Managing enterprise applications as dynamic resources in corporate semantic webs

an application scenario for semantic web services.

Fabien Gandon, Moussa Lo, Olivier Corby, Rose Dieng-Kuntz

ACACIA in short…

- Objectives: Offer methodological and software support (i.e. models, methods and tools) for construction, management and diffusion of corporate memories.
- Corporate memory: Explicit and persistent materialization of crucial knowledge and information of an organization to ease access, sharing and reuse by the members of the organization in individual and collective tasks.
- Corporate memories as corporate semantic webs

Corporate web & intranet

- Past and Current work on Corporate Memories (10 minutes)
- Current and future work with Semantic WS (10 minutes)

Corporate semantic Web

- Resources: persons, documents (XML, HTML...), services, software, hardware, etc.
- Ontologies: describing the conceptual vocabulary shared by the organisation communities
- Semantic annotations: on these resources (e.g. persons' skills, document contents, characteristics of services/software/hardware), using the vocabulary defined in the ontologies
- Diffusion on the intranet / corporate web.

CORESE semantic search engine

Ontologies

Documents

XML

Legacy sys.

Users

Schemata in RDFS

Annotations in RDF formed by instances of schema in RDFS

RDF

RDF

Rules

CG Support

CG Base

CG Rules

CG Results

CG Query

UNICODE

XML

NAMESPACES

RDF

RDFS

ONGOLOGY

RULES

Web stack

CG Support

CG Base

CG Rules

CG Results

CG Query

RDF

RDF

CG Support

CG Base

CG Rules

CG Results

CG Query

RDF

RDF

CG Support

CG Base

CG Rules

CG Results

CG Query

RDF

RDF

CG Support

CG Base

CG Rules

CG Results

CG Query

RDF

RDF

CG Support

CG Base

CG Rules

CG Results

CG Query

RDF

RDF

CG Support

CG Base

CG Rules

CG Results

CG Query

RDF

RDF

CG Support

CG Base

CG Rules

CG Results

CG Query

RDF

RDF

CG Support

CG Base

CG Rules

CG Results

CG Query
Typical query on a memory

Find the documents about Java and return the titles and the authors:

```sql
select ?doc c:title ?person where
?doc rdf:type c:Document
?doc c:concern ?topic
?topic rdf:type c:Java
?doc c:title ?title
?title ~ "web"
?doc c:author ?person
```

Request language

- Data and type operators: = <= ~ != …
- XML Schema Datatypes: number, boolean, string, date, etc.
- Natural language: xml:lang="en-us"
- Boolean expressions and / or
- Query the Schema
- Negation of arc, optional arc, paths, approximation, variables on properties, etc.

Discover paths between resources

Find a link between two persons (with maximal length of 4 relations)

```sql
?x rdf:type ex:Person
?y rdf:type ex:Person
```

Approximate search

- Request: Technical Report about Java written by an engineer?
- Approximation: Technical Report → Handout
  Engineer → Team

Production rules

Classify a resource

If a person wrote a Ph.D. thesis on a subject the s/he is a doctor and an expert on that subject.

```sql
?person author ?doc
?doc rdf:type PhDDThesis
?doc concern ?topic
?person expertIn ?topic
?person rdf:type PhD
```

Architecture
**HCI and portal generation**

- Build a list with sub-classes of Person
  
  ```xml
  <select name='ihm_person' title='Profession'>
    <query>
      ?x rdfs:subClassOf c:Person
    </query>
  </select>
  ```

- HTML rendering:
  
  ```html
  Request associated to the list :
  ?p rdf:type get:ihm_person
  ```

**DAI & DFK**

- archives distributed all over organisation
- find best archive for new annotation
- Contract-net (CfP, Proposal, Accept/Reject):

**Corporate distributed knowledge**

**Web services & Enterprise application**

- Transversal use of enterprise modeling
  - End of 90's: enterprise modeling for KM
  - In the past 2 years: technology and application integration can benefit from these models too

- Evolution of KM scenarios
  - Until end of 90's focus on: knowledge capture, storage, access and diffusion
  - More and more often: computation, decision, routing, transformation,… knowledge workflow

- Unified and integrated access to knowledge sources and corporate applications
Memories with a broaden scope

- Corporate memories including:
  - information storage services;
  - information capture services;
  - computation and inference services;
  - information flows management services;
  - information mediation services;
  - information presentation services;

- Resources may be internal or external
  - external standard library / online service;
  - interoperate smoothly and integrate workflows at the business layer.

Corporate semantic web services

- Consider service just like other corporate semantic web resources: annotate them
- Sub set of OWL-S (profile, grounding, ++)
- Rely on CORESE to provide a corporate semantic UDDI

- Access resources (documents, services)

Discover and invoke a service

Service description

```xml
<service:Service rdf:ID="PosteService_Secretaire">
  <service:present rdf:resource="#Profile_Poste_Service_Secretaire"/>
  <service:describes rdf:resource="#PosteSecretaire"/>
  <service:supports rdf:resource="#PosteGrounding_Secretaire"/>
</service:Service>

<profile:Profile rdf:ID="Profile_Poste_Service_Secretaire">
  <profile:presentedBy rdf:resource="#PosteService_Secretaire"/>
  <profile:has_process rdf:resource="#PosteGrounding_Secretaire"/>
  <profile:serviceName>PosteSecretaire</profile:serviceName>
  <profile:hasInput rdf:resource="#PosteSecr_input"/>
  <profile:hasOutput rdf:resource="#PosteSecr_output"/>
</profile:Profile>
```

Input description & extension

```xml
<process:AtomicProcess rdf:ID="PosteSecretaire">
  <process:describes rdf:resource="#PosteService_Secretaire"/>
  <process:hasInput rdf:resource="#PosteSecr_input"/>
  <process:parameterType rdf:resource="#PosteSecr_input"/>
  <process:hasOutput rdf:resource="#PosteSecr_output"/>
  <process:parameterType rdf:resource="#PosteSecr_output"/>
</process:AtomicProcess>
```
Extension parameters OWL-S

```xml
<owl:ObjectProperty rdf:ID="semanticType">
  <rdfs:domain rdf:resource="#Parameter"/>
</owl:ObjectProperty>
```

Cos:rule

```xml
<cos:if>
  ?x rdf:type c:Employee
  ?x c:Name ?n
</cos:if>
<cos:then>
  ?x c:EmployeeName ?n
</cos:then>
</cos:rule>
```

```xml
<i c:Name>"Moussa Lo"</i>
</c:Employee>
```

Composing with memory

Corporate semantic web applications

- Composite services
  - Automatic "sequences" using paths

Ongoing...

- Composing services
  - Manual: Classic scenario = IT manager describes a composed service and publishes its description
  - Macro-recording interface
  - Semi-automatic: finding a path from a given input to a given output
  - Fully-automatic ???
    - Capture and decompose end-users' needs?
    - Even small examples seem to require a lot of domain knowledge
    - Controlled workflow description

Automatic “sequences” using paths
My one cent

- Unified, integrated and simplified descriptions
  - Current implementation of SWS is too complex
  - IMHO: low-hanging fruits & convincing scenarios (RDFS was the right way to start not OWL)
- Full integration has many facets:
  - Description (of course) e.g. WSDL + Grounding + Parameter typing
  - But there are other facets, e.g. WS & SOAP, SOAP & SPARQL, SW Rules, SWS as WS sources
- Dynamic end-users’ interfaces for (composed) semantic web services