avec une sémantique formelle qui mappe les déclarations dans LOOM aux déclarations dans théorie des ensembles et Logique du premier ordre.
- Les connaissances déclaratives dans LOOM sont composées de définitions, de règles, de faits, ...
- Pour compiler ces connaissances, LOOM se sert d'un moteur déductif qui en réalité est un "classifieur" qui utilise l'unification sémantique ainsi que l'orientées objet.





# I 10 CONCLUSION

---

- Divers formalismes de représentation de connaissances:
  - Réseaux sémantiques et graphes conceptuels
  - Frames et scripts
  - Logiques classiques (propositionnelle, prédicats)
  - Logiques non classiques (floue, modale, etc.)
- Plusieurs langages de représentation de connaissances:
  - KIF
  - XML
  - RDF et RDFS
- Systèmes de représentation de connaissances:
  - KL-one
  - CLASSIC
  - LOOM

