Ontologies, Semantic Web and Virtual Enterprises

Marek Obitko
mobitko@ra.rockwell.com
Rockwell Automation Research
Center, Prague, Czech Republic
Agenda

• Motivation
• Ontologies
• Semantic Web
• Selected Applications
  – Semantic Search
  – Semantic Integration
  – Semantic Web Services
• Summary
Agenda

- Motivation
- Ontologies
- Semantic Web
- Selected Applications
  - Semantic Search
  - Semantic Integration
  - Semantic Web Services
- Summary
Motivation

• Virtual Enterprise
  – “…is a temporary alliance of enterprises that come together to share skills or core competencies and resources in order to better respond to business opportunities, and whose cooperation is supported by computer networks…“
  – Communication (=knowledge/information exchange and sharing) is necessary
  – We need more from computers than just data exchange
Motivation

• Transportation chain case study (US Army)
  – Involving different companies around the world
  – Transportation of containers weighing tons worked well
  – Information about what is in these containers was not transferred – manual repacking needed!

• Source of the problem: Interfaces between manufacturing companies, airlines, shipping and trucking companies – their systems were not made to work together
Motivation

• Exchanging information
  – Physical layer – message must be transported
  – Syntactical layer – recipient must parse (recognize symbols in) the message
  – **Semantical layer** – recipient must understand the message (i.e., know the meaning of the symbols in the message)
  • Semantics (partially) captured in ontology
Agenda

• Motivation
• Ontologies
• Semantic Web
• Selected Applications
  – Semantic Search
  – Semantic Integration
  – Semantic Web Services
• Summary
Ontology

• Ontology – description of a domain
  – Not changing (or changing rarely)
  – Engineering artifact: classes of objects, properties, relationships, restrictions…
  – what can and what cannot exist; what we can conclude from a state of affairs
  – Example: City, FlightConnection, destination

• Knowledge base – particular state of affairs
  – Example: Prague is destination of FlightConnection OK0103
Ontology

• Formal explicit specification of conceptualization

Intended models (description of the domain, what is possible in it)

Ontology (restriction of the possible models, expressing conceptualization)

Possible models expressible in the ontology language
Formality of Ontology

• Formality/usability

  - Catalog, IDs
  - Thesauri
  - Terms, Glossary
  - Formal is-a
  - Informal is-a
  - Formal instance
  - Frames, properties
  - Value restrictions
  - Disjointness, inverse, part-of
  - General logical constraints

• Description Logics
  - Rich enough for practical applications
  - Computationally tractable
  - Semantic web ontologies, tool support
Description Logics

- Formal description of concepts and roles
  - from semantic networks + frame based systems
- Attributive language $\mathcal{AL}$

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Semantics</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A$</td>
<td>$A^I \subseteq \Delta^I$</td>
<td>atomic concept</td>
</tr>
<tr>
<td>$R$</td>
<td>$R^I \subseteq \Delta^I \times \Delta^I$</td>
<td>atomic role</td>
</tr>
<tr>
<td>$\top$</td>
<td>$\Delta^I$</td>
<td>top (most general) concept</td>
</tr>
<tr>
<td>$\bot$</td>
<td>$\emptyset$</td>
<td>bottom (most specific) concept</td>
</tr>
<tr>
<td>$\neg A$</td>
<td>$\Delta^I \setminus A^I$</td>
<td>atomic negation</td>
</tr>
<tr>
<td>$C \sqcap D$</td>
<td>$C^I \cap D^I$</td>
<td>intersection</td>
</tr>
<tr>
<td>$\forall R.C$</td>
<td>${a \in \Delta^I</td>
<td>\forall b. (a, b) \in R^I \Rightarrow b \in C^I}$</td>
</tr>
<tr>
<td>$\exists R.\top$</td>
<td>${a \in \Delta^I</td>
<td>\exists b. (a, b) \in R^I}$</td>
</tr>
</tbody>
</table>

Person $\sqcap \neg$ Female  
Person $\sqcap \exists$ hasChild. $\top$  
Person $\sqcap \forall$ hasChild.Female
Agenda

• Motivation
• Ontologies
• Semantic Web
• Selected Applications
  – Semantic Search
  – Semantic Integration
  – Semantic Web Services
• Summary
Semantic Web

• Semantic Web
  – “provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries. It is a collaborative effort led by W3C (World Wide Web Consortium)…”

• “Classical” Web: computers deliver documents (text, multimedia…)

• Semantic Web: let computers process (interpret, combine, select, judge) and deliver information
Semantic Web

User interface and applications

Trust

Proof

Unifying logic

Querying: **SPARQL**

Ontologies: **OWL**  Rules: **RIF/SWRL**

Taxonomies: **RDFS**

Data interchange: **RDF**

Syntax: **XML**

Identifiers: **URI**  Character set: **UNICODE**
RDF

- Resource Description Framework (RDF)
  - triples object-predicate-subject

http://xmlns.com/foaf/0.1/Person
http://www.w3.org/1999/02/22-rdf-syntax-ns#type
http://www.w3.org/1999/02/22-rdf-syntax-ns#type
http://xmlns.com/foaf/0.1/homepage
http://xmlns.com/foaf/0.1/mbox
http://xmlns.com/foaf/0.1/givenname
mailto:joe.smith@example.org

http://xmlns.com/foaf/0.1/family_name
Joe
Smith

http://www.example.org/~joe/contact.rdf#joesmith
http://www.example.org/~joe/
http://www.example.org/~joe/contact.rdf#joesmith
http://www.example.org/~joe/contact.rdf#joesmith
RDF Serialization

- **RDF/XML**
  - XML form, standard for exchange between machines
- **N3, TURTLE**
  - More readable (and writeable) by humans

```xml
:joesmith a foaf:Person ;
  foaf:givenname "Joe" ;
  foaf:family_name "Smith" ;
  foaf:homepage
    <http://www.example.org/~joe/> ;
  foaf:mbox
    <mailto:joe.smith@example.org> .
```
RDFS

- RDF Schema (RDFS)
  - Vocabulary for RDF – taxonomies of classes and properties, domain, range, ...

```
:Dog rdfs:subClassOf :Animal.
:Person rdfs:subClassOf :Animal.
:hasChild rdfs:range :Animal;
  rdfs:domain :Animal.
:hasSon rdfs:subPropertyOf :hasChild.
:Max a :Dog.
:Abel a :Person.
:Adam a :Person;
  :hasSon :Abel.
```
OWL

• Web Ontology Language (OWL)
  – Description logic syntactically embedded into RDF(S)
    • OWL Lite – simple constraints, description logic $SHIF$
    • OWL DL – description logic $SHOIN$
    • OWL Full – no restrictions to RDF

• Example

\[
\text{Pizza} \sqsubseteq \exists \text{hasBase. PizzaBase} \\
\text{Pizza} \sqcap \text{PizzaBase} \equiv \bot \\
\text{NonVegetarianPizza} \equiv \text{Pizza} \sqcap \neg \text{VegetarianPizza} \\
\text{Tr}(\text{isIngredientOf}) \\
\text{isIngredientOf} \equiv \text{hasIngredient}^{-}
\]
OWL as RDF graph

http://example.com/pizzas.owl#NonVegetarianPizza
http://www.w3.org/2002/07/owl#equivalentClass

http://example.com/pizzas.owl#VegetarianPizza
http://www.w3.org/2002/07/owl#intersectionOf
  http://www.w3.org/1999/02/22-rdf-syntax-ns#complementOf
  http://www.w3.org/1999/02/22-rdf-syntax-ns#first
http://www.w3.org/1999/02/22-rdf-syntax-ns#rest
http://example.com/pizzas.owl#Pizza
http://www.w3.org/2002/07/owl#disjointWith
http://www.w3.org/2000/01/rdf-schema#subclassOf
http://example.com/pizzas.owl#PizzaBase
http://www.w3.org/2002/07/owl#someValuesFrom
http://example.com/pizzas.owl#hasProperty
http://example.com/pizzas.owl#hasBase
http://www.w3.org/1999/02/22-rdf-syntax-ns#type
http://www.w3.org/2002/07/owl#ObjectProperty
http://www.w3.org/1999/02/22-rdf-syntax-ns#type
http://example.com/pizzas.owl#hasIngredient
http://example.com/pizzas.owl#hasIngredientOf
http://www.w3.org/2002/07/owl#inverseOf
http://www.w3.org/1999/02/22-rdf-syntax-ns#type
http://www.w3.org/2002/07/owl#TransitiveProperty
http://www.w3.org/2002/07/owl#Restriction
OWL Reasoning Examples

- Transportation system - nodes, conveyor belts, ...

:targetNode rdfs:subPropertyOf :connectedTo.
   - “node TN is target node of a conveyor belt CB” entails “TN and CB are connected”

:connectedTo a owl:SymmetricProperty.
   - “X is connected to Y” entails “Y is connected to X”

:targetNode a owl:FunctionalProperty.
   - Commercial dept.: “Node Z1 is target node of conveyor belt BE”
   - Router supplier: “Router R5 is target node of conveyor belt BE”
   - …entails “Node Z1 and Router R5 is the same thing” (can be explicitly stated using owl:sameIndividualAs)

:contains a owl:TransitiveProperty.
   - “A contains B” and “B contains C” entails “A contains C”
SPARQL

• Simple Protocol and RDF Query Language
  – SQL like language for RDF querying
    SELECT ?name ?mbox
    WHERE { ?x foaf:name ?name .
    ?x foaf:mbox ?mbox . }
  – graph matching/construction
  – SELECT, CONSTRUCT, DESCRIBE, ASK
  – ORDER BY, DISTINCT, OFFSET, LIMIT
• Operates on any RDF graph
  – i.e., including RDFS/OWL
Agenda

• Motivation
• Ontologies
• Semantic Web

• Selected Applications
  – Semantic Search
  – Semantic Integration
  – Semantic Web Services

• Summary
Semantic Search

- One of the primary goals of semantic web
  - Not only keyword full-text search
  - Query includes relations between resources
  - Connecting data: mash-up from different sources

- Also needed for search within enterprise or enterprises
  - Relevant research: “semantic desktop” – semantic search within data in a single PC
Web versus Enterprise

• Where to get annotations?
  – Let users make them
    • Extra time needed, may require additional knowledge if annotation needs to be perfect
  – Generate them from data context
    • Data exist in some context that can be used to generate metadata for search (class, relations, …)

• Can we trust data providers?
  – Within enterprise there are not so many attempts to cheat as on the web
Semantic Search: Example

• Industrial domain: Assembly line search
  – Data in many formats – ladder logic in controllers, HMI panels, …
  – Structured data with annotations are stored in RDF form -> following queries can be formulated
    • Find ladder code rungs containing XIO instructions referring tags StartCycle and StopCycle
    • Find all the text objects with background color navy that have “axis" in their caption
Semantic Search: Example

- Find ladder code projects that have a tag used in a Gauge control on any HMI display

```
SELECT ?exp ?hmifile ?ladderfile WHERE {
  [ a file:File;
    file:hasFileFileName ?hmifile;
    gen:contains
      [ a hmi:Gauge.
        gen:contains
          [ hmi:hasTagName ?exp; ] ] ]
  [ a file:File;
    file:hasFileFileName ?ladderfile;
    gen:contains
      [ a ladder:tag;
        ladder:hasName ?exp; ] ] }
```
Semantic Search

• Both in enterprise/desktop search and in semantic web
  – “Enrichment of the current web” versus “Web of Data”
    • First option is more used for popularizing
    • Second option has made more progress and currently has bigger potential
  – Not surprising, because large part of web is generated from databases
Agenda

• Motivation
• Ontologies
• Semantic Web
• Selected Applications
  – Semantic Search
  – Semantic Integration
  – Semantic Web Services
• Summary
Semantic Integration

- Communication between enterprises or even within single enterprise
  - Even when using the same ontology language, ontologies are different
  - There is no one ontology that would satisfy all needs for everyone and forever
- Need to deal with multiple ontologies
Communication

• Possible model of communicating enterprises
  – Multi-agent or holonic system

• Only what is expressed in ontology
  – Can be stored in agent’s knowledge base
  – Can be communicated between agents

• Agents with different ontologies
  – Need of translation of messages between different ontologies during communication

• Example
  – transportation domain, ontologies expressed in OWL
“Berlin” Ontology

- Berlin local transport service
“Boeing” Ontology

- Graph ontology used for modeling infrastructure; further specializing details
Translation

• One needs to know how ontology elements are related mapping between ontologies can be expressed in OWL
  
  \[
  \begin{align*}
  \text{boeing:Place} & \equiv \text{berlin:Address} \\
  \text{boeing:Conduit} & \equiv \text{berlin:Connection} \\
  \text{boeing:connects-to-node} & \equiv \text{berlin:destSrc} \equiv \text{berlin:destination} \sqcup \text{berlin:source}
  \end{align*}
  \]

• This information is used to translate messages – OWL reasoning is used, deductions contain translation

\[
\begin{align*}
\text{boeing:Place}(placeA) \\
\text{boeing:Conduit}(path1) \\
\text{boeing:connects-to-node}(path1, placeA)
\end{align*}
\]

\[
\begin{align*}
\text{berlin:Address}(placeA) \\
\text{berlin:Connection}(path1) \\
\text{berlin:destSrc} & \equiv \text{berlin:destination} \sqcup \text{berlin:source} \\
\text{berlin:destSrc}(path1, placeA)
\end{align*}
\]
Ontology Service

- Helps with translation and other ontological tasks
  - implemented as agent, but similar service can be also embedded into SOA container as a special mediator
Translation – Architecture

• All agents handle translation themselves
  – Agents are aware of ontologies, know all the implications of translation
  – Agents do not rely on anything else

• Specialized agent handles translation
  – Agents need to be aware of translation, but a specialized ontology agent/service handles it
  – Agents can focus on their tasks

• Transparent translation
  – Translation handled directly in multi-agent (or SOA) platform
  – Agents do not have to be aware of ontologies
  – Agent’s preferred ontology needs to be supplied
Agenda

• Motivation
• Ontologies
• Semantic Web
• Selected Applications
  – Semantic Search
  – Semantic Integration
  – Semantic Web Services
• Summary
Services

- Virtual Enterprises, Agents, Services in SOA, …
  - Distributing tasks, finding appropriate services
- Web services ("Web API")
  - Standards for syntactic interoperability:
    - SOAP – protocol for accessing a service
      - Call a service; return result
  - WSDL – description of a service
    - Endpoints (ports), interface (parameters for a message)
  - UDDI – metadata about web services
    - White (contact), yellow (categorization), green (technical information) pages
Semantic Web Services

• Semantic enrichment of Web Services
  – i.e., not replacement

• Semantic interoperability
  – WSDL – available messages, data structures – but not their meaning (semantic constraints)
  – For automated discovery, composition and execution
  – Special ontologies developed for the description of services: OWL-S, WSMO, …
OWL-S

- OWL ontology for describing services
- Top level:
OWL-S Matchmaking

• Description of input and output conditions as a concept
  – In service advertisement and service request
  – E.g., price, provided by, delivery date, …

• Entailment relationship between concepts expressing the description is verified
  – Advertisement versus request (OWL reasoning)
  – Exact, plugin (advertisement is more general), subsume (request is more general), intersection (partially satisfiable), disjoint
Semantic Web Services

• Semantics necessary for realizing SOA vision
  – Situation today
    • Rather tightly coupled (not really loosely coupled) services
    • Programmers design and implement service calls
• Finding, composing and executing services
  – Core of virtual enterprises
  – Not possible in heterogeneous environment without semantic description
Agenda

- Motivation
- Ontologies
- Semantic Web
- Selected Applications
  - Semantic Search
  - Semantic Integration
  - Semantic Web Services
- Summary
Summary

• Ontologies and Semantic Web
  – Essential for achieving truly computer supported VE
  – Expressing meaning (semantics) for computers

• Illustrated on selected applications critical for VE
  – Semantic Search – enrichment of data by generated semantic annotation to perform more precise search
  – Semantic Integration – translation between different ontologies to work in heterogeneous environment
  – Semantic Web Services – service matchmaking to allow service discovery, composition and execution
Some references and further reading (1)

- Motivation

- Ontologies

- Description Logic
Some references and further reading (2)

- **Semantic Web**

- **Semantic Search**
Some references and further reading (3)

- **Semantic Integration**

- **Semantic Web Services**
  - The OWL Services Coalition: OWL-S: Semantic Markup for Web Services, 2004 http://www.daml.org/services/owl-s/1.0/
Thanks for your attention

Questions, comments?