Semantic Web Technologies for Real World Applications

Ulrich Reimer
Institute for Information and Process Management
University of Applied Sciences St. Gallen
Switzerland

Overview

1. Introduction
2. Application Areas for Semantic Web Technology
   2.1 Content-Oriented Retrieval
   2.2 Reference Modeling & Model-Driven Development
   2.3 Information & Process Integration
   2.4 Flexible Business Transactions
3. Opportunities, Barriers, and Future Development
What characterizes a Semantic Web application?

An application employing Semantic Web technology contains a model of the application domain:

- an ontology, i.e. concepts, their properties, relationships, axioms
- rules representing regularities, constraints, norms of the domain
Overview

1. Introduction
2. Application Areas for Semantic Web Technology
   2.1 Content-Oriented Retrieval
   2.2 Reference Modeling & Model-Driven Development
   2.3 Information & Process Integration
   2.4 Flexible Business Transactions
3. Opportunities, Barriers, and Future Development

Time-consuming information search (1)

Many studies show:

A large percentage of daily work deals with information (search, obtain, transform, distribute, file):

→ about 6 weeks/year for administrative workers
→ 60 – 80% for knowledge workers

Recent study of a DMS company:

80% of interviewed companies need a whole day to find relevant information (although the DMS offers a search engine).

Reason:
In many cases there are no guidelines for how to file documents.
Time-consuming information search (2)

Consequences of inefficient and insufficient information provision:
→ higher costs
→ longer process durations
→ longer time to market
→ important information is not found:
  → inferior quality of decisions, processes, etc.
  → compliance violations
  → reinventing the wheel

Information search with search engines

“tanker accident” atlantic

words contained in the documents

On 19 July 1979, the Atlantic Empress and the Aegean Captain collided with each other in the Caribbean Sea, off Tobago island. ...
The tanker accident ...

A single oil tanker accident could destroy the entire coastline. ...

Two tankers passed the coast every week, transporting oil across the Atlantic to the USA. ...

The leaking oil tanker Prestige sinks off Spain’s north-western coast ...

...
Using meta data to describe document contents

Words contained in meta data:

The leaking oil tanker Prestige sinks off Spain’s north-western coast...

Contents:
- tanker accident
- Atlantic Ocean
- oil spill
- environmental pollution

Author: -
Language: English
Dokument type: news text

Different terms in meta data and query formulation

Words contained in meta data:

Contents:
- tanker collision
- Atlantic Ocean
- oil spill
- environmental pollution

Author: -
Language: English
Dokument type: news text
Different vocabulary for indexing and querying

“tanker accident” atlantic

words contained in meta data

Contents:
tanker collision, Gulf of Biscay, oil spill, environmental pollution
Author: -
Language: English
Dokument type: news text

Controlled vocabulary for indexing and querying

“tanker accident” atlantic

Ontology: controlled vocabulary
Using background knowledge to extend query

Automatic query extension

Ontology: background knowledge

(tanker collision OR tanker accident) AND (Atlantic Ocean OR Carribean Sea OR Bermuda Sea OR ...)

Ontology

tanker collision → synonymy → tanker accident

Atlantic Ocean

part-of

Carribean Sea

Gulf of Biscay

Bermuda Sea

Semantics of relations

(tanker collision OR tanker accident) AND (Atlantic Ocean OR Carribean Sea OR Bermuda Sea OR ...)

Ontology: background knowledge
Required background knowledge can be complex

“tanker accident” atlantic

Automatic query extension

(tanker collision OR tanker accident) AND
(Atlantic Ocean OR Carribean Sea OR Bermuda Sea OR ...)

document still not found

The leaking oil tanker Prestige sinks off Spain’s north-western coast ...

Technical realisation

What is needed: RDF

- assign meta data to information objects
- content description with concepts and relations between them
- provision of background knowledge
- provision of the semantics of relations for query extension, ontology integration, etc.

RDF Schema, OWL, Rules
Ontology-based skills management

- Description of skills
- Ontology

Bitte stufen Sie Ihre Kenntnisse/Fähigkeiten zu "Marketing" ein:

<table>
<thead>
<tr>
<th>Stufe 1</th>
<th>Stufe 2</th>
<th>Stufe 3</th>
<th>Stufe 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generelle Kenntnisse:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Betriebswirtschaft:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gewählte Kenntnisse:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Marketing:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Spezielle Kenntnisse:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Marktforschung:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Produktentwicklung:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vertrieb:</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Unterbegriff vorschlagen zu "Marketing":

Senden
Query extension (1)

An ontology itself does not guarantee good query results
→ Semantic query extension:

descend the concept hierarchy:
- described skill: WLAN
- queried skill: LAN

ascend the concept hierarchy:
- described skills: LAN
- queried skills: WLAN

Query extension (2)

include siblings:
- described skill: Linux
- queried skill: BSD Unix

include synonyms:
- described skill: Voice over IP
- queried skill: VoIP

operating system:
- is-a
  - Linux
  - BSD Unix

synonymy:
- VoIP ↔ Voice over IP
Conclusion to ontology-based skills management

- an ontology is absolutely needed
- ontology development requires strong guidance by ontology experts
- keep ontology as small as possible
- query extension very important

Overview

1. Introduction
2. Application Areas for Semantic Web Technology
   2.1 Content-Oriented Retrieval
   2.2 Reference Modeling & Model-Driven Development
   2.3 Information & Process Integration
   2.4 Flexible Business Transactions
3. Opportunities, Barriers, and Future Development
Modelling vs. Programming

**Hard-wired:**

- Project information system
- Contract information system
- etc.

**Generic:**

- Project ontology
- Contracts ontology
- Generic information system
Modelling vs. Programming

Project ontology

- Portfolio contains Milestone
- Project has-part Document
- Sub-project has-part Issue
- Workpackage has-part Client
- Person is-a Status

Generic information system

How to present objects at the user interface?

Tool for Project Management

Welcome Mr. John Q. Public 11/07/2003

Portfolio

- E-Gov-Project
- Process analysis
- SOA
- Web Services
- Registry
- Orchestration
- Ontology
- GUI

Budget

- Management of actual costs for the selected structure element. Use “+” to switch between monthly and yearly views and use “Show detail” to display the sum of the sub elements. New versions of the data can be created and older versions are listed on the left hand side of the costs table. Data appears in the column. Data can be exported to Microsoft Excel.

Forecast

<table>
<thead>
<tr>
<th>Project Description</th>
<th>2004</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Value in TEUR</td>
<td>A</td>
<td>P</td>
<td>A</td>
<td>P</td>
<td>P</td>
<td>A</td>
<td>P</td>
</tr>
<tr>
<td>Internal Personnals (Projects)</td>
<td>2,01</td>
<td>0,18</td>
<td>5,02</td>
<td>0,20</td>
<td>9,05</td>
<td>0,20</td>
<td>20,03</td>
</tr>
<tr>
<td>Internal Personnals (Fbt. Genauigung)</td>
<td>2,01</td>
<td>0,18</td>
<td>5,02</td>
<td>0,20</td>
<td>9,05</td>
<td>0,20</td>
<td>20,03</td>
</tr>
<tr>
<td>Internal Personnals (Fbt. Genauigung)</td>
<td>2,01</td>
<td>0,18</td>
<td>5,02</td>
<td>0,20</td>
<td>9,05</td>
<td>0,20</td>
<td>20,03</td>
</tr>
<tr>
<td>Project Budget</td>
<td>20,03</td>
<td>1,00</td>
<td>40,06</td>
<td>0,08</td>
<td>60,00</td>
<td>0,08</td>
<td>90,00</td>
</tr>
<tr>
<td>Sales Budget</td>
<td>20,03</td>
<td>1,00</td>
<td>40,06</td>
<td>0,08</td>
<td>60,00</td>
<td>0,08</td>
<td>90,00</td>
</tr>
<tr>
<td>Total Budget</td>
<td>20,03</td>
<td>1,00</td>
<td>40,06</td>
<td>0,08</td>
<td>60,00</td>
<td>0,08</td>
<td>90,00</td>
</tr>
</tbody>
</table>

Save
Edit
Cancel
How to present objects at the user interface?

- **Navigation**: How to navigate through the user interface.
- **Functions**: What functions are available.

Presenting subprojects and workpackages differently

- **Sub-projects** vs. **Workpackages**: Different presentation styles for subprojects and workpackages.
Presenting subprojects and workpackages differently

Mapping GUI objects to information objects
Presenting subprojects and workpackages differently

**Portfolio**
- E-Gov-Project
- Process analysis
- SOA
- Web Services
- Registry
- Orchestration
- Ontology
- GUI

**VERSUS**

**Project: E-Gov-Project**

<table>
<thead>
<tr>
<th>Status</th>
<th>Report Date</th>
<th>Responsible</th>
<th>Start</th>
<th>End</th>
<th>Budget</th>
</tr>
</thead>
<tbody>
<tr>
<td>April 06</td>
<td>Johnny B. Good</td>
<td>1 March 2006</td>
<td></td>
<td></td>
<td>8 kEuro</td>
</tr>
</tbody>
</table>

**Workpackages:**
- Web Services
- Registry
- Orchestration

**Mapping GUI objects to information objects**

- Portfolio
- Milestone
- Document
- Issue
- Client
- Status
- Person
- Responsible

- size
- Tab sequence
- position
- GUI objects

- “Process analysis”
- “SOA”

- instance-of
- has-part
- contains
- has-part
- instance-of
- instance-of
Navigation hierarchy from relations between concepts

Relationships in ontology:

- Project
  - Milestone
  - Document
  - Issue
  - Client

Navigation hierarchy:

Project 1
- Milestone
  + Milestone 1
  + Milestone 2
+ Document
+ Issue
+ Client

instance
Navigation hierarchy from instances of related concepts

Relationships in ontology:

- Project
  - Sub-project
    - Workpackage

Navigation hierarchy:

- Project 1
  - Sub-project 1
    - Workpackage 11
    - Workpackage 12
  - Sub-project 2
  - Sub-project 3
  - Sub-project 4

instance

Relationships in ontology:

- Project
  - Sub-project
    - Workpackage

Navigation hierarchy:

- Project 1
  - Sub-project 1
    - Workpackage 11
    - Workpackage 12
  - Sub-project 2
  - Sub-project 3
  - Sub-project 4
Navigation hierarchy from attribute values: classification

Attribute in ontology:

Project → delay: { yes, no }

Navigation hierarchy:

Project

+→ Projects: delayed

+→ Projects: in time

concept
Navigation hierarchy from type of related instance

Relationships in ontology:
- Project → Client
  - Bank → Client
  - Insurance → Client

Navigation hierarchy:
- Project
  - Client: Bank
  - Client: Insurance

Navigation hierarchy from attribute values

Relationships in ontology:
- Project → Client
  - Bank → Client
  - Insurance → Client

Navigation hierarchy:
- Project
  - Client: Bank
  - Client: Insurance
Many aspects to model

Some aspects are declarative (i.e. model elements),
some are procedural (i.e. pieces of code).

Reference modelling
Aggregation of sub-models: Holistic vs. modular

Holistic Project Information System

- domain ontology
- rules
- workflows
- functions
- roles & rights

Modular Project Information System

- Project Info System: Basic
  - applicational ontologies
  - GUI ontology

- Project Info System: Milestones
  - applicational ontologies
  - GUI ontology

- Project Info System: Budget
  - applicational ontologies
  - GUI ontology

Reference modelling: Specialization

Reference Project Model
- Project ontology
- Generic project information system

Reference Controlling Model
- Controlling ontology
- Generic controlling information system

Reference Automotive Project Model
- Project ontology for Automotive
- Project information system for Automotive

Reference Automotive Controlling Model
- Controlling ontology for Automotive
- Controlling information system for Automotive
Reference modelling: Specialization and Aggregation

Reference Project Model
- Project ontology
  - Generic project information system

Reference Controlling Model
- Controlling ontology
  - Generic controlling information system

Reference Automotive Project Model
- Project ontology for Automotive
  - Project information system for Automotive

Reference Automotive Controlling Model
- Controlling ontology for Automotive
  - Controlling information system for Automotive

Reference Project Model
- Project ontology for Acme Inc.
  - Project information system for Acme Inc.

Reference Controlling Model
- Controlling ontology for Acme Inc.
  - Controlling information system for Acme Inc.
Different concepts for same/similar things hamper integration

Basic Project Ontology

Ontology for Resource Management

Extension of sub-ontologies destroys independence

Basic Project Ontology

Ontology for Resource Management
Instead relate sub-ontologies to an upper-level-ontology

Upper-level ontology facilitates aggregation of sub-models
Model-driven development: From models to code

Configuration workbench

Reference Project Model

Project ontology for Automotive

Generic information system

Specialization

Project ontology for Acme Inc.

Generic information system

Compilation

Run-time system

Project information system for Acme Inc.

Overview

1. Introduction

2. Application Areas for Semantic Web Technology
   2.1 Content-Oriented Retrieval
   2.2 Reference Modeling & Model-Driven Development
   2.3 Information & Process Integration
   2.4 Flexible Business Transactions

3. Opportunities, Barriers, and Future Development
Revenue tax declaration: Chaotic process with many media breaks

Process & information integration

Trustee
Company
A-Bank
B-Bank
Tax authority 1
Tax authority 2
Different process for each province

Reference process model for revenue tax declaration
Reference process model plus specializations of it

Reference Model: Revenue tax declaration

Company

Trustee

tax documents

A-Bank

B-Bank

Specialized model for Province 1

Tax authority 1

Specialized model for Province 2

Tax authority 2

Different vocabulary for each province

Trustee

Company

tax documents

A-Bank

B-Bank

Ontology 1

Tax authority 2

Ontology 2
Reference ontology with mappings to local ontologies

- Trustee
- Company
- A-Bank
- B-Bank
- Reference Ontology
- Ontology 1
- Tax authority 1
- Tax authority 2
- Ontario 2

Process participants can get different views

- Reference Model: Revenue tax declaration
  - Ontology
  - Process descriptions
  - Law & regulations

- Specialized model for Province 1
  - Ontology
  - Process descriptions
  - Law & regulations

- Specialized model for Province 2
  - Ontology
  - Process descriptions
  - Law & regulations
How to get from models to IT systems?

Reference model with all specializations

Ontology  Process descriptions  Law & regulations

Ontology  Ontology  Ontology  Ontology

Process descriptions

Law & regulations

Model-driven development: From the model to SOA

Reference model with all specializations

Ontology  Process descriptions  Law & regulations

Process control

electronic tax documents

Orchestrated Web Services (BPEL)

Orchestration Engine

Rule Engine

Service-oriented architecture

Company  Trustee  Bank  Tax authority

Web Services
Model-driven development: The ideal case

Business process view:
• application-driven

Web Services / SOA:
• legacy independence
• reusability
• integration

Top-down:
From process model to orchestrated web service
Model-driven development: The ideal case

Business process view:
• application-driven

Web Services / SOA:
• legacy independence
• reusability
• integration

Top-down:
From process model to orchestrated web service

Model-driven development: The reality

Business process view:
• application-driven

Web Services / SOA:
• legacy independence
• reusability
• integration

semantic gap
Model-driven development: The reality

Business process view:
• application-driven

It does not fit!

Web Services / SOA:
• legacy independence
• reusability
• integration

Model-driven development: The solution

Business process view:
• application-driven

Top-down:
From the model to orchestrated web service

Bottom-up:
Build the model from the existing web services
Model-driven development: The solution

Business process view:
- application-driven

Top-down:
From the model to orchestrated web service

Bottom-up:
Build the model from the existing web services

Web Services / SOA:
- legacy independence
- reusability
- integration

Overview

1. Introduction
2. Application Areas for Semantic Web Technology
   2.1 Content-Oriented Retrieval
   2.2 Reference Modeling & Model-Driven Development
   2.3 Information & Process Integration
   2.4 Flexible Business Transactions
3. Opportunities, Barriers, and Future Development
Travel industry: Emergence of new business models

Current business models:
- manual bundling
- pre-fabricated travel packages

Advantages of travel packages:
- higher margins
- simple business processes

Disadvantages:
- fixed itineraries
- inflexible dates
- limited options

Dynamic Packaging: New business model

Advantages:
- single point of contact
- satisfaction of individual customer needs
- increased revenues by revenue management

New distribution model needed

Current distribution model impedes dynamic packaging

Traditional Distribution Model

Travel Supplier / Inventory Owner → Tour Operator / Wholesaler → Retailer / Sales Order Processing → Consumer

New distribution model: Combine information from distributed sources

Electronic Demand-driven Marketplace

Travel Supplier / Inventory Owner → Retailer → Consumer

Travel Supplier / Inventory Owner
Electronic marketplace needs Semantic Web Services

Requirements for dynamic packaging:

• reach across multiple markets, countries, currencies, suppliers
• automatic comparison of products
• automatic aggregation of products
• automatic negotiation of discounts
• trustworthy transactions

→ With current technology not efficient and not profitable!

Dynamic information gathering and transactions needed

conventional registries require programming

airlines hotels railway car rentals
Dynamic information gathering and transactions needed

machine understandable service descriptions

airlines  hotels  railway  car rentals

Input:  <place>
Output:  bag-of( url( <car-rentals>) )

Precondition:
\neg \exists r: ( \text{known( car-rental( r ) \land place-of( r ) = <place> )} \\
\land \neg \text{known(\neg \exists r: car-rental( r ) \land place-of( r ) = <place> )} 

Postcondition:
\exists r: ( \text{known( car-rental( r ) \land place-of( r ) = <place> )} \\
\lor \text{known(\neg \exists r: car-rental( r ) \land place-of( r ) = <place> )} 

Ontology

Semantic Web Services:
Ontology-based service description
Business logic needed for negotiation and aggregation

Electronic Demand-driven Marketplace

Sales Order Processing

Travel Supplier / Inventory Owner

business logic as rule markup

The full potential of the Semantic Web

Dynamic

Web Services
UDDI, WSDL, SOAP

Semantic Web Services
OWL-S, WSMF, etc.

Static

WWW
URI, HTML, HTTP

Semantic Web
RDF Schema, OWL

Texts, images,
Web Service descriptions

Semantic
Overview

1. Introduction

2. Application Areas for Semantic Web Technology
   2.1 Content-Oriented Retrieval
   2.2 Reference Modeling & Model-Driven Development
   2.3 Information & Process Integration
   2.4 Flexible Business Transactions

3. Opportunities, Barriers, and Future Development

Semantic Web Technologies: Opportunities

- Alleviation of information overload
- Querying facts instead of searching texts
- Integration of heterogeneous information sources
- Process integration
- Dynamic formation of business transactions
- Model-driven development of information systems
- Specialization and aggregation of reference models
Semantic Web Technologies: Barriers

• Where do the ontologies come from?
  But: Many ontologies already exist

• Where do the semantic content descriptions come from?
  Ontology development is expensive.

• Not many people have the expertise to build ontologies.

• How to achieve good cost/benefit ratio?

Many ontologies or predecessors already exist

UMLS (Unified Medical Language System):
  • semantically related medical terms
  • terms from ca. 100 heterogeneous classification systems and medical terminologies in 15 languages

UNSPSC (United Nations Standard Products and Services Code):
  • hierarchical classification system for products and services

Geneontology:
  • controlled vocabulary to describe genes and proteins
  • 19’000 concepts
  • is-a and part-of relationships
  and many more

→ Ontologies do not necessarily need to be big!
Conclusion

- Semantic Web → Semantic Web technology

- Modeling will gain in importance over programming.

- Models will be ubiquitous because
  - they are easier to create, modify, maintain
  - they are easier to exchange and adapt (specialized, aggregated)
  - versions can be better maintained (changes propagated top-down)

- Practical applications mostly need lightweight semantics, whereas most research is about heavy-weight semantics.