HADAS

HETEROGENEOUS AUTONOMOUS DISTRIBUTED DATA SERVICES

Conquer: une base de connaissance pour Continuum

Anis Benyelloul
Fabrice Jouanot
Marie-Christine Rousset

Hadas dans Continuum

- Deux thèmes liés à la représentation et à la gestion des connaissances
  - prise en compte du contexte
    - modèle pour le contexte
    - raisonner sur le contexte
  - résoudre l'hétérogénéité
    - au sein de la description du contexte
    - des dispositifs, des services et des tâches

Etat d'avancement aux WPs 1/4

- WP0: Dissémination scientifique et pédagogique
  - publications: encore timide, les premiers résultats vont permettre une meilleure dissémination

A. Benyelloul, F. Jouanot, M.C. Rousset. Conquer: an RDFS-based model for context querying. Ubimob 2010, Lyon.

  - enseignement: cours de M2R, introduction à la problématique de gestion du contexte (illustration dans le contexte Continuum).

Etat d'avancement aux WPs 2/4

- WP1: Identification du cadre socio-économique
  - participation à la définition du scénario prospectif
    - aspect scénario
    - aspect contexte
- WP2: s'adapter au contexte
  - nombreuses réunions HADAS-LIG-Rainbow pour concevoir les modèles et l'architecture de la plate-forme continuum
  - mise en évidence de la nécessité d'une base de connaissance au cœur de Continuum
  - définition et implémentation d'une première version de base de connaissance (Conquer), intégration en cours

Etat d'avancement aux WPs 3/4

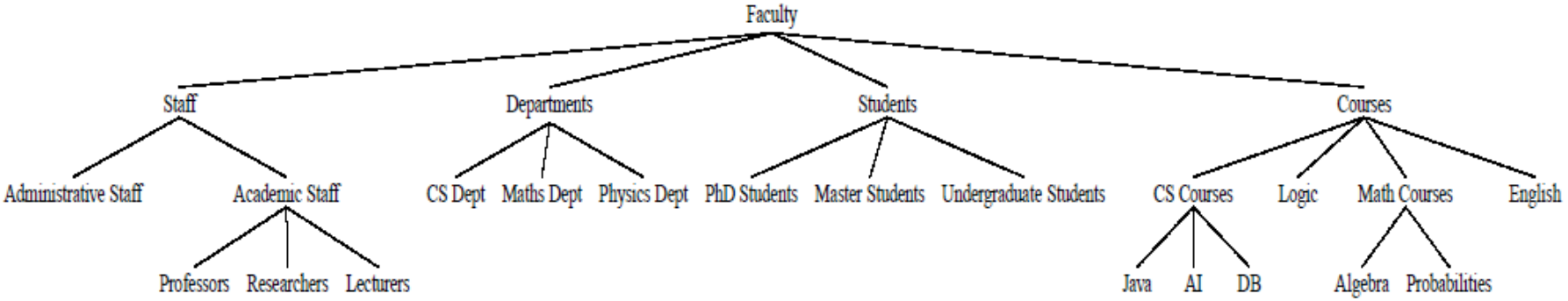
- WP3 & 5: Maîtriser l'hétérogénéité
  - Représentation des informations du contexte dans une base de connaissance
    - modèle du contexte extensible et spécialisable
    - interrogation exploitant les informations sémantiques
    - possibilité d'inférer de nouveaux faits (règles)
  - Implémentation d'un premier prototype
    - démonstrateur autonome
    - prototype accessible via une interface upnp pour intégration à Wcomp (WP5)

Contribution: Conquer, représenter le contexte sous la forme d'une ontologie

- ontologie = description formelle d'un domaine d'intérêt
  - un vocabulaire (classes et propriétés)
  - enrichi par des contraintes sémantiques
    - *Java* peut être une *danse*, une *île*, a *langage de programmation* ou un *cours*
    - la contrainte *java* est une sous classe de *Cours* permet une interprétation correcte
- permet d'utiliser les paradigmes de la logique
  - ensemble d'axiomes et de faits
  - utilisation de la logique de description et des sous-familles avec de bonnes propriétés (complétude, terminaison,performance)
- Cette représentation offre des capacités à raisonner

Un exemple d'ontology

- **Simplement une taxonomie (représentation graphique des contraintes de sous-classes)**



+ **le concept de propriétés (relations) entre classes (RDFS)**

  Teaches (Academic Staff, Courses)
  TeachesTo (Academic Staff, Students)
  Manager (Staff , Departments)

+ **d'autres types de contraintes (OWL)**

  Students disjoint de *Staff*
  Seuls *Professors* ou *Lecturers* peuvent enseigner aux *Undergraduate Students*
  Chaque *Department* doit posséder un unique *Manager* qui est un *Professor*

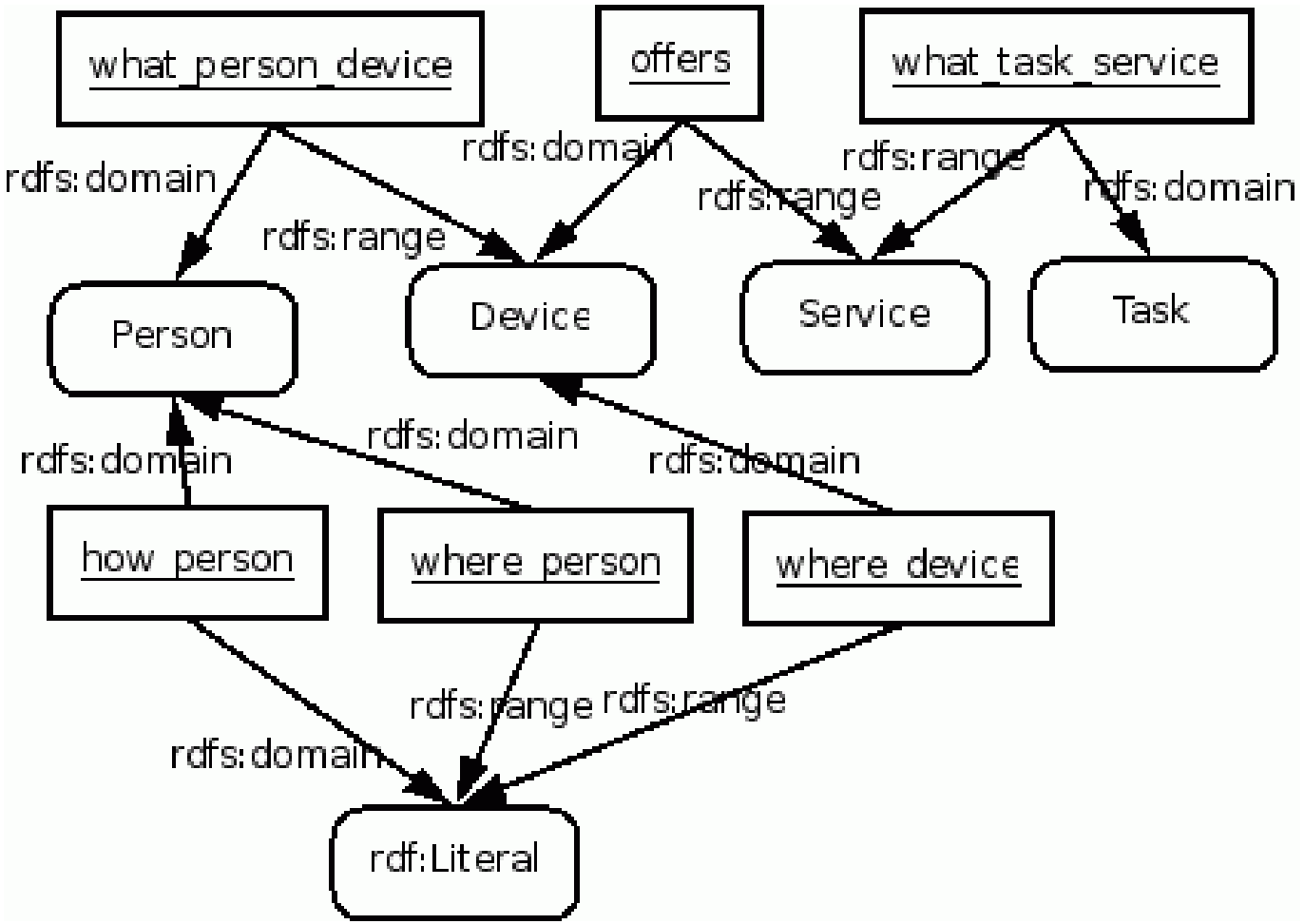
Le contexte peut se représenter sous la forme d'une ontologie

- la modélisation du contexte devient simple
  - aspect déclaratif du modèle et de l'interrogation
  - possibilité d'utiliser les outils du Web Sémantique (RDFS, SPARQl)
- Capacités à raisonnement
  - pour bénéficier d'un langage d'interrogation déclaratif basé sur un moteur d'inférence
  - pour comparer et calculer un facteur de qualité des résultats (pour trouver les meilleurs dispositifs candidats à une tâche)
  - pour inférer de nouveaux faits à l'aide de règles avancées (reconnaissance de l'entrée/sortie d'une personne, etc.)

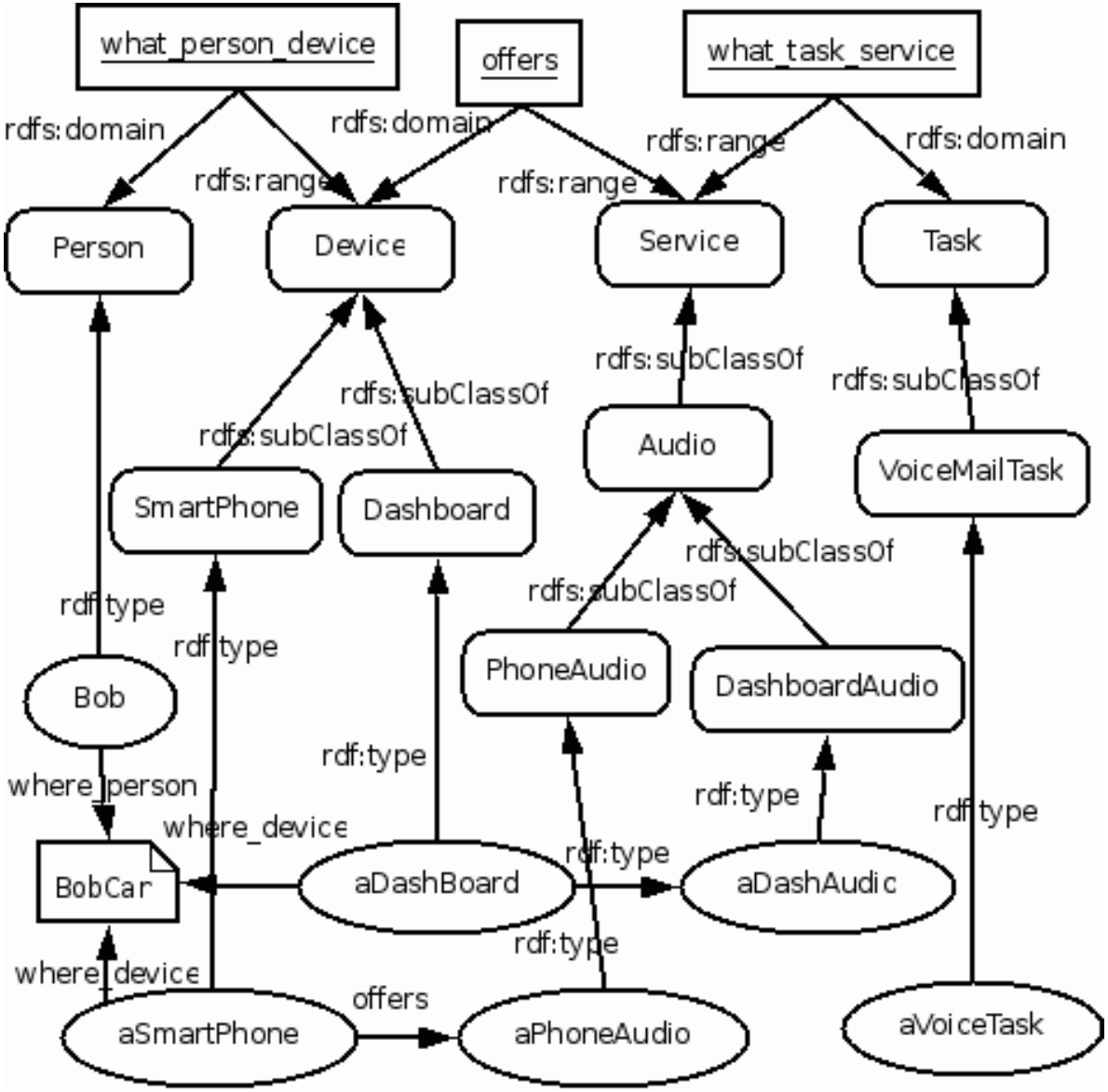
Modèle du contexte 1/2

- Le contexte est représenté à partir de 4 classes de bases dans un schéma RDFS:
  - **Person**: utilisateurs d'un scénario applicatif, en interaction avec des dispositifs, avec les propriétés
    - **What** caractérise le dispositif avec lequel une personne interagit,
    - **Where** spécifie sa localisation courante,
    - **How** modélise le comportement physique de la personne.
  - **Device:** dispositifs physiques disponibles dans l'environnement, avec les propriétés:
    - **offers:** services offerts,
    - **where**: localisation.
  - **Service:** services logiciels offerts par des dispositifs ou requis pour la réalisation d'une tâche.
  - **Task:** objectifs à atteindre, avec la propriété
    - **What**: ensemble de dispositifs nécessaires à la réalisation de la tâche.

Modèle du contexte 2/2



Exemple de contexte



Interrogation 1/2

- Utilisation du langage d'interrogation déclaratif SPARQL
  - L'évaluation correcte des requêtes SPARQL est assurée par l'exécution du moteur SPARQL sur un ensemble de faits saturés.

"*qu'elles sont les dispositifs offrant un service de type audio dans la voiture de Bob ?*"

```
SELECT ?d WHERE { ?d type Device .
  ?d where_device BobCar .
  ?d offers ?s .
  ?s type Audio . }
```

Interrogation 2/2

- Une requête peut s'exprimer sous la forme d'une conjonction de prédicats de logique du premier ordre:

$$q(x): Device(x) \wedge where\_device(x,BobCar) \wedge offer(x,s) \wedge Audio(s)$$

- en considérant les axiomes d'inclusion décrits dans le schéma RDFS du contexte:

$SmarPhone \subseteq Device$ $DashBoard \subseteq Device$

$PhoneAudio \subseteq Audio$ $DashBoardAudio \subseteq Audio$

- L'évaluation de cette expression logique sur un ensemble saturé de faits donne le résultat

{aSmarthPhone,aDashBoard}.

Démonstrateurs

- web service Conquer: implémente une base de faits RDFS et offrent les opérations suivantes pour la manipulée,
  - *Add* : Ajout d'un triplet dans la base ;
  - *Delete* : Suppression d'un triplet ;
  - *Query* : Exécution d'une requête SPARQL sur la base
- Le service maintient une base de faits RDFS saturée.
- Démonstrateur Web http://conquer.liglab.fr/

Softs livrés

- Dispositif upnp Conquer: version upnp du web service Conquer pour intégration au sein de Wcomp. Il offre les opérations suivantes,
  - *Add* : Ajout d'un triplet dans la base ;
  - *Delete* : Suppression d'un triplet ;
  - *Query* : Exécution d'une requête SPARQL sur la base

Livrables livrés (ou presque)

- D2.1+D2.2 (fusion): s'adapter au contexte
  - revoir l'intégration des différents travaux dans le texte
- D3.1: document de spécification du langage, de sa sémantique logique et de ses propriétés inférentielles (reprise détaillée de la section Hadas de D2.2)

Softs et livrables en cours

- Livrable D3.2: étude, description et illustration sur des scénarios pertinents des algorithmes d'inférence sur les alignements et les assemblages.
  - des expérimentations sont en cours pour comparer les résultats par rapport à d'autres outils (Jena).
- Dispositif upnp Conquer:
  - Passerelle d'événement:
    - ajouter les événements à la base et routage sur le gestionnaire de contexte (apparition/disparition de dispositifs)
    - générer de nouveaux événements inférés à partir de règles dans la base

Livrable à repousser

- D3.3: Formalisation logique de certaines propriétés de qualité et de continuité de service; spécification et mise en œuvre des techniques de raisonnement pour les vérifier ou les inférer.
  - demande un retour sur intégration pour mieux cerner les besoins "qualité"
  - demande une étude plus poussée des contraintes à ajouter au modèle du contexte à partir de scénarios complexes.