Linking the Semantic Web with Existing Sources

Wolfgang May
Institut für Informatik
Universität Freiburg
Germany
`may@informatik.uni-freiburg.de`

DEXA “Web Semantics” Workshop
Aix-en-Provence, Sept. 3, 2002

SEMANTIC WEB

- Lots of documents in the Web
- How to use them?
- Global semantics
- Ontologies

XML

- Self-describing data
- Local DTD/semantics

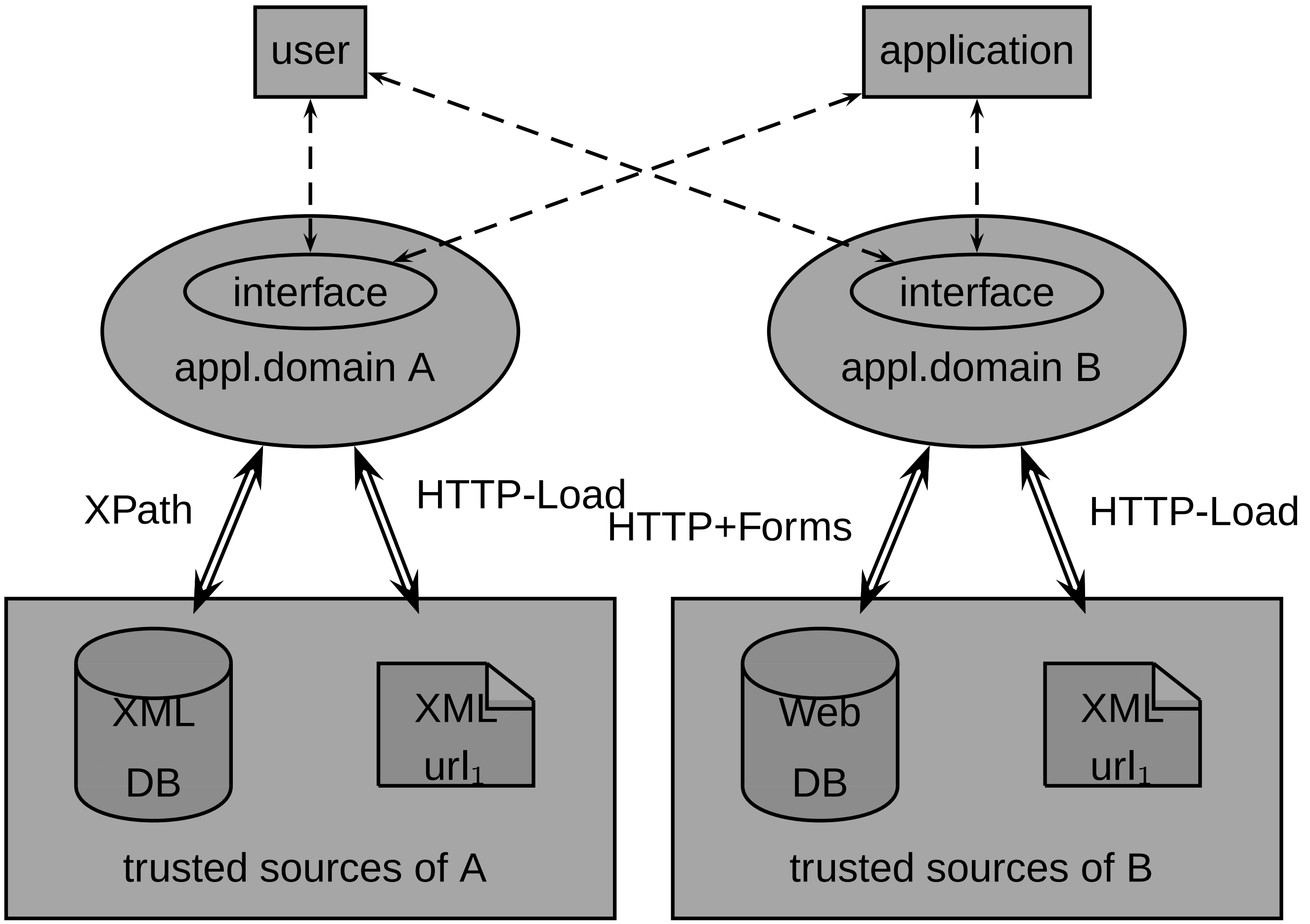
SEMANTIC OVERLAY APPROACH

- integrate the actual data with semantic knowledge by a layer of agents

Applications
and Users

active
Semantic
Layer

autonomous
sources



Web Agents Layer

- We do not place Semantic Web documents in the Web
- but integrate existing sources into the Semantic Web by using
  - semantics-aware agents
  - based on XML technology
- applications or users query the agents of the semantic layer via an XML/XPath interface
- each agent is dedicated to a certain application domain
  - ontological knowledge (DAML+OIL, OWL ...)
  - linked with (autonomous) trustworthy sources
  - expressive reasoning mechanisms
- mediating between semantic queries and plain information sources
- agents transform the query into another query that actually accesses the underlying sources

Technology: XML, XLink/XPointer

use standard XML technology

- semantic knowledge/metadata in XML
- connection between metadata and the data itself extends the ideas of XLink
- “overlay” that is put over a relevant Web fragment.
- extends it with information that allows to compose dynamic XPointers

Advantages

- use the original data sources with all their internal optimizations, and
- extend them with a layer that allows for a more comfortable, semantic access, e.g., based on ontologies.

Technology: XLink/XPointer

- XPointer combines URLs with an extension of XPath for addressing fragments of documents:

  url#xpath-expr

- XML Linking Language (XLink)
  - xlink:type
  - xlink:href

```
<!ELEMENT country (. . . capital . . . )>
<!ELEMENT capital EMPTY>
<!ATTLIST capital xlink:type (· · ·)  #FIXED “simple”
                  xlink:href  CDATA #REQUIRED >

<country id=“germany”>
  <capital href=“file:cities.xml#//city[@name=’Berlin’]”/>
  ⋮
</country>
```

Additionally: out-of line links (arcs) as an “overlay” on existing (autonomous) sources.

TECHNOLOGY: XLINK/XPOINTER

- our approach: overlay does not only contain (static) links, but generates pointers based on
  - internal semantic knowledge
  - addresses of sources
  - knowledge about concepts and properties described by sources
    - which source
    - relative query

AGENTS' KNOWLEDGE

- semantically describes the concepts of its application domain
- no data of the application-level, but
- links/pointers to the actual data.

Here: simple generic conceptual model (EER/ODMG/UML)

- defining concepts, properties, and relationships.
- extended by elements (similar to XLink elements) that specify *how* the actual data instances of the semantic notions can be obtained from the Web sources.
  (Global-as-View)

AGENTS' KNOWLEDGE

- identification of sources by global XPointers
- identification of concepts
  - source
  - key (local XPointer)
- applicable properties and relationships
  - name
  - concept/type/class of its result
  - how they can be obtained from the sources
    - name of the reference database
    - a local XPointer how to get the property relative to the given instance

- provides all information how to define an integrated view over these concepts
- global view definition is also in XML

AGENTS' KNOWLEDGE

```
<source name="cia" xlink:type="simple"
        xlink:href="www.mondial/cia.xml"/>
<source name="gs" xlink:type="simple"
        xlink:href="www.mondial/gs.xml"/>
<concept name="country">
    <source name="cia" key="name" query="//country"/>
    <source name="gs" key="name" query="//country"/>
    <property name="name" value="string" card="1 1">
      <source base="cia" query="{$country}/@name"/>
      <source base="gs" query="{$country}/name/text()"/>
    </property>
    <property name="population" value="int" card="1 1">
      <source base="cia" query="{$country}/population"/>
      <constraint name="cpop" check="{$population}>1"/>
    </property>
    <relationship name="has_city" value="city"
                        inverse="in_country" card="1 *">
      <source base="gs" query="{$country}//city">
    </relationship>
    <relationship name="has_capital" value="city" card="1 1"
             subrelationship_of="has_city" inverse="is_capital">
      <source base="cia" query="{$country}.has_city
                                        [name={$country}/@capital]"/>
    </relationship>
```

AGENTS' KNOWLEDGE

```
<concept name=“country”>
    ⋮
    <relationship name=“neighbor” value=“country”
                          symmetric=“yes” card=“0 *”>
      <source base=“cia”
        query=“country[name={$country}/borders/@country]”/>
    </relationship>
</concept>
<concept name=“independent_country”>
    <subclass_of name=“country”
            discriminator=“not isa dependent_area”/>
</concept>
<concept name=“dependent_area”>
    <subclass_of name=“country”
            discriminator=“{$country}.belongs”/>
    <relationship name=“belongs” value=“country” card=“1 1”>
      <source base=“cia” query=“{$country}/dependent”/>
    </relationship>
</concept>
    ⋮
```

Agents' Knowledge

```
<concept name=“city”>
    <source xlink:type=“simple” name=“gs”
                key=“name country”
                xlink:href=“www.mondial/gs.xml#//city”/>
    <property name=“name” value=“string” card=“1 1”>
            <source base=“gs” query=“{$city}/name/text()”/>
    </property>
    <relationship name=“country” value=“country” card=“1 1”>
      <source base=“gs” query=“{$city}/ancestor::country”/>
    </relationship>
    <property name=“population” value=“int” card=“1 1”>
      <source base=“gs” query=“{$city}/population”/>
      <constraint name=“ctypop” check=“{$population}>1”/>
    </property>
      ⋮
    <constraint name=“citycountry”
           assert=“{$city}.in_country = {$city}.country”/>
</concept>
```

AGENTS' REASONING MECHANISM

Internal Data Model

- class hierarchy, nonmonotonic inheritance
- metadata level
- e.g. description logics, or F-Logic

QUERYING

- knowledge base does not give an integration program
- for each object and each of its properties, it is described how it can be obtained.
- The final query is then constructed by putting these subqueries together.

“Select the names of all countries whose capital is a big city”.

- countries: at CIA and at GlobalStatistics
- population: from the CIA source.
- capital:
  - based on the CIA database (name of the capital),
  - then use the semantic property *country.has_city*.
  - requires to use the equivalent *country* from the GS source (by using the *name* key) and to retrieve its *city* subelements.

QUERYING

“Select the names of all countries whose capital is a big city”.

```
FOR  $cia_country IN
          document(www.mondial/cia.xml)//country     (1,4)
LET  $name := $cia_country/@name,                     (7)
     $gs_country := document(www.mondial/gs.xml)
                      //country[name=$name]           (2,5,8)
     $cap := $gs_country//city                  (17,14,7,8)
                           [name/text()=              (36)
                            $cia_country/@capital]    (17)
WHERE $cap/population/text() > 1.000.000              (40)
RETURN {$name}
```

Conclusion and Perspectives

- semantic, integrating level over existing Web information sources, consisting of agents.
- intelligent mediators based on their ontological knowledge and on links that describe where concepts and their properties can be found in the sources.
- approach does not require annotated Web pages.
- agents' knowledge base has an XML representation that allows also to exchange it between agents.
- internal reasoning mechanism

Issues

- intelligent query reformulation

Contents