

# Atelier ARCo

Pierre-Antoine Champin (LIRIS – Lyon 1)  
Emmanuel Sander (Paragraphe – Paris 8)

Semaine de la connaissance - Nantes  
27 juin 2006

# Plan de la présentation

- Pré-supposés
- Expérience
- Résultats
- Perspectives

# 1. Pré-supposés

- La naïveté de l'utilisateur peut se situer à plusieurs niveaux, qui nous donnent une grille de lecture pour l'expérience
  - naïveté par rapport à l'outil
    - son ergonomie
  - naïveté par rapport au formalisme
    - sa pertinence cognitive
  - naïveté par rapport à l'ontologie formelle
    - les prises de position ontologique

# Pré-supposés outil

- Toute utilisation d'un logiciel par un utilisateur confronte la logique d'utilisation avec la logique de fonctionnement / conception
  - ontologie naïve = logique d'utilisation
  - ontologie formelle = explicitation de la logique de fonctionnement / conception
- Question : l'accès à l'ontologie formelle peut-elle aider à faire coïncider les deux logiques ?

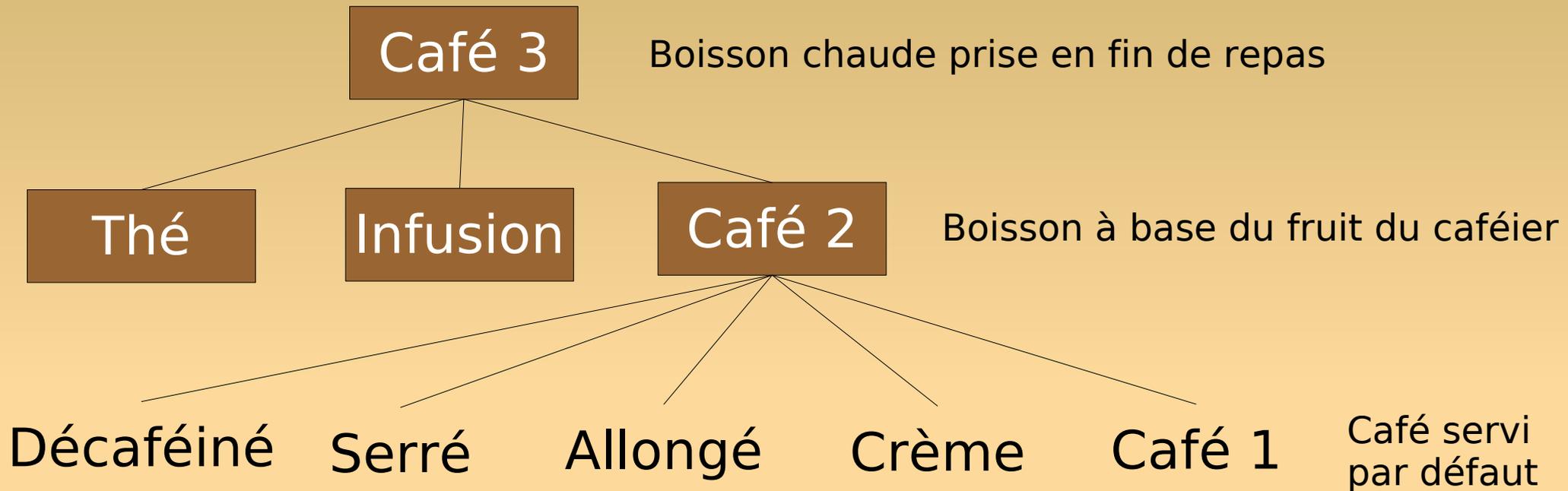
# Pré-supposés naïveté formalisme

- Les inférences formellement valides ne sont pas forcément accessibles pour l'individu
  - exemple : modus tollens
- Les inférences cognitivement valides ne sont pas forcément réalisées/réalisables par la machine
  - exemple :  
année de publication d'un chapitre = année de publication du livre qui le contient  
(pas possible en OWL-DL pur → SWRL)

# Pré-supposés ontologie formelle

- Les connaissances sur le domaine ne coïncident pas forcément avec celles explicitées dans l'ontologie
- Ces décalages peuvent conduire à des effets inattendus
- Question de la polysémie des termes et de la sensibilité au contexte

# Pré-supposés ontologie formelle



« Je voudrais 3 cafés, un déca, un crème et un allongé. »

# Pré-supposés ontologie formelle

**Chercheur 1**

Personne qui fait de la recherche

**Doctorant**

**Enseignant  
Chercheur**

**Chercheur 2**

Chercheur titulaire  
à temps plein

## 2. Expérience

- Outil : Protégé 3.2
- Formalisme : OWL-DL
- Ontologie : chercheurs v0.2
  - recherche académique française
  - personnes, groupes, documents
- Tâche de peuplement  
« page personnelle sémantique » :  
description de soi et de son  
environnement immédiat

# Expérience : tâches (1/2)

- explicitation écrite de l'ontologie naïve
- peuplement de l'ontologie formelle
  - dans Protégé
  - sans accès direct à l'ontologie
  - en commençant par l'utilisateur
- nouvelle explicitation écrite de l'ontologie naïve

# Expérience : peuplement

The screenshot displays the Protégé 3.2 beta application window. The title bar indicates the file path: `file:/home/pa/perso/homepage/emmanuel/emmanuel.pprj, OWL / RDF Files`. The menu bar includes **File**, **Edit**, **Project**, **OWL**, **Code**, **Tools**, **Window**, and **Help**. The toolbar contains various icons for file operations and navigation. The main interface features several tabs: **Metadata**, **OWL Viz**, **Jambalaya**, and **Knowledge Acquisition**. Below these are panels for **OWLClasses**, **Properties**, **Forms**, and **Individuals**. The **INSTANCE EDITOR** is active, showing details for the instance **Emmanuel Sander** (instance of **Maître de conférences**). The editor includes input fields for **prénom** (Emmanuel) and **nom** (Sander). It also features several property lists with selection icons:

- premier auteur de**:
  - Ontologies et apprentissage
  - L'analogie du naïf au créatif
- membre de**:
  - UFR de Psychologie
  - Paragraphe
  - Université Paris 8
- encadrant de thèse de**:
  - Vincent Tambourin
  - Khider Hakem
  - Sylvie Gamo
- auteur de**:
  - Le rôle des interprétations en résoluti
  - Ontologies et apprentissage
  - L'analogie du naïf au créatif
- salarié de**:
  - Université Paris 8
- éditeur de**: (empty list)

# Expérience : tâches (2/2)

- consultation de l'ontologie formelle
  - OWLDoc
  - onglet instances de Protégé
  - OWLViz
  - Jambalaya (relations entre classes)

# Expérience : OWLDoc

The screenshot shows the OWLDoc web application in Mozilla Firefox. The browser window title is "OWLDoc - Mozilla Firefox". The address bar shows the file path: `file:///home/pa/recherche/arco-sdc/owl-doc/index.html`. The page content is organized into several sections:

- Contents:** A sidebar menu with links to [Ontology](#), [All Resources](#), [All Classes](#), [All Object Properties](#), [All Datatype Properties](#), [All Annotation Properties](#), [Individuals](#), and [\[p1\] Resources](#).
- All Classes:** A sidebar menu listing various classes such as [ATER](#), [Actes de conférence](#), [Actes publiés](#), [Article dans actes](#), [Article de revue](#), [Chapitre](#), [ChargéDeRecherche](#), [Chercheur](#), [Collège A](#), [Collège B](#), [Contractuel](#), [Directeur de recherche](#), [Doctorant](#), [Documen simple](#), and [Document](#).
- Class: Chapitre:** The main content area displays the class definition: "Chapitre dans un livre collectif (les chapitres dans un livre "non collectif" ne sont pas des publications à proprement parler).". It lists subclasses: [owl:Thing](#) (with sub-classes [Document](#), [Documen simple](#), [Publication](#), and [Chapitre](#)).
- Super Classes:** A box containing [compose ONLY Livre collectif](#), [Publication](#), and [compose MIN 1 owl:Thing](#).
- Disjoint Classes:** A box containing [Article de revue, Article dans actes](#).
- Usage:** A section titled "Class Description/Definition (Necessary Conditions)" with a box containing [Livre collectif](#).

The browser status bar at the bottom shows the current file path: `file:///home/pa/recherche/arco-sdc/owl-doc/chapitre.html`, along with icons for GP and Adblock.

# Expérience : onglet instances

The screenshot displays the Protégé 3.2 beta interface. The main window title is "emmanuel Protégé 3.2 beta (file:/home/pa/perso/homepage/emmanuel/emmanuel.pprj, OWL / RDF Files)". The menu bar includes File, Edit, Project, OWL, Code, Tools, Window, and Help. The toolbar contains various icons for file operations and navigation. The main workspace is divided into several panes:

- CLASS BROWSER:** Shows a class hierarchy for the project "emmanuel". The hierarchy includes owl:Thing, Document, Fichier, Groupe, Laboratoire (2), UnitéDEnseignem, Établissement, and Personne.
- INSTANCE BROWSER:** Shows instances for the class "Laboratoire". It has tabs for "Asserted" and "Inferred". The "Asserted" tab is active, showing instances: "Paragraphe" and "Équipe Compréhension, Rai".
- INDIVIDUAL EDITOR:** Shows the editor for the individual "Équipe Compréhension, Raisonement et Acquisition des Connaissances". The "nom" property is set to "Équipe Compréhension, Raisonement et Acquisition des Connaissances". The "sous groupe de" property is set to "Paragraphe". The "a pour membre" property is set to a list of names: Khider Hakem, Vincent Tambourin, Jean-François Richard, Rémi Brissiaud, Anne Bastide, and Sylvie Gam o.

A red rectangle highlights the CLASS BROWSER and INSTANCE BROWSER panes.

# Expérience : OWLViz

The screenshot displays the Protégé 3.2 beta interface with the OWLViz plugin active. The main window shows a hierarchical ontology visualization. The root class is `owl:Thing`, which branches into `Document` and `Personne`. `Document` is further divided into `Document composite` and `Document simple`. `Document composite` includes `Actes de conférence` and `Livre collectif`. `Document simple` includes `Livre`, `Livre simple`, `Poster`, and `Publication`. `Livre` is further divided into `Livre simple` and `Actes publiés`. `Personne` includes `Étudiant` and `Salarié`. `Étudiant` includes `Doctorant`, and `Salarié` includes `Contractuel` and `Fonctionnaire`. A `Fichier` class is also shown as a subclass of `owl:Thing`. The `Document simple` class is highlighted with a blue border. The left sidebar shows the 'CLASS BROWSER' for the project 'emmanuel', listing the asserted hierarchy: `Document simple`, `Livre simple`, `Poster`, `Présentation`, and `Publication`. A legend in the bottom right corner provides controls for showing/hiding classes and their children/parents.

emmanuel Protégé 3.2 beta (file:/home/pa/perso/homepage/emmanuel/emmanuel.pprj, OWL / RDF Files)

File Edit Project OWL Code Tools Window Help

protégé

OWLClasses Properties Forms Individuals Metadata OWLViz Jambalaya Knowledge Acquisition

Asserted model Inferred model

CLASS BROWSER

For Project emmanuel

Asserted Hierarchy

- Document simple
  - Livre simple
  - Poster
  - Présentation
  - Publication

owl:Thing

- Document
  - Document composite
    - Actes de conférence
    - Livre collectif
  - Document simple
    - Livre
      - Livre simple
      - Actes publiés
    - Livre simple
    - Poster
    - Publication
    - Présentation
- Personne
  - Étudiant
    - Doctorant
  - Salarié
    - Contractuel
    - Fonctionnaire
- Fichier

Legend:

- Show class
- Show children
- Show parents
- Hide class
- Hide children
- Hide all classes
- Show Info



# 3. Résultats

- influence de l'ontologie formelle
- influence de la tâche
- importance de la hiérarchie de spécialisation (« sorte de ») dans l'ontologie formelle
- « création » d'instance et spatialisation
- robustesse à l'approximation / pertinence cognitive de la sous-spécification, du multi-typage

# Influence de l'ontologie formelle

- Pas de différence notable entre l'approche d'Emmanuel et celle de Pierre-Antoine
  - à part en ce qui concerne la prise en main de l'outil et le formalisme
- La naïveté par rapport à l'ontologie elle-même ne semble pas avoir eu une grande influence sur la tâche de peuplement
  - Les explicitations liées à l'ontologie ont suffisamment contraint la tâche pour que le peuplement soit correct

# Influence de l'ontologie formelle

- Résultat à modérer dans la mesure où Emmanuel n'était pas totalement naïf par rapport au domaine
- Cependant, on peut se demander dans quelle situation un novice total devrait peupler une ontologie

# Influence de la tâche

- Le peuplement en soi n'est pas une tâche naturelle. Un peuplement est contextualisé et orienté par un but, ce qui a des conséquences
- Influence importante du point d'entrée sur le résultat produit.
  - Si on avait débuté par l'équipe, Emmanuel aurait cherché une construction symétrique, alors que dans ce contexte il a voulu être précis le concernant et de plus en plus vague au fur et à mesure de l'éloignement des instances

# Influence de la tâche

- Illusion cognitive qu'un domaine bien connu est facile à décrire de manière exhaustive
  - La tâche est vite apparue comme titanesque, décrire complètement l'ensemble de l'équipe aurait pris des heures, contrairement aux attentes

# Importance de la hiérarchie

- Les deux ontologies naïves (avant et *après* la tâche de peuplement) sont dessinées selon une hiérarchie « partie/tout » et non selon une hiérarchie « sorte de »
  - malgré le fait que l'interface n'utilise que la seconde
- Pendant la tâche de peuplement, en présence d'une difficulté, la dimension « partie/tout » reprend le dessus
  - recherche de la classe *Équipe de recherche* sous la classe *Laboratoire*

# Importance de la hiérarchie

- La hiérarchie la plus « intuitive » semble dépendre du domaine (ou sous-domaine)
  - ici, on suppose que c'est la notion de *Groupe* qui a conduit à adopter la hiérarchie « partie/tout »
  - pas de problème avec les sous-classes de *Personne*
  - hiérarchie la « plus structurante » choisie en priorité ?
    - autres exemples : chien/animal, roue/voiture, plume/aile, farine/gâteau

# Importance de la hiérarchie

- Biais des outils (et des « ontologues ») pour qui la hiérarchie « sorte de » tient une place particulière
- Note à propos de « sorte de » et « partie/tout »
  - ces deux hiérarchies ne sont pas totalement étrangères :  
une classe  $C1$  *spécialise* une classe  $C2 \equiv$   
l'ensemble des instances de  $C1$  est une *partie de* l'ensemble des instances de  $C2$

# Importance de la hiérarchie

- Nécessité de pouvoir passer facilement d'une hiérarchie à l'autre ?
  - possible dans Protégé, mais seulement pour la conception d'ontologie, pas pour le peuplement
  - serait-ce pertinent pour un utilisateur novice ?
- Nécessité de mixer les hiérarchies ?
  - presque fait par Emmanuel dans son explicitation de l'ontologie naïve

# Importance de la hiérarchie

The screenshot displays the Protégé 3.2 beta interface. The main window is titled "<new> Protégé 3.2 beta" and features a menu bar (File, Edit, Project, OWL, Code, Tools, Window, Help) and a toolbar. The interface is divided into several panes:

- SUBCLASS EXPLORER:** Shows the asserted hierarchy for the project. The classes listed are: owl:Thing, Université, UFR, Service\_administratif, Formation, Laboratoire, and Équipe\_de\_recherche.
- EXISTENTIAL RELATIONS...:** This pane is highlighted with a red rectangle. It shows the hierarchy for the "Contient" property. The classes listed are: Université, Service\_administratif, UFR, Formation, Laboratoire, and Équipe\_de\_recherche.
- CLASS EDITOR:** Shows the editor for the class "Université". It includes a table for properties and values, and a list of instances.

Property	Value
rdfs:comment	

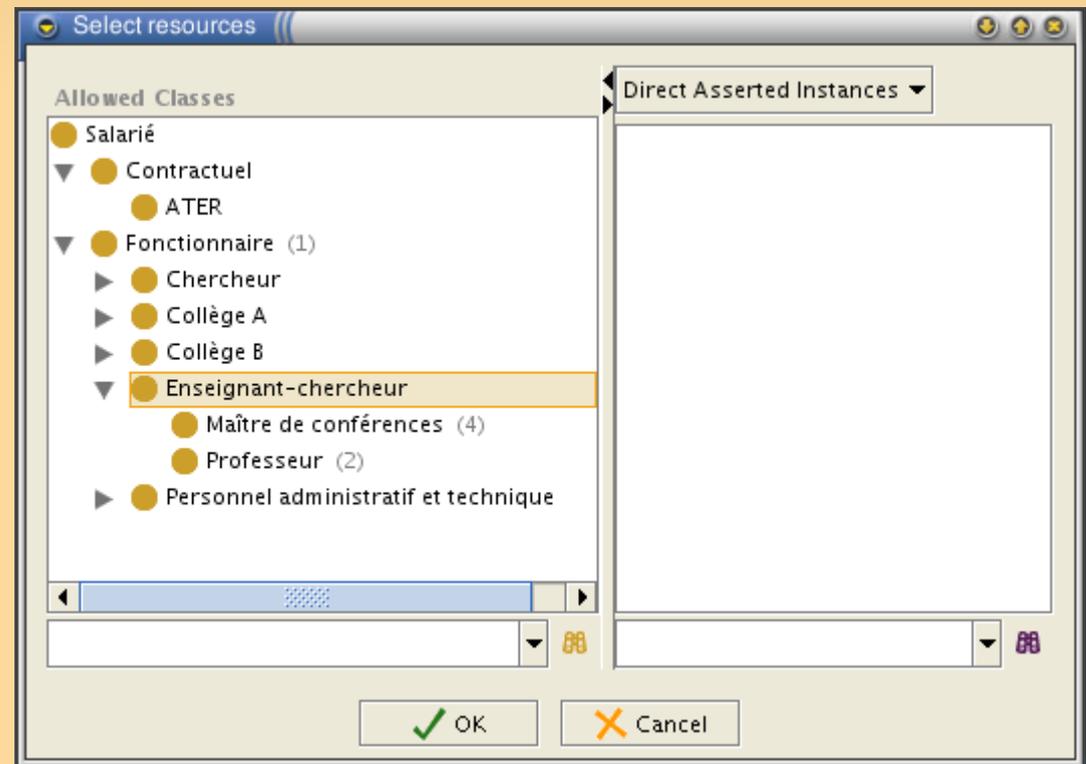
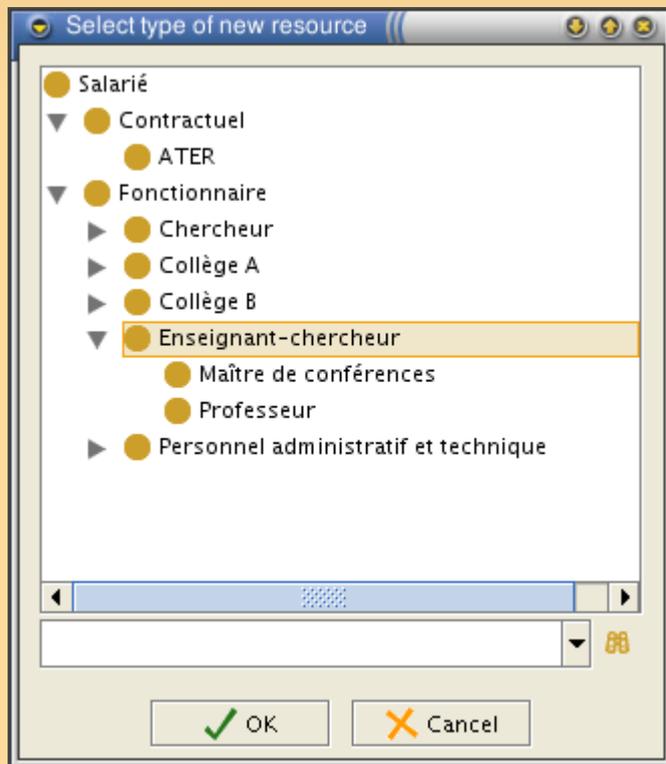
Instances listed in the CLASS EDITOR:

- owl:Thing
- contient **some** UFR
- contient **some** Service\_administratif

The bottom of the interface shows a status bar with "Logic View" selected and "Properties View" unselected.

# « Création » d'instances

- Problème récurrent avec la distinction création/ajout



# « Création » d'instances

- Distinction techno-centrée
  - dans le domaine à décrire, tous les individus « existent » déjà
  - la tâche de peuplement consiste à créer un « représentant » pour ces individus dans la base de connaissance
  - cette distinction entre création et ajout concerne donc ces « représentants », notion abstraite pour l'utilisateur non expert
  - idem pour retrait/suppression : distinction artificielle pour le novice

# Spatialisation des hiérarchies

- Pour les hiérarchies de fichiers, deux présentations co-existent : arborescente (gauche) et « immersive » (droite)
  - les utilisateurs utilisent préférentiellement la présentation immersive pour naviguer
  - les arbres « dépliant » sont réservés au repérage
- Or les outils de conception et peuplement d'ontologie utilisent presque exclusivement la présentation en arbre

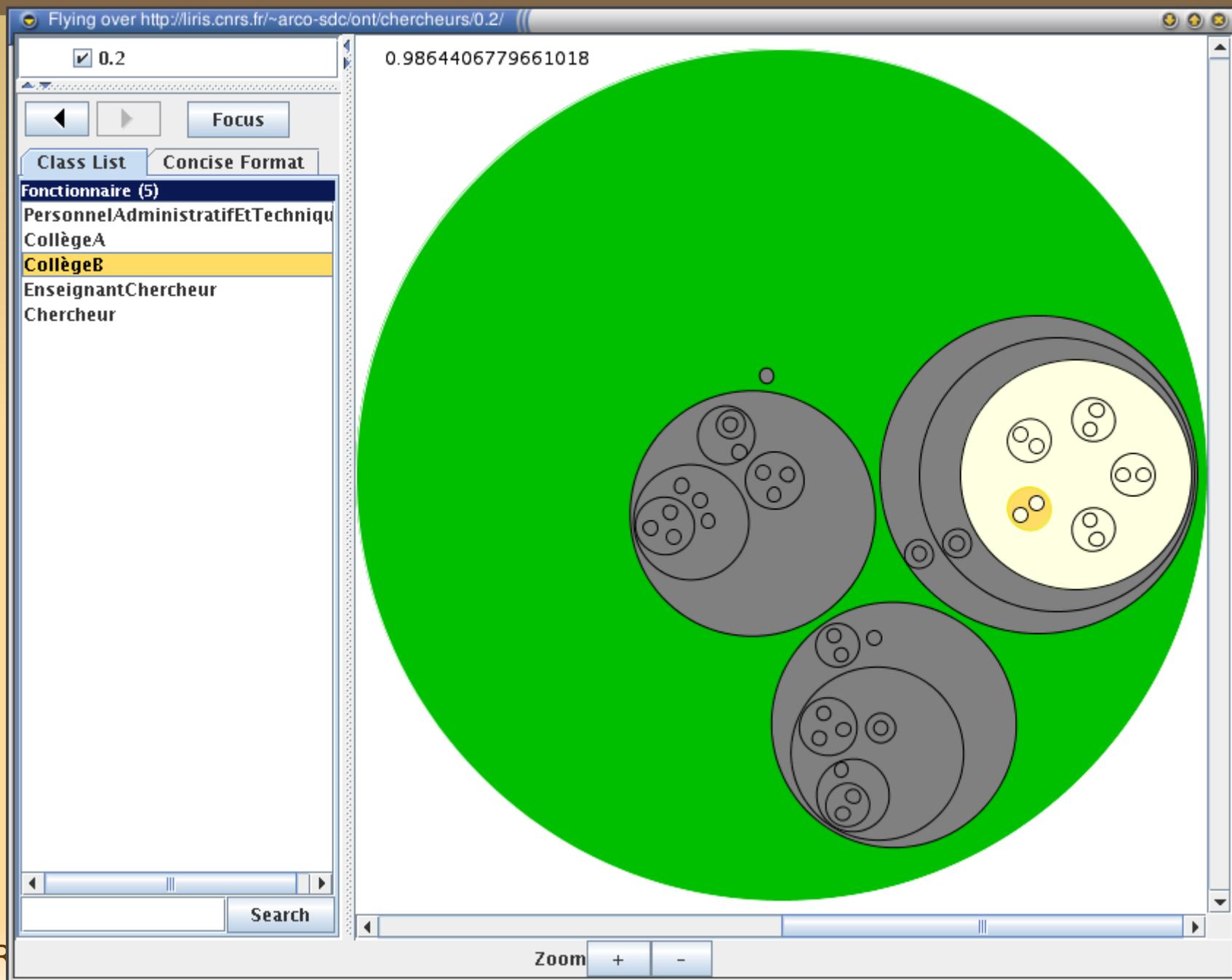
# Spatialisation des hiérarchies

- Certains outils utilisent une présentation en « poupés russes », mais ils sont en général réservés aux utilisateurs experts
  - certains modes de Jambalaya
  - mode « mothership » de Swoop
- Pourtant, cette représentation, fondée sur l'inclusion spatiale, constitue une ressource cognitive
  - Par exemple la transitivité correspond à l'inclusion des contenus

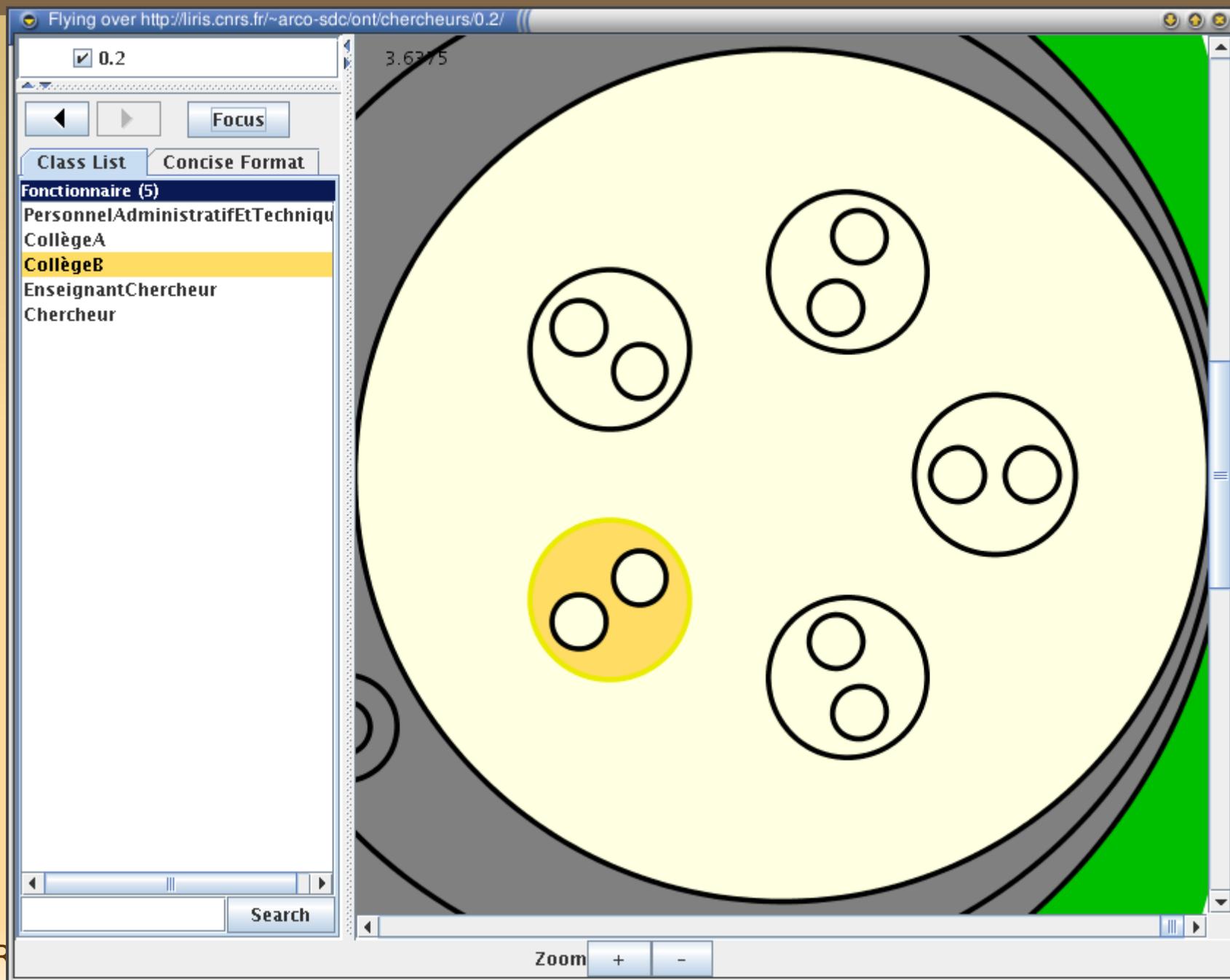
# Spatialisation des hiérarchies

The screenshot displays the Protégé 3.2 beta interface. The main window shows a spatialized hierarchy diagram for the class `owl:Thing`. The diagram is organized into nested rectangular frames. The outermost frame, labeled `owl:Thing`, contains two classes: `Document` and `Fichier`. A red dashed arrow points from `Document` to `Fichier`, and a blue dashed arrow points from `Fichier` to `Document`. Inside the `Document` frame, there is a class `Groupe`. Within the `Groupe` frame, there is a class `Fonctionnaire`. The `Fonctionnaire` frame contains several smaller, unlabeled boxes. A red dashed arrow points from `Groupe` to `Fonctionnaire`, and a blue dashed arrow points from `Fonctionnaire` to `Groupe`. The `Fonctionnaire` frame also contains a label `employé membre`. The interface includes a menu bar (File, Edit, Project, OWL, Code, Tools, Window, Jambalaya, Help), a toolbar, and several panels: **CLASS BROWSER** (For Project: emmanuel) showing a class hierarchy with `Fonctionnaire` selected; **INSTANCE BROWSER** (For Class: Fonctionnaire) showing the instance `Christelle Bosc`; and a **OWLviz** panel with various visualization tools. The status bar at the bottom indicates the hierarchy type: `has subclass and has instance`, and other settings like `Node Labels: Above Node (fixed)`, `Arc Labels: checked`, and `Navigation: Magnify`.

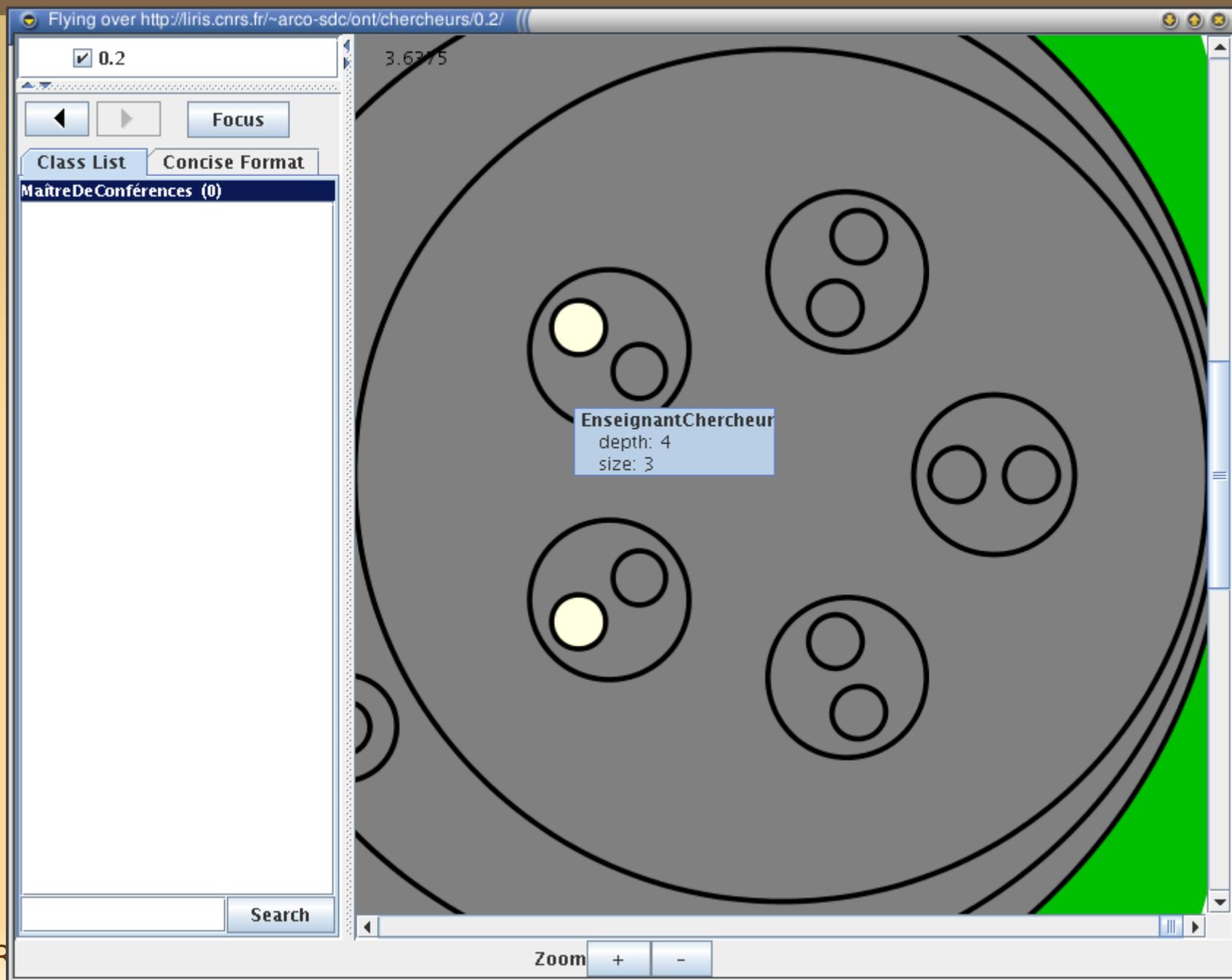
# Spatialisation des hiérarchies



# Spatialisation des hiérarchies



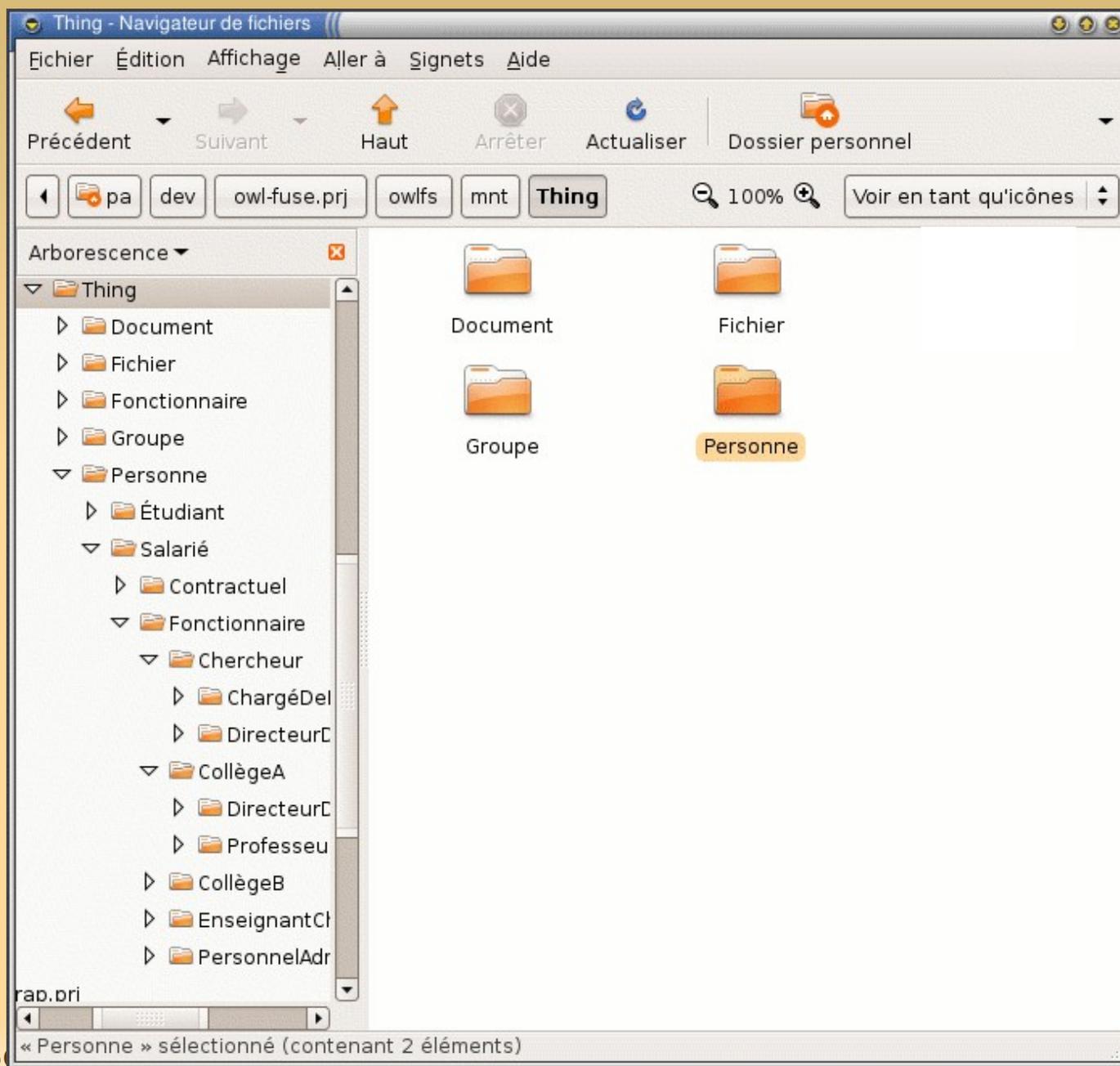
# Spatialisation des hiérarchies



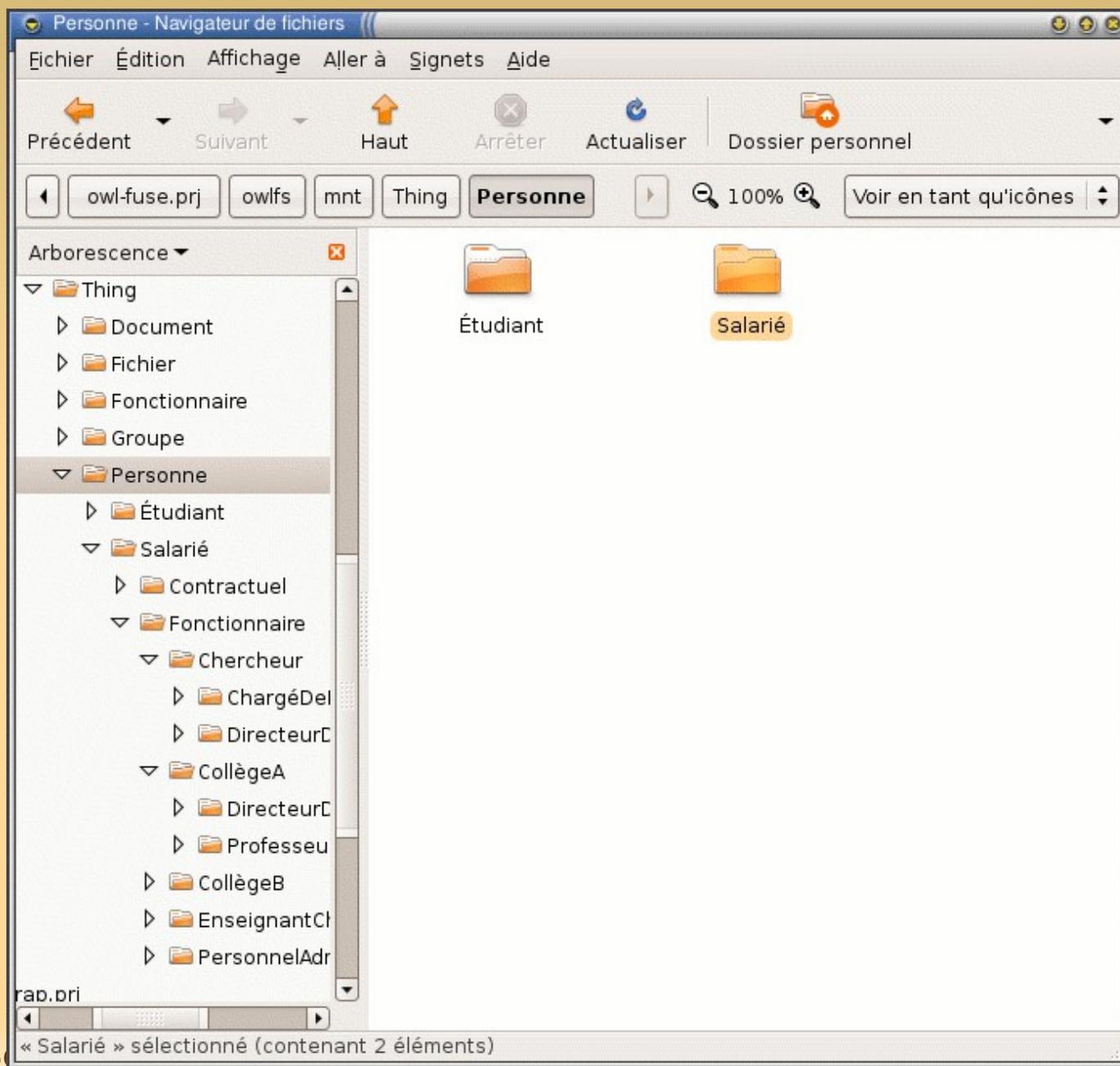
# Spatialisation des hiérarchies

- Proposition : sélection / création d'instance à l'aide d'une interface unifiée, similaire à une interface de sélection / création de fichiers
  - qui pourrait également servir à créer facilement une hiérarchie de classes
  - prototype implémenté sous Linux (concevable en Java)

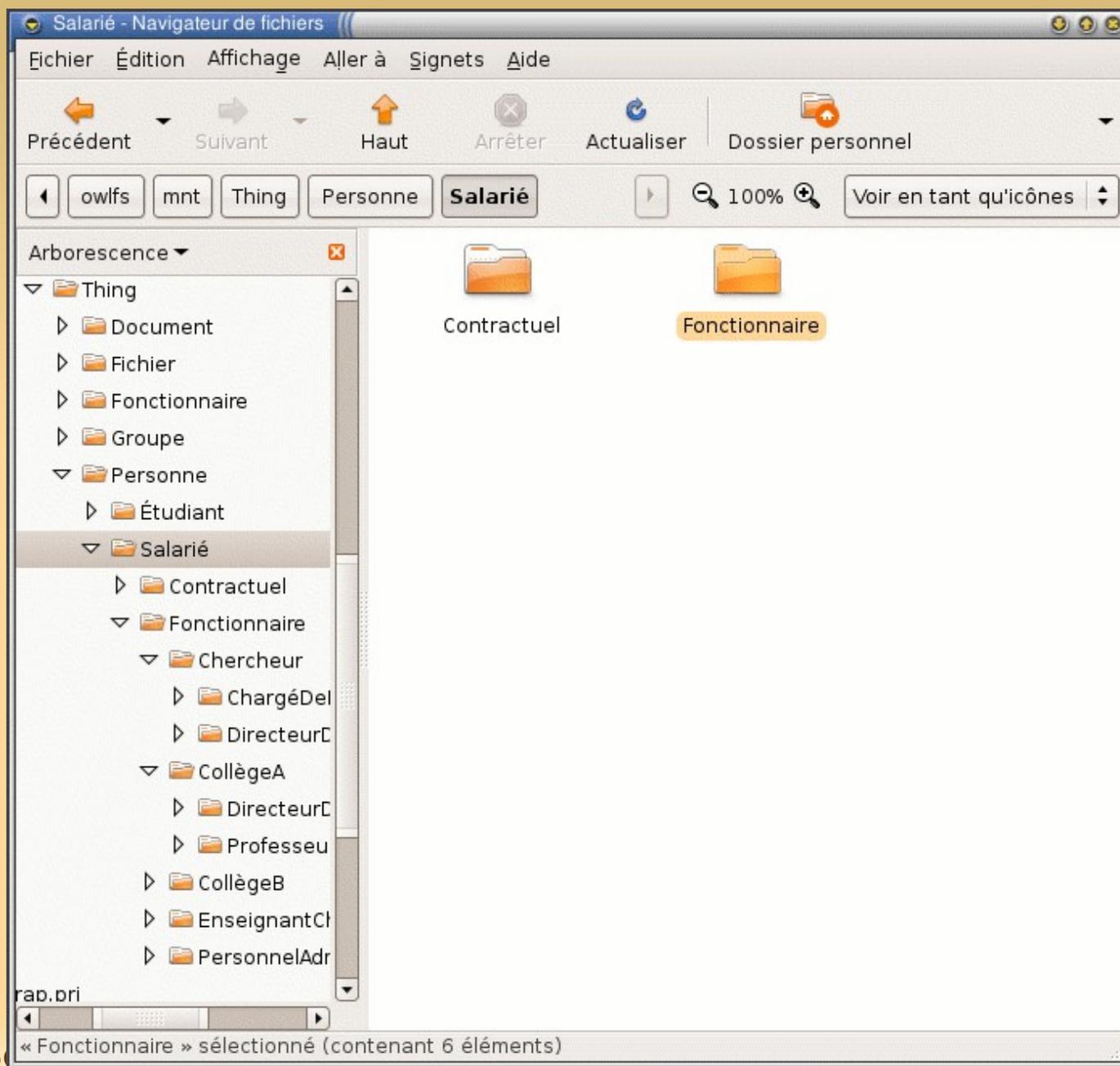
# Spatialisation des hiérarchies



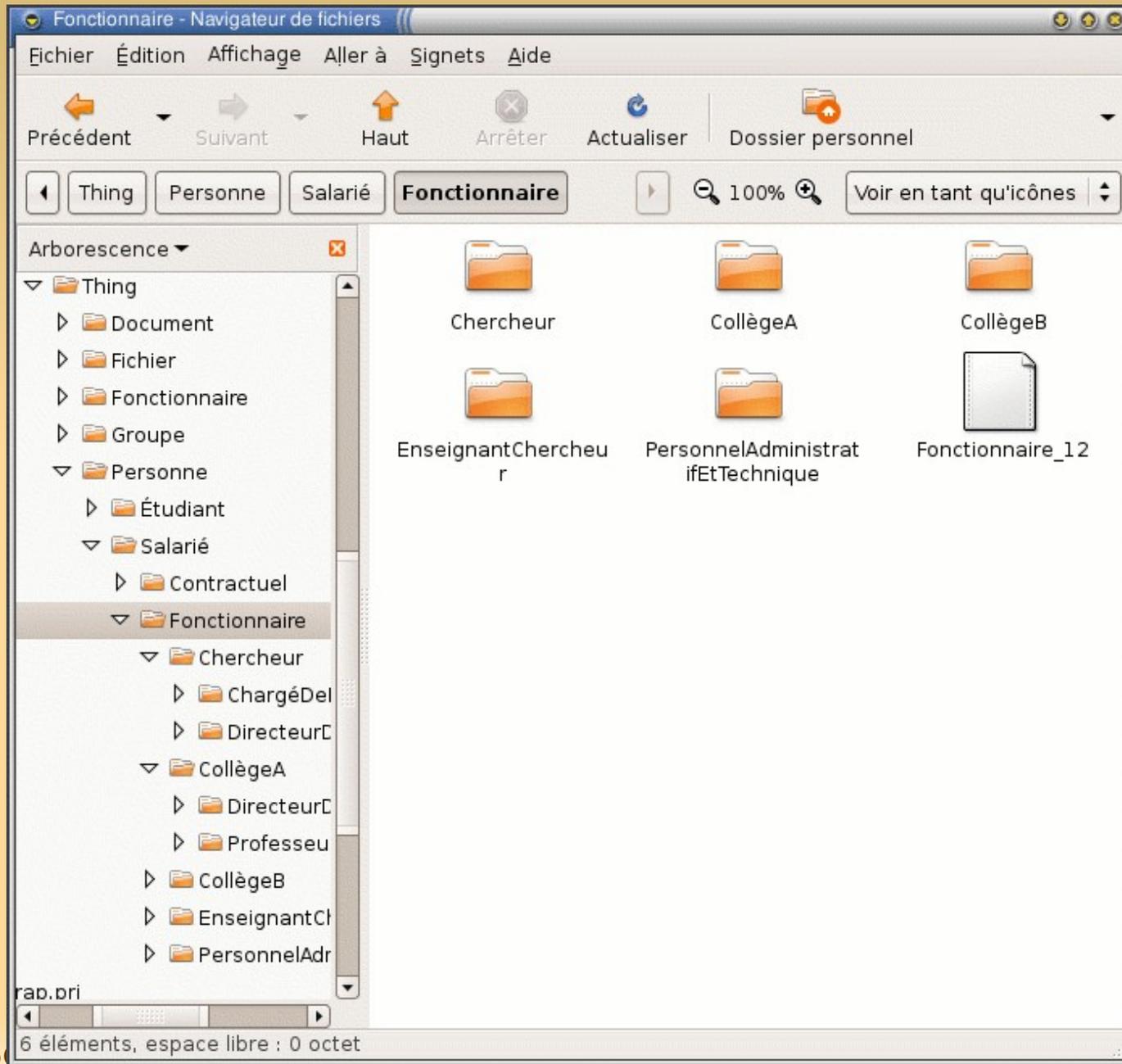
# Spatialisation des hiérarchies



# Spatialisation des hiérarchies



# Spatialisation des hiérarchies



# Absence de classe appropriée

- Solution possibles
  - approximation : utilisation d'une classe proche
    - ex: IUFM assimilé à une université
  - sous-spécification : utilisation de la classe la plus spécifique
    - ex: fonctionnaire
  - multi-typage : utilisation de plusieurs classes appropriées (faute d'avoir une sous-classe commune)
    - ex: établissement public d'enseignement qui n'est ni une université, ni une école d'ingénieur

# Absence de classe appropriée

- Problème avec l'approximation (1)
  - les impacts peuvent être inattendus, et ne sont pas forcément immédiat
  - analogie HTML:
    - H1 utilisé pour mettre du texte en gras
    - conséquence inattendue lors de l'utilisation d'un outil générant une table des matières

# Absence de classe appropriée

- Problème avec l'approximation (2)
  - les inférences faites à l'aide d'une ontologie formelle peuvent avoir les conséquences importantes, la plus gênante étant la contradiction logique, difficile à diagnostiquer et donc à corriger
  - l'approximation est donc une *erreur* du point de vue formel, mais une erreur fréquente, et sans doute inévitable
    - d'autant que l'ontologie et les inférences qu'elle permet ne couvrent pas la totalité du domaine du point de vue de l'utilisateur

# Absence de classe appropriée

- Sous-spécification
  - solution « naturelle » du point de vue du formalisme, d'autant que le moteur d'inférence peut parfois déduire une classe plus spécifique
  - pas forcément intuitive pour un novice, qui s'attend à n'avoir des instances que pour les classes « terminales »

# Absence de classe appropriée

- Multi-typage
  - complément utile de la sous-spécification
  - pas possible facilement avec l'interface de Protégé utilisée dans l'expérience (uniquement depuis l'onglet « Instances »)
  - serait-ce intuitif, même avec une interface appropriée ?

# 4. Perspectives

- Continuer à explorer la problématique de la présentation
  - en poussant l'analogie avec le système de fichiers
  - en cherchant à faire cohabiter plusieurs hiérarchies (feuilles de styles d'ontologies ?)
- Étudier l'impact de ces présentations sur l'ontologie naïve des utilisateurs
  - peut-elle faciliter l'adéquation entre logique d'utilisation et logique de fonctionnement ?

Merci de votre attention

Des questions ?