Outline

• The Semantic Web – vision and purpose
• Technology – building blocks
• Application areas
• Future
What is the Semantic Web?

• “The 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 with participation from a large number of researchers and industrial partners. It is based on the Resource Description Framework (RDF).” ([http://www.w3.org/2001/sw/](http://www.w3.org/2001/sw/))

• "The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation"
  
  *Tim Berners-Lee*
The Semantic Web is about

• common formats for integration and combination of data drawn from diverse sources
  – the Web: main focus on the interchange of documents
• language for recording how the data relates to real world objects
  – connects databases through “being about the same thing”
The Web lacks semantics

- Web pages are designed for presentation
  - visual effects, fast browsing and navigation
  - the meaning of the content is hidden in the text
- Search => information overload
  - keyword search results in more hits than needed
  - find irrelevant information, ignore relevant information
- Can’t use the Web to get answers to questions
  - the user has to interpret the content, solve ambiguities, and integrate information from several sources to find an answer
- Not machine processable
  - need to use ad-hoc mechanisms to make Web content generally available for processing
Structure of Web today

- Resources (Web pages) having URI, all of the same type
- Links do not give information about the type
- The user imbeds all meaning content in the structure
- The computer only sees a simple graph
What is Semantics?

• Semantics is the study of *meaning content*, particularly in languages, but also in logics, mathematics, and business

• The *semiotic triangle*:

  ![Semiotic Triangle Diagram](http://www.jfsowa.com/ontology/ontometa.htm)

• Some semantic complications
  – Synonyms: several expressions have the same reference ("car" and "automobile", "fast" and "quick")
  – Homonyms: same expression have several references ("lie", "fair")

*Figure source: http://www.jfsowa.com/ontology/ontometa.htm*
Ontologies and metadata

• Ontology (philosophy)
  – The study of what is (can be)
  – What types of things exist, relationships

• Ontology (semantic web and AI)
  – "A specification of a conceptualization" (Gruber)
  – Conceptual models: classes, properties, relations.
  – Knowledge representation

• Metadata
  – "Data about data"
  – Describes content, format of a data record or information resource, e.g., about content, origin, quality, category, etc.
  – Can be embedded within the information resource (often the case with web resources) or it can be held separately in a database
Semantic Web

• Resources
  – Concepts of specified type
  – Global unique identity (URI)

• Links
  – Also of specified type
  – Global unique identity (URI)

• Rich semantic model that can be used as metadata for ordinary Web content.
Outline

• The Semantic Web – vision and purpose
• **Technology** – building blocks
• Application areas
• Future
The Semantic Web Layer Cake

- Analysis, verification, trust
  - analyse for consistency, make inferences
  - Logic, Proof, Trust
- Properties and relationships
  - describe/represent knowledge
  - RDF, RDFS, Ontology (OWL)
- The base
  - syntax, structure, identification
  - XML, XML Schema
  - URI

Image source: http://www.w3.org/2007/03/layerCake.png
The Layers of the Cake...

- **Trust** – authentication, trustworthiness of statements
- **Logic and Proof** – establish truth of statements, infer unstated facts
- **Ontology** – vocabularies, shared meaning
- **RDF Schema** – RDF resource types
- **RDF** – metadata, "say anything about anything"
- **XML Schema** – datatypes and structure
- **XML** – common syntax
- **URI** – for identifying things
URI – Universal Resource Identifier

• Allows anybody create (meta)data on any resource on the Web
  – e.g., same XHTML file annotated through different terms
  – semantics added to existing Web resources via URIs
• URIs make it possible to link (via properties) data with one another
  – makes the Semantic Web a Semantic Web!
• Provide a syntax for naming and ground RDF into the Web
  – information can be retrieved using existing tools
• What makes the merging in data integration possible
XML & XML Schema

• XML from presentation (HTML) to structure

• Purposes for using XML
  – serialization syntax for other markup languages
  – separate form from content (e.g. use XSLT)
  – uniform data exchange format

• XML Schema & DTD: define grammar for XML documents
RDF – Resource Description Framework

- Formal model for metadata that describe Web resources in machine-readable form
- Standardize definition and use of metadata descriptions of web based resources
  - also for representing data
- Provides
  - a way of describing resources via metadata (data about data)
  - interoperability between applications that exchange machine understandable information on the Web.
    - XML as interchange syntax
  - together with RDFS: a **lightweight** ontology system
RDF cont.

• Basic concepts
  – Subject: the **resource** being described (URI)
  – Predicate: a **property/attribute** of the resource (URI)
  – Object: the **value** of the property (URI)
  – A combination of these is a **statement/assertion**

• A resource description consists of
  – one or more assertions, where each assertion is a triple *(subject,predicate,object)*:
    
    `<resource, attribute, value>`

    – value can be another resource or a literal (text/number)

• XML for expressing RDF assertions (RDF/XML)
• Literals can be typed (integer, string,...)
  – Ex: “John Doe”^^<http://www.w3.org/2001/XMLSchema#string>
RDF example

('http://.../person/id1234', 'hasName', 'John Doe')

('http://.../person/id1234', 'authorOf', 'http://.../papers/paperID45')

('http://papers/paperID45', 'hasTitle', 'The Title')
RDF example
- in XML serialization

```xml
<rdf:Description rdf:about="http://.../person/id1234">
  <hasName rdf:datatype="http://www.w3.org/2001/XMLSchema#string">John Doe</hasName>
</rdf:Description>

<rdf:Description rdf:about="http://.../person/id1234">
  <authorOf rdf:resource="http://.../papers/paperID45"/>
</rdf:Description>

<rdf:Description rdf:about="http://.../papers/paperID45">
  <hasTitle rdf:datatype="http://www.w3.org/2001/XMLSchema#string">The Title</hasTitle>
</rdf:Description>
```
Structured values

- describe aggregates (e.g. address as a structure with street address, city, postal code, country, ..)
- can be represented as
  - a resource (URI) that we can make statements about
  - a blank node

('http://.../depid/123', address, 'http://.../addrid/345')
('http://.../addrid/345', streetAdr, "Gaustadalleen 23")
('http://.../addrid/345', city, "Oslo")
('http://.../addrid/345', country, "Norway")
('http://.../addrid/345', postCode, "N-0371")
Blank nodes

• represent something that does not have URI ref, but can be described through other information
  – can be said to represent an “existential” statement (“there is a resource such that…”)
  – require attention when merging (identical nodeIDs in different graphs...)
  – blank node identifiers in triples: ‘:_:name’

• alternatives for serialization:
  – create URI, visible from outside
  – create internal nodeID, but not real URI, visible from outside
  – let system create ”nodeID” (when name not important)
Other RDF Capabilities

• RDF Containers
  – vocabulary for representing groups of resources or literals
  – rdf:Bag, rdf:Seq, rdf:Alt

• RDF Collections
  – a group of things represented as a list structure
  – vocabulary for describing groups containing only specified members
  – rdf:List, rdf:first, rdf:rest, rdf:nil

• RDF Reification
  – vocabulary for describing RDF statements
  – rdf:Statement, rdf:subject, rdf:predicate, rdf:object
  – Ex: ‘xyz’ says that ‘The Author of http://.../index.html is John Doe’

• XML Literals
  – values that are XML fragments (or text containing XML markup)
Example: RDF Collection

```xml
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:s="http://example.org/students/vocab#">
  <rdf:Description rdf:about="http://example.org/courses/6.001">
    <s:students rdf:parseType="Collection">
      <rdf:Description rdf:about="http://example.org/students/Amy"/>
      <rdf:Description rdf:about="http://example.org/students/Mohamed"/>
      <rdf:Description rdf:about="http://example.org/students/Johann"/>
    </s:students>
  </rdf:Description>
</rdf:RDF>
```

Note: Example is taken from RDF Primer: http://www.w3.org/TR/rdf-primer/

- **rdf:ParseType** indicates that contents of element should be interpreted in a special way
  - "Collection", "Resource", "Literal", ...
RDF is a flexible and extendable formalism

- From simple triples:
- we can build complex resource descriptions

Figure source: http://ilrt.org/discovery/2001/04/annotations/images/annotation.png

INF5100 Autumn 2007 23
Triple stores

- repositories that can store RDF content
- varying capabilities and focus
  - different focus:
    - providing rich means to reason over triples (e.g., Jena)
    - storing large quantities of data (e.g., threestore, semantic technology - oracle)
  - different operation/function:
    - plugins to current Web browsers (e.g., piggy-bank)
    - systems that can operate with a range of existing third-party DBs (e.g., openrdf.org)
Limitations of RDF

• RDF properties represent relationships between resources
• RDF provides no mechanisms for describing
  – the properties used for describing resources
  – relationships between these properties and other resources

=> RDF Schema (RDF vocabulary description language)
RDFS – RDF Schema

- Defines classes and properties for describing classes, properties and other resources
  - `rdfs:Class`, `rdfs:Property`,...
  - `rdfs:subClassOf`, `rdfs:domain`, `rdfs:range`,...

- RDFS mechanism provides a basic type system for RDF models

- Allow developers to
  - define a particular vocabulary for RDF data (e.g., 'hasName')
  - specify the kinds of objects to which these attributes can be applied
RDFS example

• RDFS type system use predefined terms
  – e.g., Class, subPropertyOf, subClassOf,...
• RDFS expressions are valid RDF expressions

```xml
<rdfs:Class rdf:about="Paper"/>
<rdfs:Class rdf:about="ScientificPaper">
    <rdfs:subClassOf rdf:resource="#Paper"/>
</rdfs:Class>

<ScientificPaper rdf:resource="http://.../papers/paperID45"/>

<rdfs:Property rdf:about="hasTitle">
    <rdfs:domain rdf:resource="#Paper"/>
</rdfs:Property>
```
RDFS vs XML Schema

• XML Schema & DTDs: prescribe order and combination of tags in an XML document

• RDFS: only provides information about the interpretation of the statements given in an RDF data model
  – does not constrain the syntactical appearance on an RDF description
Ontologies – Basis for semantic systems

• Shallow (horizontal) ontologies
  – Generic and reusable, few terms that organize large amounts of data
  – E.g., customer, account number, persons, places

• Deep (vertical) ontologies
  – Domain specific, scientifically oriented
Ontologies (OWL)

- RDF expresses metadata instances, but not the concepts the instances represents
- RDFS does not solve all possible requirements
- Complex applications may need more possibilities
  - term similarity and/or differences (properties or classes)
  - construct classes (not just naming)
  - reasoning
  - ...
- => Ontology Web Language (OWL)
OWL – Web Ontology Language

• OWL is a language to define ontologies
• Main concepts in OWL
  – Classes/subclasses (taxonomies)
  – Properties (inheritance of properties)
  – Datatypes (legal values of properties)
• OWL is based on logics (DL - Description Logic) and opens for reasoning over semantic contents
• OWL allow combining ontologies in an application
Three layers of OWL

• OWL Full
  – no constraints on constructs
  – a real superset of RDFS
  – may be undecidable

• OWL DL (restricts Full)
  – classes and individuals strictly separated
    • a class cannot be an individual of another class
  – no characterization of datatype properties

• OWL Lite (restricts DL)
  – all of DL’s restrictions, plus some more:
    • class construction can be done only through intersection or property constraints
    • cardinality restriction with 0 and 1 only
  – provide a minimal useful subset, easily implemented
Simple example in OWL: Organisation ontology

<?xml version="1.0"?>
<!DOCTYPE owl[
  <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#">
  <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">
  <!ENTITY owl "http://www.w3.org/2002/07/owl#">
  <!ENTITY org "http://www.ifi.uio.no/dmms/organisation#">
]>  
<rdf:RDF  
xmlns:rdf = "&rdf;"  
xmlns:rdfs = "&rdfs;"  
xmlns:owl = "&owl;"  
xmlns:xsd = "&xsd;"  
xmlns:org = "&org;"  
xml:base = "http://www.ifi.uio.no/dmms/organisation#">  
<owl:Ontology rdf:about="&org;">  
  <rdfs:comment> Organisations </rdfs:comment>  
  <rdfs:label>Organisation Ontology</rdfs:label>  
</owl:Ontology>  
<owl:Class rdf:ID="Organisation"/>  
<owl:Class rdf:ID="Hospital">  
  <rdfs:subClassOf rdf:resource="#Organisation"/>  
</owl:Class>  
<owl:Class rdf:ID="OrgNumber"/>  
<owl:DatatypeProperty rdf:ID="address">  
  <rdfs:domain rdf:resource="#Organisation"/>  
  <rdfs:range rdf:resource="&xsd;string"/>  
</owl:DatatypeProperty>  
<owl:DatatypeProperty rdf:ID="telephone">  
  <rdfs:domain rdf:resource="#Organisation"/>  
  <rdfs:range rdf:resource="&xsd;string"/>  
</owl:DatatypeProperty>  
<owl:ObjectProperty rdf:ID="orgNumber">  
  <rdfs:domain rdf:resource="#Organisation"/>  
  <rdfs:range rdf:resource="#OrgNumber"/>  
</owl:ObjectProperty>  
</owl:RDF>

Validated to OWL Lite using  
WonderWeb OWL Ontology Validator  
( http://www.mygrid.org.uk/OWL/Validator )
OWL Classes

• Can construct classes from existing ones
  – enumerate its content
  – through intersection, union, complement
  – through property restrictions

...<owl:Class>
  <owl:unionOf rdf:parseType="Collection">
    <owl:Class rdf:about="#UserProfile"/>
    <owl:Class rdf:about="#DeviceProfile"/>
    <owl:Class rdf:about="#InformationProfile"/>
    <owl:Class rdf:about="#RescueScenarioProfile"/>
  </owl:unionOf>
</owl:Class>
...


OWL Property Restrictions

• (Sub) classes created by restricting the property values on that class

• Value constraints (i.e., on the range)
  – all/some values must be from a class
    • owl:allValuesFrom, owl:someValuesFrom,
  – specific value
    • owl:hasValue

• Cardinality constraints
  – how many times the property is used on an instance
  – minimum, maximum, exact
    • owl:minCardinality, owl:maxCardinality, owl:Cardinality
OWL Property Restrictions Example

```xml
<owl:Class rdf:ID="TeamMember">
  <rdfs:subClassOf rdf:resource="#RescueOperationRole"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#reportsTo" />
      <owl:allValuesFrom rdf:resource="#TeamLeader" />
    </owl:Restriction>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#memberOf" />
      <owl:minCardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:minCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

- A team member always reports to a team leader
- A team member is always member of a team
SPARQL– Information Retrieval

• Query language – ”SQL for RDF”
  – independent of programming language
  – complex querying into RDF data
• Finds RDF triples that match search pattern:
  – graph patterns contains unbounded symbols
  – subgraphs are selected by binding the symbols
  – if such a selection exists, the query returns the bound resources
• Can use pattern constraints (filter), limit number of returned results, remove duplicates, sort results,...
SPARQL Example

SELECT ?isbn ?price ?currency
    FILTER(?currency == €) }

• Query Returns: [ [http://.../..49x,50,€], [http://.../..6682,60,€] ]

SPARQL usage in practice

• locally
  – bound to programming environment (e.g. JENA)
  – less and less typical

• remotely
  – over network
  – separate documents define protocol and result format
    • SPARQL protocol for RDF with HTTP and SOAP bindings
    • SPARQL results in XML format
  – big datasets often offer ”SPARQL endpoints” for this protocol
Bridging RDBs and RDF

- Difficult to export/map all data stored in RDBs into RDF
- Bridges are being defined
  - layer between RDF and the DB
  - RDB tables "mapped" to RDF graphs
  - SQL bridges
- SPARQL endpoints defined to query the data.

RDF bootstrap problem

• Semantic web vision: web of machine understandable documents and data.
  – need ontological support
  – need availability of annotations within documents

• Applications need semantically tagged data to be useful
  – where will the data come from?
  – today: few contain semantic tags a priori
  – => circular dependency

• GRDDL possible solution...
GRDDL – Gleaning Resource Descriptions from Dialects of Languages

• Mechanisms for bootstrapping RDF content from uniform XML dialects
  – extract RDF from XML and XHTML documents
    • e.g., RDFa, microformats, embedded RDF,...

• Associate transformations with document
  – typical XSLT
  – inclusion of references (direct)
  – profile documents (indirectly)
GRDDL Example – extracting RDF from XHTML document

```
<html xmlns="http://www.w3.org/1999/"
     profile="http://www.w3.org/2003/g/data-view">
    <title>Some Document</title>
    <link rel="transformation" href="http:.../dc-extract.xsl"/>
    <meta name="DC.Subject" content="Some Subject"/>
    ...
</head>
...
<span class="date">2006-01-02</span>
...
</html>

<rdf:Description rdf:about="..">
    <dc:subject>Some Subject</dc:subject>
    <dc:date>2006-01-02</dc:date>
</rdf:Description>
```

RIF – Rule Interchange Format

• W3G working group - work started 2005 (www.w3c.org/2005/rules)
• Support and interoperate across a variety of rule-based formats
  – e.g., SWRL, RuleML
• Will address several rule-based formalisms
  – Horn-clause logics, higher-order logics, production systems, etc.
Available tools

• Listed on W3C’s Wiki
  (http://esw.w3.org/topic/SemanticWebTools)
• RDF programming environment
  – for many languages, e.g., C, C++, Python, Java, Javascript, PHP,...
• Triple Stores
• SPARQL ”endpoints”
• RDF converters (to & from)
• validators for RDF, OWL,...
Available technologies for data integration

Outline

• The Semantic Web – vision and purpose
• Technology – building blocks
• Application areas
• Future
Semantic Web – Areas of Use

• Semantic interoperability
  – integration through common conceptualization (ontologies)
  – data integration

• Knowledge Management
  – explicit meaning, categorisation, more precise searches

• Intelligent systems
  – decision support, automatization, agents
Integration between computer systems

• EAI - Enterprise Application Integration
  – Communication through central bus/hub
• SOA - Service Oriented Architecture
  – Applications composed of services delivered via standard interfaces
• WS - Web Services
  – SOA over Internet/Web (SOAP, WSDL, UDDI)
• But none of these solve the main problem: *Differences in semantics* between sender and receiver
Semantic mapping between applications

- Establish common ontology (OWL)
- Develop mapping of application data to a common ontology format
- Communicate at run-time the mapping to/from the common format
Knowledge Management (KM)

- Processes to create, register, communicate, use, and re-create knowledge to contribute to better achievement and higher value creation

"Deliver the right knowledge to the right person, in the right place, at the right time, and in the right context to support the work of that person"
How can KM be supported by semantics

• Constraints on KM technology of today:
  – Information Search: Keyword based search engines
  – Information Extraction: Need humans to browse, find, interpret and combine information
  – Information Maintenance:
    • Inconsistence in terminologies, outdated information
    • Visualize information
    • Difficult to define views over Web information

• Improvements adding semantics:
  – knowledge organised in concept space in relation to content meaning
  – automated tools for knowledge maintenance and discovery
  – semantic answering of questions across multiple knowledge sources
  – manage “who-knows-what” in social networks
  – ...
Automatic annotation

- Annotation – labeling the semantic content of resources (documents, images,...)
- Manual annotation does not scale – the key is automatic generation of metadata
- E.g., Kim Platform for Knowledge & Information Management.

Figure source: http://www.ontotext.com/kim/KIMPlatform.pdf
Example application of Semantic Web in Norway
NRK

• Digital Music Archive (DMA)
  – enormous amounts of metadata on music tracks

• Purpose
  – integration and alignment of archives, easy access
  – improved search, information discovery

• RDBM storing objects, properties and relations

• Semantic Web based publication layer

• In production summer 2007

Figure source: http://www.w3.org/2001/sw/sweo/public/UseCases/NRK/
Example application of Semantic Web in Norway

AKSIO

- Knowledge management in the oil and gas domain
- Transfer of inter-project experience both in planning and operation of drilling process
- Provide decision makers in drilling processes with the best available knowledge in a task-relevant, timely, and contextual manner

Figure source: http://www.w3.org/2001/sw/sweo/public/UseCases/Aksio/
Outline

• The Semantic Web – vision and purpose
• Technology – building blocks
• Application areas
• Future
Semantic social networks

• Social networks
  – Who knows/talks to whom
  – Formal and informal

• "Social software"
  – Synchronous/asynchronous cooperation
  – Personal networks – LinkedIn, ..
  – Personal publishing – Blogging, Wiki, ..
  – Feedback and reputation – Amazon, ..

• Convergence between social networks, sosial software, and semantic technology
  – E.g., FoaF (Friend-of-a-Friend) – machine readable (RDF) information about persons that want contact or can be contacted, ontology based
"Folksonomies"

- **Web-scale tagging**
  - information tagged by people that use it (interested in it)
  - Tag: a word/label to describe something

- **Represents a structure that emerges organically when individuals organize their own information requirements**

- **"Social software", Web 2.0**
  - social networking sites, folksonomies, wikis
  - facilitate collaboration and sharing
  - e.g., Flickr (flickr.com): photographs, del.icio.us: sharing bookmarks

- **Alternative to "shallow" ontologies?**
  - not always
  - lightweight ontologies used when needed, e.g., Friend-of-a-Friend (appl:Flickr)
"Folksonomies” different purpose from ontologies

Ontologies:
- defined through
  - careful explicit process
  - attempting to resolve ambiguity
- define by URI (on Semantic Web)
- inferential process:
  - logic based
  - uses operations such as 'join'
- allow mappings and interactions between data held in different formats

"Folksonomies”:
- definition of tag is
  - loose and implicit process where
  - ambiguity may remain
- define using words
- Inferential process:
  - statistical
  - employ techniques such as clustering
- ?
Semantic Web vision – Personal agents

• Agents are autonomous and proactive
  – object oriented technology and AI combined in the agent model

• A personal agent in the Semantic Web:
  – Receives tasks and preferences from the user
  – Search for information from Web resources, incl. communication with other agents
  – Compares retrieved information with the user’s requirements, and make decisions
  – Delivers answer/solution to the task (collects advice from the user if in doubt)
Semantic Web Services
Bringing the Semantic Web to its full potential

Dynamic

Web Services
UDDI, WSDL, SOAP

Intelligent Web Services

Static

WWW
URI, HTML, HTTP

Semantic Web
RDF, RDF(S), OWL

Syntax

Semantics

INF5100 Autumn 2007
Some research challenges...

• How do we
  – effectively query huge numbers of decentralized information repositories of varying scales?
  – align and map between ontologies?
  – construct a Semantic Web browser effectively visualizing and navigating the huge connected RDF graph?

• Provenance
  – the when, where, and conditions under which data originated
  – has become key requirement in many applications
  – need to understand
    • how information and concepts are spread on the Web
    • how provenance and trustworthiness can be established
    • may need help of researchers in areas of social network analysis, epidemology etc.