Semantic Engagement

Dieter Fensel, Andreea Gagiu, Birgit Leiter, and Andreas Thalhammer

Semantic Engagement

Why use semantics?

• Problems with current day search engines:
  – Recall issues
  – Results are dependent on the vocabulary
  – Results are single Web pages
  – Human involvement is necessary for result interpretation
  – Results of Web searches are not readily accessible by other software tools

• Content is not machine-readable:
  – It is difficult to distinguish between:
    “I am a professor of computer science.”
    and
    “You may think, I am a professor of computer science. Well, actually. . .”
Outline

1. Overview
2. Semantic Analysis (= Natural Language Processing)
3. Semantics as a channel (= Semantic vocabularies)
4. Semantic Content Modelling (= Ontologies)
5. Semantic Match Making (= Automatic distribution)
6. Summary

Semantic Analysis

What a computer understands from text messages:
Semantic Analysis

What is Semantic Analysis?

- Discovering facts in texts
- Deriving additional facts from them
- Somewhere in the Web the text fragment “Dieter is married to Anna” occurs (extracted statement)
- Named Entity Recognition tells us that Dieter is a (German) male given name, and Anna is a female given name (enriched with background knowledge)
- We can infer that Dieter and Anna are persons and
  - Dieter is male
  - Anna is female
  - Anna is married to Dieter
  - What with “Anna-Marie is married with Dieter”? (derive new facts)

Semantic as a channel
Semantic as a channel

Not to be interpreted by humans, but machines can make something out of it:

• Publishing Linked Data (data represented in accordance to the Semantic Web paradigms) can take various forms:
  – serialized graph (e.g. a RDF-XML file)
  – hidden in markup of the text
  – access through open graph databases (triple stores)

• Publishing Linked Data also involves publishing the used format:
  – There is a huge amount of different formats
  – Formats are often used in combination with another
Semantic Content Modelling

Separate format and potential channel.

**Same Event**

Semantic Content Modelling

- **A Ontology is a**
  - “formal (mathematical),
  - explicit specification (male and female are disjunct)
  - of a shared conceptualisation”

- (... of a domain)
  [Gruber, 1993]
Semantic Content Modelling

Weaver

Branch specific concepts

Distribute content

Collect feedback + statistics

Web/Blog

Social Web

Web 3.0/Mobile/Other

Semantic Match Making

Matcher

Branch specific concepts

Distribute content

Collect feedback + statistics

Web/Blog

Social Web

Web 3.0/Mobile/Other
Semantic Match Making

- The number of digital publishing channels has increased exponentially in the past decade
- Content production has risen tremendously in the past century
- Everybody has to put a lot of content into a lot of channels
- Manual efforts begin to be futile
- Automatic review and adjustment of content and dissemination to channels

Outline

1. Overview
2. Semantic Analysis (= Natural Language Processing)
3. Semantics as a channel (= Semantic vocabularies)
4. Semantic Content Modelling (=Ontologies)
5. Semantic Match Making (= Automatic distribution)
6. Summary
Semantic Analysis

Text mining:

“Text mining, sometimes alternately referred to as text data mining, roughly equivalent to text analytics, refers to the process of deriving high-quality information from text.”

→ How do we get “high-quality information”?

Example:

“I saw the man on the hill with a telescope”

Quiz questions:
• Do I have a telescope?
• Does the hill have a telescope?
• Does the man have a telescope?
• Is the man on the hill?
• Am I on the hill?
• Is the telescope on the hill?
Semantic Analysis

Methods:

**Key methods include:**

- Statistics
- Machine Learning
- Linguistics

Seven typical tasks of Information Extraction from Natural Language:

1. Topic detection
2. Named entity recognition
3. Co-reference and Disambiguation
4. Relation Extraction
5. Sentiment detection and Opinion mining
6. Social annotation
7. Text summarization
1. Topic detection

“In linguistics, the topic (or theme) of a sentence is often defined as what is being talked about, and the comment (rheme or focus) is what is being said about the topic.”


- Topic detection: detect the topics of a document (e.g. chat log, tweets, etc) on a meta level.
- Catalogues for topics: Open Directory Project (dmoz), Wikipedia,...
- Techniques:
  - Classification
  - Clustering
  - other Machine Learning techniques

Detected topics could be:
US elections, campaign, michelle obama, donations
2. Named Entity Recognition (NER)

- NER involves identification of *proper names* in texts, and classification into a set of predefined categories of interest.
- Three universally accepted categories: person, location and organisation
- Often also: measures (percent, money, weight etc), email addresses, recognition of date/time expressions etc.
- Domain-specific entities: names of hotels, medical conditions, names of ships, bibliographic references etc.
- "Named entity recognition: recognition of known entity names (for people and organizations), place names, temporal expressions, and certain types of numerical expressions, employing existing knowledge of the domain or information extracted from other sentences.
- For example, in processing the sentence "M. Smith likes fishing", named entity detection would denote detecting that the phrase "M. Smith" does refer to a person, but without necessarily having (or using) any knowledge about a certain M. Smith who is (or, "might be") the specific person whom that sentence is talking about. "

---

2. Named Entity Recognition (NER)
3. Co-reference and Disambiguation

"In linguistics, co-reference occurs when multiple expressions in a sentence or document refer to the same thing; or in linguistic jargon, they have the same 'referent'."


- Is used connect information, e.g.
  "I bought a new car. It is a green Mercedes convertible with 200 horse power"

"In computational linguistics, word-sense disambiguation (WSD) is an open problem of natural language processing, which governs the process of identifying which sense of a word (i.e. meaning) is used in a sentence, when the word has multiple meanings."


- Knowing who or what is meant:
  "Former president George Bush started the war in Irak, code-named Operation Desert Storm."

4. Relation Extraction

"A relationship extraction task requires the detection and classification of semantic relationship mentions within a set of artifacts, typically from text or XML documents. The task is very similar to that of information extraction (IE), but IE additionally requires the removal of repeated relations (disambiguation) and generally refers to the extraction of many different relationships."


Example: "Dieter is married to Anna"

Relation

Dieter Fensel

Anna Fensel
5. Sentiment and Opinion Detection

- "Sentiment analysis and opinion refers to the application of natural language processing, computational linguistics, and text analytics to identify and extract subjective information in source materials.
- Sentiment analysis aims to determine the attitude of a speaker or a writer with respect to some topic or the overall contextual polarity of a document.
- The attitude may be his or her judgment or evaluation, affective state, or the intended emotional communication." (wikipedia)
- A sentiment is a thought, view, or attitude, especially one based mainly on emotion rather than reason.
- Must consider features such as:
  - Subtlety of sentiment expression e.g. irony
  - Domain/context dependence
  - Effect of syntax on semantics
5. Sentiment and Opinion Detection

- Extraction of opinions and their meaning from text.
- Very difficult and not yet solved task.
- Example:
  1. This is a great hotel.
  2. A great amount of money was spent for promoting this hotel.
  3. One might think this is a great hotel.

Opinion Barack Obama: „We don't have enough money, yet“
6. Social Annotation

- Developed for web users to organize and share their favorite web pages online by social annotations

- Emergent useful information that has been explored for folksonomy, visualization, semantic web, etc

- : delicious, bibsonomy, last.fm, ...

6. Social Annotation
7. Text Summarization

- Automatic summarization involves reducing a text document or a larger corpus of multiple documents into a short set of words or paragraph that conveys the main meaning of the text.
  - Extractive methods work by selecting a subset of existing words, phrases, or sentences in the original text to form the summary.
  - Abstractive methods build an internal semantic representation and then use natural language generation techniques to create a summary that is closer to what a human might generate.

- The state-of-the-art abstractive methods are still quite weak, so most research has focused on extractive methods.

- Two particular types of summarization often addressed in the literature are
  - keyphrase extraction, where the goal is to select individual words or phrases to "tag" a document,
  - and document summarization, where the goal is to select whole sentences to create a short paragraph summary.” wikipedia
Outline

1. Overview
2. Semantic Analysis (= Natural Language Processing)
3. Semantics as a channel (= Semantic vocabularies)
4. Semantic Content Modelling (=Ontologies)
5. Semantic Match Making (= Automatic distribution)
6. Summary

Semantic as a Channel

The Semantic Web Approach

- Represent Web content in a form that is more easily machine-processable.
- Use intelligent techniques to take advantage of these representations.
- Knowledge will be organized in conceptual spaces according to its meaning.
- Automated tools for maintenance and knowledge discovery
- Semantic query answering
- Query answering over several documents
- Defining who may view certain parts of information (even parts of documents) will be possible.
- Semantic Web does not rely on text-based manipulation, but rather on machine-processable metadata
Semantic as a Channel

• **Scope:** Add machine-processable semantics to the information
  ⇒ Search and aggregation engines can provide much better service in finding and retrieving information

• **Search Engine Optimization**
  – Are potential customers finding your web site?
  – Is it possible that potential customers might not be aware that your site exists?
  – Do your targeted search terms have high search engine rankings?
  – Does your website attract a large number of daily visitors?

• Search engines are driven by keywords:
  – Search engine optimization is concerned with improving the visibility of a website or web page in search engines’ unpaid search results
  – The more frequent a site appears in the search results list, the more visitors it will receive

• Semantic Search:
  – Semantic search tries to understand the searcher’s intent and meaning of the query instead of parsing the keywords like a dictionary

Semantic as a Channel

**Implementations – Rich Snippets**

• Implementation realization of an application, plan, idea, model, or design.

• Snippets—the few lines of text that appear under every search result—are designed to give users a sense for what’s on the page and why it’s relevant to their query.

• If Google understands the content on your pages, we can create rich snippets—detailed information intended to help users with specific queries.
Semantic as a Channel

- Google’s rich snippets:

![Image](https://www.sti-innsbruck.at)

Semantic as a Channel

- SPARQL query [1]:

```sparql
PREFIX bif: <http://linkeddata.org/bif#>
PREFIX lgeo: <http://lod-cloud.net/geo#>
PREFIX lgdo: <http://lod-cloud.net/lgdo#>

SELECT ?tourismname ?tourism ?tourismgeo
FROM <http://linkedgeodata.org>
WHERE {
  ?tourism a lgdo:Tourism .
  Filter(bif:st_intersects
    (?tourismgeo, bif:st_point (11.404102, 47.269212), 1)) .
}
```

[1] Prefixes are omitted for reasons of simplicity

![Image](https://www.sti-innsbruck.at)
Semantic as a Channel

Evolution of the Web: Web of Data

[Picture from [3]]

"As We May Think", 1945

[Picture from [4]]
Motivation: From a Web of Documents to a Web of Data

- Web of Documents
  
  ![Diagram of Web of Documents]

- Fundamental elements:
  1. Names (URIs)
  2. Documents (Resources) described by HTML, XML, etc.
  3. Interactions via HTTP
  4. (Hyper)Links between documents or anchors in these documents

- Shortcomings:
  - Untyped links
  - Web search engines fail on complex queries

Motivation: From a Web of Documents to a Web of Data

- Web of Documents
  
  ![Diagram of Web of Documents]

- Web of Data
  
  ![Diagram of Web of Data]
Motivation: From a Web of Documents to a Web of Data

- Characteristics:
  - Links between arbitrary things (e.g., persons, locations, events, buildings)
  - Structure of data on Web pages is made explicit
  - Things described on Web pages are named and get URIs
  - Links between things are made explicit and are typed

Web of Data

“Things”

Vision of the Web of Data

- The Web today
  - Consists of data silos which can be accessed via specialized search engines in an isolated fashion.
  - One site (data silo) has movies, the other reviews, again another actors.
  - Many common things are represented in multiple data sets
  - Linking identifiers link these data sets

- The Web of Data is envisioned as a global database
  - consisting of objects and their descriptions
  - in which objects are linked with each other
  - with a high degree of object structure
  - with explicit semantics for links and content
  - which is designed for humans and machines

Content on this slide by Chris Bizer, Tom Heath and Tim Berners-Lee
The three dimensions

Format
e.g. RDFa

Implementation
e.g. OWLIM

Vocabulary
e.g. foaf

- Format is an explicit set of requirements to be satisfied by a material, product, or service.
  - The most known examples are RDF and OWL.

- A (Semantic Web) vocabulary can be considered as a special form of (usually light-weight) ontology, or sometimes also merely as a collection of URIs with an (usually informally) described meaning*.
  - URI = uniform resource identifier
  - Semantic vocabularies include: FOAF, Dublin Core, Good Relations, etc.

- Implementation realization of an application, plan, idea, model, or design.
  - OWLIM - a family of semantic repositories, or RDF database management system

* http://semanticweb.org/wiki/Ontology
Semantic Formats

Format

- an explicit set of requirements to be satisfied by a material, product, or service.
- is an encoded format for converting a specific type of data to displayable information.

Methods of describing Web content:

<table>
<thead>
<tr>
<th>Format</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>RDFs</td>
<td>1998</td>
</tr>
<tr>
<td>RDF</td>
<td>1999</td>
</tr>
<tr>
<td>RDFa</td>
<td>2004</td>
</tr>
<tr>
<td>Microformats</td>
<td>2005</td>
</tr>
<tr>
<td>OWL</td>
<td>2007</td>
</tr>
<tr>
<td>SPARQL</td>
<td>2008</td>
</tr>
<tr>
<td>OWL 2</td>
<td>2009</td>
</tr>
<tr>
<td>RDF+</td>
<td>2010</td>
</tr>
<tr>
<td>Microdata</td>
<td>2011</td>
</tr>
</tbody>
</table>
Semantic Formats

Format – HTML Meta Elements

• HTML or XHTML elements which provide structured metadata about a Web page
• Represented using the <meta...> element
• Can be used to specify page description, keywords and any other metadata not provided through the other head elements and attributes
• Example:

```html
<meta http-equiv="Content-Type" content="text/html" />
```

Semantic Formats

Format – HTML Meta Elements

• Search engine optimization attributes: keywords, description, language, robots
  – keywords attribute - although popular in the 90s, search engine providers realized that information stored in meta elements (especially the keywords attribute) was often unreliable and misleading, or created to draw users towards spam sites
  – description attribute - provides concise explanation of a Web page's content
  – the language attribute - tells search engines what natural language the website is written in
  – the robots attribute - controls whether or not search engine spiders are allowed to index a page, and whether or not they should follow links from a page
Semantic Formats

Format – HTML Meta Elements

• Example: metadata contained by www.wikipedia.org:

```html
<meta charset="utf-8">
<meta name="title" content="Wikipedia">
<meta name="description" content="Wikipedia, the free encyclopedia that anyone can edit.">
<meta name="author" content="Wikimedia Foundation">
<meta name="copyright" content="Creative Commons Attribution-Share Alike 3.0 and GNU Free Documentation License">
<meta name="publisher" content="Wikimedia Foundation">
<meta name="language" content="Many">
<meta name="robots" content="index, follow">
<!--[if IE 7]>
<meta http-equiv="imagetoolbar" content="no">
<![endif]-->
<meta name="viewport" content="initial-scale=1.0, user-scalable=yes">
```

Semantic Formats

HTML Meta Elements – There was soon a need

• For controlled vocabularies used in these meta data
• Richer formats to add these meta data and define their properties
RDF

Format – RDF

• The Resource Description Framework (RDF) is a language for representing information about resources in the World Wide Web.

• RDF provides a common framework for expressing information so it can be exchanged between applications without loss of meaning.

• It is based on the idea of identifying things using Web identifiers (called Uniform Resource Identifiers, or URIs) and describing resources in terms of simple properties and property values.

• Thus, RDF can represent simple statements about resources as a graph of nodes and arcs representing the resources, and their properties and values.

• It specifically supports the evolution of schemas over time without requiring all the data consumers to be changed.

Source: http://www.iis.sinica.edu.tw/~trc/public/courses/Fall2008/week15/slide-w15.html#%287%29
RDF Format – RDF

• Based on triples <subject, predicate, object>

• An RDF triple contains three components:
  – the subject, which is an RDF URI reference or a blank node
  – the predicate, which is an RDF URI reference
  – the object, which is an RDF URI reference, a literal or a blank node
  – An RDF triple is conventionally written in the order subject, predicate, object.
  – The predicate is also known as the property of the triple.

• Triple data model:
  <subject, predicate, object>

  – Subject: Resource or blank node
  – Predicate: Property
  – Object: Resource (or collection of resources), literal or blank node

• Example:
  <ex:john, ex:father-of, ex:bill>

RDF Format – RDF

• An RDF graph is a set of RDF triples.

• The set of nodes of an RDF graph is the set of subjects and objects of triples in the graph.

• Person ages (:age) and favorite friends (:fav)

Properties encoded as XML entities:

```xml
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:example="http://fake.host.edu/example-schema#">
  <example:Person>
    <example:name>Smith</example:name>
    <example:age>21</example:age>
    <example:fav>Jones</example:fav>
  </example:Person>

  <example:Person>
    <example:name>Blake</example:name>
    <example:age>12</example:age>
    <example:fav>George</example:fav>
  </example:Person>

</rdf:RDF>
```
Resources

- A resource may be:
  - Web page (e.g. http://www.w3.org)
  - A person (e.g. http://www.fensel.com)
  - A book (e.g. urn:isbn:0-345-33971-1)
  - Anything denoted with a URI!

- A URI is an identifier and not a location on the Web

- RDF allows making statements about resources:
  - http://www.w3.org has the format text/html
  - http://www.fensel.com has first name Dieter
  - urn:isbn:0-345-33971-1 has author Tolkien

URI, URN, URL

- A Uniform Resource Identifier (URI) is a string of characters used to identify a name or a resource on the Internet

- A URI can be a URL or a URN

- A Uniform Resource Name (URN) defines an item’s identity
  - the URN urn:isbn:0-395-36341-1 is a URI that specifies the identifier system, i.e. International Standard Book Number (ISBN), as well as the unique reference within that system and allows one to talk about a book, but doesn’t suggest where and how to obtain an actual copy of it

- A Uniform Resource Locator (URL) provides a method for finding it
  - the URL http://www.sti-innsbruck.at identifies a resource (STI’s home page) and implies that a representation of that resource (such as the home page’s current HTML code, as encoded characters) is obtainable via HTTP from a network host named www.sti-innsbruck.at
RDF-XML

- RDF-XML serialization of a university canteen (note the different vocabularies):

```xml
<rdf:Description rdf:about="http://lom.sti2.at/mensen/8">
  <rdf:type rdf:resource="http://purl.org/goodrelations/v1#Location"/>
  <gr:name>Musik Penzing</gr:name>
  <foaf:page rdf:resource="http://menu.mensen.at/index/index/locid/8"/>
  <vcard:adr rdf:resource="http://lom.sti2.at/mensen/8/adr"/>
  <vcard:geo rdf:resource="http://lom.sti2.at/mensen/8/geo"/>
</rdf:Description>

<rdf:Description rdf:about="http://lom.sti2.at/mensen/8/adr">
  <rdf:type rdf:resource="http://www.w3.org/2006/vcard/ns#Work"/>
  <vcard:street-address>Penzinger Straße 7</vcard:street-address>
  <vcard:postal-code>1140</vcard:postal-code>
  <vcard:locality>Wien</vcard:locality>
  <vcard:country>Austria</vcard:country>
</rdf:Description>

<rdf:Description rdf:about="http://lom.sti2.at/mensen/8/tel">
  <rdf:type rdf:resource="http://www.w3.org/2006/vcard/ns#Work"/>
  <rdf:value>+43 1 89 42 146</rdf:value>
</rdf:Description>

<rdf:Description rdf:about="http://lom.sti2.at/mensen/8/geo">
  <vcard:latitude rdf:datatype="http://www.w3.org/2001/XMLSchema#double">48.1897501</vcard:latitude>
</rdf:Description>
```

RDFS Vocabulary

- RDFS Extends the RDF Vocabulary
- RDFS vocabulary is defined in the namespace:

```
http://www.w3.org/2000/01/rdf-schema#
```

RDFS Classes

- rdfs:Resource
- rdfs:Class
- rdfs:Literal
- rdfs:Datatype
- rdfs:Container
- rdfs:ContainerMembershipProperty

RDFS Properties

- rdfs:domain
- rdfs:range
- rdfs:subPropertyOf
- rdfs:subClassOf
- rdfs:member
- rdfs:seeAlso
- rdfs:isDefinedBy
- rdfs:comment
- rdfs:label
RDFS Principles

- **Resource**
  - All resources are implicitly instances of `rdfs:Resource`

- **Class**
  - Describe sets of resources
  - Classes are resources themselves - e.g. Webpages, people, document types
    - Class hierarchy can be defined through `rdfs:subClassOf`
    - Every class is a member of `rdfs:Class`

- **Property**
  - Subset of RDFS Resources that are properties
    - **Domain**: class associated with property: `rdfs:domain`
    - **Range**: type of the property values: `rdfs:range`
    - Property hierarchy defined through: `rdfs:subPropertyOf`
OWL

- Web Ontology Language (OWL)
- Used to define complex semantic relations
- Defines formal semantics

Format – OWL

- Family of knowledge representation languages for authoring ontologies
- WebOnt developed OWL language
- OWL based on earlier languages OIL and DAML+OIL
- Characterized by formal semantics and RDF/XML-based serializations for the Semantic Web
- Endorsed by the World Wide Web Consortium (W3C)

Source: McGuinness, COGNA October 3, 2003
Design Goals for OWL

• **Shareable**  
  – Ontologies should be publicly available and different data sources should be able to commit to the same ontology for shared meaning. Also, ontologies should be able to extend other ontologies in order to provide additional definitions.

• **Changing over time**  
  – An ontology may change during its lifetime. A data source should specify the version of an ontology to which it commits.

• **Interoperability**  
  – Different ontologies may model the same concepts in different ways. The language should provide primitives for relating different representations, thus allowing data to be converted to different ontologies and enabling a "web of ontologies."

• **Inconsistency detection**  
  – Different ontologies or data sources may be contradictory. It should be possible to detect these inconsistencies.

• **Balancing expressivity and complexity**  
  – The language should be able to express a wide variety of knowledge, but should also provide for efficient means to reason with it. Since these two requirements are typically at odds, the goal of the web ontology language is to find a balance that supports the ability to express the most important kinds of knowledge.

• **Ease of use**  
  – The language should provide a low learning barrier and have clear concepts and meaning. The concepts should be independent from syntax.
Design Goals for OWL

• **Compatible with existing standards**
  – The language should be compatible with other commonly used Web and industry standards. In particular, this includes XML and related standards (such as XML Schema and RDF), and possibly other modeling standards such as UML.

• **Internationalization**
  – The language should support the development of multilingual ontologies, and potentially provide different views of ontologies that are appropriate for different cultures.

OWL Sublanguages

• The W3C-endorsed OWL specification includes the definition of three variants of OWL, with different levels of expressiveness (ordered by increasing expressiveness):
  – **OWL Lite** - originally intended to support those users primarily needing a classification hierarchy and simple constraints
  – **OWL DL** - was designed to provide the maximum expressiveness possible while retaining computational completeness, decidability, and the availability of practical reasoning algorithms.
  – **OWL Full** - designed to preserve some compatibility with RDF Schema

Each of these sublanguage is a syntactic extension of its simpler predecessor.

• The following set of relations hold. Their inverses do not.
  – Every legal OWL Lite ontology is a legal OWL DL ontology.
  – Every legal OWL DL ontology is a legal OWL Full ontology.
  – Every valid OWL Lite conclusion is a valid OWL DL conclusion.
  – Every valid OWL DL conclusion is a valid OWL Full conclusion.

• Development of OWL Lite tools has thus proven almost as difficult as development of tools for OWL DL, and OWL Lite is not widely used

Source: McGuinness, COGNA October 3, 2003
Format – OWL

• Class Axioms
  – oneOf (enumerated classes)
  – disjointWith
  – sameClassAs applied to class expressions
  – rdfs:subClassOf applied to class expressions

• Boolean Combinations of Class Expressions
  – unionOf
  – intersectionOf
  – complementOf

• Arbitrary Cardinality
  – minCardinality
  – maxCardinality
  – cardinality

• Filler Information
  – hasValue Descriptions can include specific value information

Example:

```xml
<owl:Class>
  <owl:intersectionOf rdf:parseType="collection">
    <owl:Class rdf:about="#Person"/>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasChild"/>
      <owl:allValuesFrom>
        <owl:unionOf rdf:parseType="collection">
          <owl:Class rdf:about="#Doctor"/>
          <owl:Restriction>
            <owl:onProperty rdf:resource="#hasChild"/>
            <owl:someValuesFrom rdf:resource="#Doctor"/>
          </owl:Restriction>
        </owl:unionOf>
      </owl:allValuesFrom>
    </owl:Restriction>
  </owl:intersectionOf>
</owl:Class>
```

Source: McGuinness, COGNA October 3, 2003
Experience with OWL

• OWL playing key role in increasing number & range of applications
  – eScience, eCommerce, geography, engineering, defence, …
  – E.g., OWL tools used to identify and repair errors in a medical ontology: “would have led to missed test results if not corrected”

• Experience of OWL in use has identified restrictions:
  – on expressivity
  – on scalability

• These restrictions are problematic in some applications

• Research has now shown how some restrictions can be overcome

• W3C OWL WG has updated OWL accordingly
  – Result is called OWL 2

• OWL 2 is now a Proposed Recommendation

OWL 2 in a Nutshell

• Extends OWL with a small but useful set of features
  – That are needed in applications
  – For which semantics and reasoning techniques are well understood
  – That tool builders are willing and able to support

• Adds profiles
  – Language subsets with useful computational properties (EL, RL, QL)

• Is fully backwards compatible with OWL:
  – Every OWL ontology is a valid OWL 2 ontology
  – Every OWL 2 ontology not using new features is a valid OWL ontology

• Already supported by popular OWL tools & infrastructure:
  – Protégé, HermiT, Pellet, FaCT++, OWL API
**OWL2**

**Format – OWL 2**

- Inherits OWL 1 language features
- Makes some patterns easier to write
- Does not change expressiveness, semantics and complexity
- Provides more efficient processing in implementations
- Syntactic sugar:
  - DisjointUnion - Union of a set of classes; all the classes are pairwise disjoint
  - DisjointClasses - A set of classes; all the classes are pairwise disjoint
  - NegativeObjectPropertyAssertion - Two individuals; a property does not hold between them
  - NegativeDataPropertyAssertion - An individual; a literal; a property does not hold between them
- OWL 2 allows the same identifiers (URIs) to denote individuals, classes, and properties
- Interpretation depends on context
- A very simple form of meta-modelling

Source: McGuinness, CIGNA October 3, 2003

**OWL2**

- Qualified cardinality restrictions
  - e.g., persons having two friends who are republicans
- Property chains
  - e.g., the brother of your parent is your uncle
- Local reflexivity restrictions
  - e.g., narcissists love themselves
- Reflexive, irreflexive, and asymmetric properties
  - e.g., nothing can be a proper part of itself (irreflexive)
- Disjoint properties
  - e.g., you can’t be both the parent of and child of the same person
- Keys
  - e.g., country + license plate constitute a unique identifier for vehicles
Format – OWL 2

- An OWL 2 profile (commonly called a fragment or a sublanguage in computational logic) is a trimmed down version of OWL 2 that trades some expressive power for the efficiency of reasoning.

- OWL 2 profiles
  - OWL 2 EL is particularly useful in applications employing ontologies that contain very large numbers of properties and/or classes.
  - OWL 2 QL is aimed at applications that use very large volumes of instance data, and where query answering is the most important reasoning task.
  - OWL 2 RL is aimed at applications that require scalable reasoning without sacrificing too much expressive power.

- OWL 2 profiles are defined by placing restrictions on the structure of OWL 2 ontologies.

Example property chains in OWL2:
Rule Interchanged Format (RIF)

Format – RIF

- A collection of dialects (rigorously defined rule languages)
- Intended to facilitate rule sharing and exchange
- RIF framework is a set of rigorous guidelines for constructing RIF dialects in a consistent manner
- The RIF framework includes several aspects:
  - Syntactic framework
  - Semantic framework
  - XML framework
- RIF can be used to map between vocabularies (one of the proposed use cases)

Goals:

- Exchange of Rules
  - The primary goal of RIF is to facilitate the exchange of rules

- Consistency with W3C specifications
  - A W3C specification that builds on and develops the existing range of specifications that have been developed by the W3C
  - Existing W3C technologies should fit well with RIF

- Scale Adoption
  - Rules interchange becomes more effective the wider is their adoption (“network effect”)
Rule Interchanged Format (RIF)

Format – RIF

• The standard RIF dialects are:
  – **Core** - the fundamental RIF language. It is designed to be the common subset of most rule engines. (It provides "safe" positive datalog with builtins.)
  – **BLD** (Basic Logic Dialect) - adds a few things that Core doesn't have: logic functions, equality in the then-part, and named arguments. (This is positive Horn logic, with equality and builtins.)
  – **PRD** (Production Rules Dialect) - adds a notion of forward-chaining rules, where a rule fires and then performs some action, such as adding more information to the store or retracting some information.

• Although RIF dialects were designed primarily for interchange, each dialect is a standard rule language and can be used even when portability and interchange are not required.

• The XML syntax is the only one defined as a standard for interchange. Various presentation syntaxes are used in the specification, but they are not recommended for sending between different systems.

Rule Interchanged Format (RIF)

Requirements

• Compliance model
  – Clear conformance criteria, defining what is or is not a conformant to RIF

• Different semantics
  – RIF must cover rule languages having different semantics

• Limited number of dialects
  – RIF must have a standard core and a limited number of standard dialects based upon that core

• OWL data
  – RIF must cover OWL knowledge bases as data where compatible with RIF semantics
Rule Interchanged Format (RIF)

Requirements

- RDF data
  - RIF must cover RDF triples as data where compatible with RIF semantics

- Dialect identification
  - The semantics of a RIF document must be uniquely determined by the content of the document, without out-of-band data

- XML syntax
  - RIF must have an XML syntax as its primary normative syntax

- Merge rule sets
  - RIF must support the ability to merge rule sets

- Identify rule sets
  - RIF must support the identification of rule sets

Basic Principle: a Modular Architecture

- RIF wants to cover: rules in logic dialects and rules used by production rule systems (e.g. active databases)

  - Logic rules only add knowledge

  - Production rules change the facts!

  - Logic rules + Production Rules?
    - Define a logic-based core and a separate production-rule core
    - If there is an intersection, define the common core
Rule Interchanged Format (RIF)

Format – RIF

- A simplified example of RIF-Core rules combined with OWL to capture anatomical knowledge that can be used to help label brain cortex structures in MRI images.

Source: [http://www.w3.org/2005/rules/wiki/Modeling_Brain_Anatomy](http://www.w3.org/2005/rules/wiki/Modeling_Brain_Anatomy)

SPARQL

```
SELECT ?o WHERE { ?s dbpedia2:blackboard ?o. }
SELECT ?o WHERE { ?s dbpedia2:blackboard ?o. }
SELECT ?o WHERE { ?s dbpedia2:blackboard ?o. }
SELECT ?o WHERE { ?s dbpedia2:blackboard ?o. }
SELECT ?o WHERE { ?s dbpedia2:blackboard ?o. }
SELECT ?o WHERE { ?s dbpedia2:blackboard ?o. }
SELECT ?o WHERE { ?s dbpedia2:blackboard ?o. }
SELECT ?o WHERE { ?s dbpedia2:blackboard ?o. }
SELECT ?o WHERE { ?s dbpedia2:blackboard ?o. }
SELECT ?o WHERE { ?s dbpedia2:blackboard ?o. }
```
SPARQL

Format – SPARQL

- A recursive acronym for SPARQL Protocol and RDF Query Language
- On 15 January 2008, SPARQL 1.0 became an official W3C Recommendation
- Query language based on RDQL
- Used to retrieve and manipulate data stored in RDF format
- Uses SQL-like syntax

SPARQL

- RESTful interface:
  
  http://rdf.sti2.at:8080/openrdf-sesame/repositories/lom4?query%3Dselect%20%20%20where%20%7B%20%7B%7Frdf%20%3Fp%20%7Do%7D

- SPARQL protocol and RDF query language (recursive acronym)

```sparql
PREFIX vcard:<http://www.w3.org/2006/vcard/ns#>
PREFIX xsd:<http://www.w3.org/2001/XMLSchema#>
PREFIX gr:<http://purl.org/goodrelations/v1#>
PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>
PREFIX foaf:<http://xmlns.com/foaf/0.1/>

select ?lat where { 
  ?s rdf:type gr:Location. 
  ?s vcard:geo ?loc. 
  ?loc vcard:latitude ?lat. 
  FILTER(?lat > 48) 
}
```
SPARQL Queries

PREFIX uni: <http://example.org/uni/>
SELECT ?name
FROM <http://example.org/personal>
WHERE { ?s uni:name ?name. ?s rdf:type uni:lecturer }

• PREFIX
  – Prefix mechanism for abbreviating URIs
• SELECT
  – Identifies the variables to be returned in the query answer
  – SELECT DISTINCT
  – SELECT REDUCED
• FROM
  – Name of the graph to be queried
  – FROM NAMED
• WHERE
  – Query pattern as a list of triple patterns
• LIMIT
• OFFSET
• ORDER BY

SPARQL

Format – SPARQL

• Example SPARQL Query:
  – “Return the full names of all people in the graph”

PREFIX vCard: <http://www.w3.org/2001/vcard-rdf/3.0#>
SELECT ?fullName
WHERE {?x vCard:FN ?fullName}

@prefix ex: <http://example.org/#> .
@prefix vcard: <http://www.w3.org/2001/vcard-rdf/3.0#> .
ex:john
  vcard:FN "John Smith";
  vcard:N [ vcard:Given "John";
             vcard:Family "Smith" ];
  ex:hasAge 32;
  ex:marriedTo:mary.
ex:mary
  vcard:FN "Mary Smith";
  vcard:N [ vcard:Given "Mary";
             vcard:Family "Smith" ];
  ex:hasAge 29.

– Results:

<table>
<thead>
<tr>
<th>fullName</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;John Smith&quot;</td>
</tr>
<tr>
<td>&quot;Mary Smith&quot;</td>
</tr>
</tbody>
</table>

8/17/2012
**SPARQL Query keywords**

- **PREFIX**: based on namespaces

- **DISTINCT**: The DISTINCT solution modifier eliminates duplicate solutions. Specifically, each solution that binds the same variables to the same RDF terms as another solution is eliminated from the solution set.

- **REDUCED**: While the DISTINCT modifier ensures that duplicate solutions are eliminated from the solution set, REDUCED simply permits them to be eliminated. The cardinality of any set of variable bindings in an REDUCED solution set is at least one and not more than the cardinality of the solution set with no DISTINCT or REDUCED modifier.

- **LIMIT**: The LIMIT clause puts an upper bound on the number of solutions returned. If the number of actual solutions is greater than the limit, then at most the limit number of solutions will be returned.

**SPARQL Query keywords**

- **OFFSET**: OFFSET causes the solutions generated to start after the specified number of solutions. An OFFSET of zero has no effect.

- **ORDER BY**: The ORDER BY clause establishes the order of a solution sequence.

- Following the ORDER BY clause is a sequence of order comparators, composed of an expression and an optional order modifier (either ASC() or DESC()). Each ordering comparator is either ascending (indicated by the ASC() modifier or by no modifier) or descending (indicated by the DESC() modifier).
Microdata

Format – Microdata

- Use HTML5 elements to include semantic descriptions into web documents aiming to replace RDFa and Microformats.
- Introduce new tag attributes to include semantic data into HTML.
- Unless you know that your target consumer only accepts RDFa, you are probably best going with microdata.
- While many RDFa-consuming services (such as the semantic search engine Sindice) also accept microdata, microdata-consuming services are less likely to accept RDFa.

Advantages:
- the variable groupings of data within published area tables may not be the detail required for a particular application (e.g., age group, ethnic group or occupational classification).
- the cross-tabulations of variables available in area tables may not be those needed for a study (e.g., counts of individuals by age and ethnic group and occupation).

- Search engines, web crawlers, and browsers can extract and process Microdata from a web page and use it to provide a richer browsing experience for users.
- Microdata uses a supporting vocabulary to describe an item and name-value pairs to assign values to its properties.
- Microdata helps technologies such as search engines and web crawlers better understand what information is contained in a web page, providing better search results.

Two important vocabularies:
http://www.data-vocabulary.org/
http://schema.org/
Microdata

Microdata Global Attributes

- **itemscope** – Creates the Item and indicates that descendants of this element contain information about it.

- **itemtype** – A valid URL of a vocabulary that describes the item and its properties context.

- **itemid** – Indicates a unique identifier of the item.

- **itemprop** – Indicates that its containing tag holds the value of the specified item property. The properties name and value context are described by the items vocabulary. Properties values usually consist of string values, but can also use URLs using the a element and its href attribute, the img element and its src attribute, or other elements that link to or embed external resources.

- **itemref** – Properties that are not descendants of the element with the itemscope attribute can be associated with the item using this attribute. Provides a list of element itemids with additional properties elsewhere in the document.

---

**Example: Microdata in use**

```html
<div>
  My name is Bob Smith but people call me Smithy. Here is my home page:
  <a href="http://www.example.com">www.example.com</a>
  I live in Albuquerque, NM and work as an engineer at ACME Corp.
</div>
```

Enriched with microdata:

```html
<div itemscope itemtype="http://data-vocabulary.org/Person">
  My name is <span itemprop="name">Bob Smith</span>
  but people call me <span itemprop="nickname">Smithy</span>
  Here is my home page:
  <a href="http://www.example.com">www.example.com</a>
  I live in Albuquerque, NM and work as an engineer at ACME Corp
</div>
```
Microdata

Format – Microdata

• Example without microdata:

```html
<section>
    Hello, my name is John Doe, I am a graduate research assistant at the University of Dreams. My friends call me Johnny. You can visit my homepage at http://www.JohnnyD.com. I live at 1234 Peach Drive Warner Robins, Georgia.
</section>
```

Microdata

Format – Microdata

• Example using microdata:

```html
<section itemscope itemtype="http://data-vocabulary.org/Person">
    Hello, my name is <span itemprop="name">John Doe</span>, I am a graduate research assistant at the University of Dreams. My friends call me <span itemprop="nickname">Johnny</span>. You can visit my homepage at http://www.JohnnyD.com. I live at 1234 Peach Drive Warner Robins, Georgia.
</section>
```
Microformats

What are Microformats?

• An approach to add meaning to HTML elements and to make data structures in HTML pages explicit.

• “Designed for humans first and machines second, microformats are a set of simple, open data formats built upon existing and widely adopted standards. Instead of throwing away what works today, microformats intend to solve simpler problems first by adapting to current behaviours and usage patterns (e.g. XHTML, blogging).” [6]

What are Microformats?

• Are highly correlated with semantic (X)HTML / “Real world semantics” / “Lowercase Semantic Web” [9].
• Real world semantics (or the Lowercase Semantic Web) is based on three notions:
  – Adding of simple semantics with microformats (small pieces)
  – Adding semantics to the today's Web instead of creating a new one (evolutionary not revolutionary)
  – Design for humans first and machines second (user centric design)
• A way to combine human with machine-readable information.
• Provide means to embed structured data in HTML pages.
• Build upon existing standards.
• Solve a single, specific problem (e.g. representation of geographical information, calendaring information, etc.).
• Provide an “API” for your website.
• Build on existing (X)HTML and reuse existing elements.
• Work in current browsers.
• Follow the DRY principle (“Don’t Repeat Yourself”).
• Compatible with the idea of the Web as a single information space.
Format – Microformats

- Directly use meta tags of XHTML to embed semantic information in web documents;
- Microformats were developed as a competing approach directly using some existing HTML tags to include meta data in HTML documents;
- As of 2010, microformats allow the encoding and extraction of events, contact information, social relationships and so on;
- Advantages:
  - you can publish a **single, human readable** version of your information in HTML and then make it machine readable with the addition of a few standard class names;
  - No need to learn another language;
  - Easy to add;
- **However:** they overload the class tag which causes problems for some parsers as it makes semantic information and styling markup hard to differentiate.

Microformats Illustrated

---

Structure understandable by humans  
Structure understandable by machines
Microformats

Example: Microformats in use

<div class="vcard"><img class="photo" src="www.example.com/bobsmith.jpg" /><strong class="fn">Bob Smith</strong><span class="title">Senior editor</span> at <span class="org">ACME Reviews</span><span class="adr"><span class="street-address">200 Main St</span><span class="locality">Desertville</span>, <span class="region">AZ</span> <span class="postal-code">12345</span></span></div>

Enriched with microformat:

<div class="vcard"><img class="photo" src="www.example.com/bobsmith.jpg" /><strong class="fn">Bob Smith</strong><span class="title">Senior editor</span> at <span class="org">ACME Reviews</span><span class="adr"><span class="street-address">200 Main St</span><span class="locality">Desertville</span>, <span class="region">AZ</span> <span class="postal-code">12345</span></div>

Source: http://support.google.com/webmasters/bin/answer.py?hl=en&answer=146897

RDFa

- RDFa is a W3C recommendation.
- RDFa is a serialization syntax for embedding an RDF graph into XHTML.
- Goals: Bringing the Web of Documents and the Web of Data closer together.
- Overcomes some of the drawbacks of microformats
- Both for human and machine consumption.
- Follows the DRY ("Don’t Repeat Yourself") – principles.
- RDFa is domain-independent. In contrast to the domain-dedicated microformats, RDFa can be used for custom data and multiple schemas.
- Benefits inherited from RDF: Independence, modularity, evolvability, and reusability.
- Easy to transform RDFa into RDF data.
- Tools for RDFa publishing and consumption exist.
RDFa

Syntax: How to use RDFa in XHTML

- Relevant XHTML attributes: `@rel`, `@rev`, `@content`, `@href`, and `@src` (examples and explanations on the following slides)

- New RDFa-specific attributes: `@about`, `@property`, `@resource`, `@datatype`, and `@typeof` (examples and explanations on the following slides)

Format – RDFa

- Is a W3C Recommendation that adds a set of attribute-level extensions to XHTML for embedding rich metadata within Web documents.
- Adds a set of attribute-level extensions to XHTML enabling the embedding of RDF triples;
- Integrates best with the W3C meta data stack built on top of RDF
- **Benefits** [Wikipedia RDFa, n.d.]:
  - **Publisher independence:** each website can use its own standards;
  - **Data reuse:** data is not duplicated - separate XML/HTML sections are not required for the same content;
  - **Self containment:** HTML and RDF are separated;
  - **Schema modularity:** attributes are reusable;
  - **Evolv-ability:** additional fields can be added and XML transforms can extract the semantics of the data from an XHTML file;
  - **Web accessibility:** more information is available to assistive technology.
- **Disadvantage:** the uptake of the technology is hampered by the web-master’s lack of familiarity with this technology stack
RDFa

Format – RDFa

- RDFa Attributes:
  - about and src – a URI or CURIE specifying the resource the metadata is about
  - rel and rev – specifying a relationship or reverse-relationship with another resource
  - href and resource – specifying the partner resource
  - property – specifying a property for the content of an element
  - content – optional attribute that overrides the content of the element when using the property attribute
  - datatype – optional attribute that specifies the datatype of text specified for use with the property attribute
  - typeof – optional attribute that specifies the RDF type(s) of the subject (the resource that the metadata is about).

Syntax: How to use RDFa in XHTML

- @ref: a whitespace separated list of CURIEs (Compact URIs), used for expressing relationships between two resources ('predicates');

- All content on this site is licensed under <a rel="license" href="http://creativecommons.org/licenses/by/3.0/"> a Creative Commons License </a>.
**Example: RDFa in use**

```html
<div xmlns:v="http://rdf.data-vocabulary.org/#" typeof="v:Person">
  My name is <span property="v:name">Bob Smith</span>, but people call me <span property="v:nickname">Smithy</span>. Here is my home page:
  I live in Albuquerque, NM and work as an <span property="v:title">engineer</span> at <span property="v:affiliation">ACME Corp</span>.
</div>
```

Enriched with RDFa:

```html
<div xmlns:v="http://rdf.data-vocabulary.org/#" typeof="v:Person">
  My name is <span property="v:name">Bob Smith</span>, but people call me <span property="v:nickname">Smithy</span>. Here is my home page:
  I live in Albuquerque, NM and work as an <span property="v:title">engineer</span> at <span property="v:affiliation">ACME Corp</span>.
</div>
```

Source: [http://support.google.com/webmasters/bin/answer.py?hl=en&answer=146898](http://support.google.com/webmasters/bin/answer.py?hl=en&answer=146898)

**Microformats, RDFa, Microdata**

- RDFa adds a set of attribute-level extensions to XHTML enabling the embedding of RDF triples
- Microformats directly use meta tags of XHTML to embed semantic information in web documents
- Microdata use HTML5 elements to include semantic descriptions into web documents aiming to replace RDFa and Microformats.
- For the moment, we have three competing proposals that should be supported in parallel until one of them can take a dominant role on the web.
Microformats, RDFa, Microdata

- RDFa integrates best with the W3C meta data stack built on top of RDF.
- However, this also seems to hamper the uptake of this technology by many webmasters that are not familiar with this technology stack.
- Microformats were developed as a competing approach directly using some existing HTML tags to include meta data in HTML documents.
- Actually, they overload the class tag which causes problems for some parsers as it makes semantic information and styling markup hard to differentiate.
- Microdata instead introduce new tag attributes to include semantic data into HTML.

Microformats, RDFa, Microdata

- The use of RDFa has increased rapidly, whereas the deployment of microformats in the same period has not advanced remarkably.
Microformats, RDFa, Microdata

- All three of them are intended for machines to interpret content, i.e. search engines

- Major search engines support Microformat in combination with the Schema.org vocabulary.

- If your choice is not supported by Google, Bing, and Yahoo your choice is useless.

→ schema.org

Semantic Channels: Vocabularies
Vocabularies:

- A (Semantic Web) vocabulary can be considered a special form of (usually light-weight) ontology, or sometimes also merely as a collection of URIs with an (usually informally) described meaning.

- Recap “what ontologies are”: “An ontology is a formal specification of a shared conceptualization”


... and a lot more
**Semantic Channels: Vocabularies**

**SIOC**

- Vocabulary to describe content in social channels

- Enables...
  - semantic links between online communities
  - to fully describe the content and structure of community sites
  - the integration of online community information
  - browsing of connected Semantic Web items
  - to add a social aspect to the Semantic Web

---

**Semantic Channels: Vocabularies**

**SIOC**

A SIOC document, unlike a traditional Web page, can be combined with other SIOC and RDF documents to create a unified database of information.
Vocabulary – DublinCore

- Early Dublin Core workshops popularized the idea of "core metadata" for simple and generic resource descriptions.
- Metadata terms are a set of vocabulary terms which can be used to describe resources for the purposes of discovery.
- The terms can be used to describe a full range of web resources: video, images, web pages etc. and physical resources such as books and objects like artworks.
- The Dublin Core standard includes two levels:
  - Simple Dublin Core comprises 15 elements;
  - Qualified Dublin Core includes three additional elements: Audience, Provenance and RightsHolder; as well as a group of element refinements, also called qualifiers, that refine the semantics of the elements in ways that may be useful in resource discovery.

Source: [http://dublincore.org](http://dublincore.org) (tutorials)

Characteristics of DublinCore:
- All elements are optional
- All elements are repeatable
- Elements may be displayed in any order
- Extensible
- International in scope

- The fifteen core elements are usable with or without qualifiers
- Qualifiers make elements more specific:
  - Element Refinements narrow meanings, never extend
  - Encoding Schemes give context to element values
- If your software encounters an unfamiliar qualifier, look it up – or just ignore it!

Source: [http://dublincore.org](http://dublincore.org) (tutorials)
Semantic Channels: Vocabularies

Vocabulary – DublinCore

- Expressing Dublin Core in HTML/XHTML meta and link elements:

```html
<head profile="http://dublincore.org/documents/dcq-html/">
<title>Expressing Dublin Core in HTML/XHTML meta and link elements</title>
<link rel="schema.DC" href="http://purl.org/dc/elements/1.1/" />
<link rel="schema.DCTERMS" href="http://purl.org/dc/terms/" />
<meta name="DC.title" lang="en" content="Expressing Dublin Core in HTML/XHTML meta and link elements" />
<meta name="DC.creator" content="Andy Powell, UKOLN, University of Bath" />
<meta name="DCTERMS.issued" scheme="DCTERMS.W3CDTF" content="2003-11-01" />
<meta name="DC.identifier" scheme="DCTERMS.URI" content="http://dublincore.org/documents/dcq-html/" />
<link rel="DCTERMS.replaces" hreflang="en" href="http://dublincore.org/documents/2000/08/15/dcq-html/" />
<meta name="DCTERMS.abstract" content="This document describes how qualified Dublin Core metadata can be encoded in HTML/XHTML &lt;meta&gt; elements" />
<meta name="DC.format" scheme="DCTERMS.IMT" content="text/html" />
<meta name="DC.type" scheme="DCTERMS.DCMIType" content="Text" />
</head>
```

Semantic Channels: Vocabularies

Vocabulary – FOAF

- Friend of a Friend

- Uses RDF to describe the relationship people have to other “things” around them

- FOAF permits intelligent agents to make sense of the thousands of connections people have with each other, their jobs and the items important to their lives;

- Because the connections are so vast in number, human interpretation of the information may not be the best way of analyzing them.

- FOAF is an example of how the Semantic Web attempts to make use of the relationships within a social context.
**Friend of a Friend (FOAF)**

- Vocabulary to link people and information using the Web.

- Enables...
  - to describe Agents, Documents, Groups, Organizations, Projects and their relationships.
  - a distributed social network that can be crawled.

---

**Friend of a Friend (FOAF)**

- Friend of a Friend is a project that aims at providing simple ways to describe people and relations among them.

- FOAF adopts RDF and RDFS


- Tools based on FOAF:
  - FOAF search ([http://foaf.qdos.com/](http://foaf.qdos.com/))
Semantic Channels: Vocabularies

**FOAF Schema**

Example

```xml
<foaf:Person>
    <foaf:name>Dan Brickley</foaf:name>
    <foaf:mbox_sha1sum>
        748934f32135cfcf6f8c06e253c53442721e15e7
    </foaf:mbox_sha1sum>
</foaf:Person>
```

Which says "there is a Person called Dan Brickley who has an email address whose sha1 hash is..."
FOAF RDF Example

```xml
<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/>
  <foaf:Person rdf:ID="DieterFensel">
    <foaf:name>Dieter Fensel</foaf:name>
    <foaf:title>Univ.-Prof. Dr.</foaf:title>
    <foaf:givenname>Dieter</foaf:givenname>
    <foaf:family_name>Fensel</foaf:family_name>
    <foaf:mbox_sha1sum>773a221a09f1887a224653c9de06c3480e714278a</foaf:mbox_sha1sum>
    <foaf:workplaceHomepage rdf:resource="http://www.sti-innsbruck.at"/>
    <foaf:workInfoHomepage rdf:resource="http://www.sti-innsbruck.at/about/team/details/?uid=40"/>
  </foaf:Person>
</rdf:RDF>
```

GoodRelations

- GoodRelations is a standardized vocabulary (also known as "schema", "data dictionary", or "ontology") for product, price, store, and company data that can ... 
  
  1. be embedded into existing static and dynamic Web pages and that 
  
  2. can be processed by other computers. This increases the visibility of your products and services in the latest generation of search engines, recommender systems, and other novel applications.
Semantic Channels: Vocabularies

Vocabulary – GoodRelations

- A lightweight ontology for annotating offerings and other aspects of e-commerce on the Web.
- The only OWL DL ontology officially supported by both Google and Yahoo.
- It provides a standard vocabulary for expressing things like
  - that a particular Web site describes an offer to sell cellphones of a certain make and model at a certain price,
  - that a pianohouse offers maintenance for pianos that weigh less than 150 kg,
  - or that a car rental company leases out cars of a certain make and model from a particular set of branches across the country.
- Also, most if not all commercial and functional details of e-commerce scenarios can be expressed, e.g. eligible countries, payment and delivery options, quantity discounts, opening hours, etc.

GoodRelations

Design Principles:

"Keep simple things simple and make complex things possible"

- Cater for LOD and OWL DL worlds
- Academically sound
- Industry-strength engineering
- Practically relevant

http://purl.org/goodrelations
Semantic Channels: Vocabularies

Vocabulary – GoodRelations

• Example:

```xml
<rdf:RDF xmlns="http://www.heppnetz.de/ontologies/examples/gr#"
  xmlns:toy="http://www.heppnetz.de/ontologies/examples/toy#"
  xmlns:gr="http://purl.org/goodrelations/v1#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:protege="http://protege.stanford.edu/plugins/owl/protege#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:owl="http://www.w3.org/2002/07/owl#">
  <owl:Ontology rdf:about="">
    <owl:imports rdf:resource="http://www.heppnetz.de/ontologies/examples/toy"/>
    <owl:imports rdf:resource="http://purl.org/goodrelations/v1"/>
  </owl:Ontology>

  <gr:BusinessEntity rdf:ID="ElectronicsCom">
    <gr:legalName rdf:datatype="&xsd:string">Electronics.com Ltd.</gr:legalName>
    <gr:offers rdf:resource="#Offering_1"/>
  </gr:BusinessEntity>
</rdf:RDF>
```

Semantic Channels: Vocabularies

Schema.org

• A microdata vocabulary.

• Schema.org is a collection of schemas, i.e., html tags, that webmasters can use to markup their pages in ways recognized by major search providers.

• Search engines including Bing, Google, Yahoo! and Yandex rely on this markup to improve the display of search results, making it easier for people to find the right web pages.
Example:

Imagine you have a page about the movie Avatar—a page with a link to a movie trailer, information about the director, and so on. Your HTML code might look something like this:

```html
<div>
  <h1>Avatar</h1>
  <span>Director: James Cameron (born August 16, 1954)</span>
  <span>Science fiction</span>
  <a href="../movies/avatar-theatrical-trailer.html">Trailer</a>
</div>
```
**Semantic Channels: Vocabularies**

**Schema.org**
- In-page structured data for search
- Do not ask “so, how do we describe hotels?”, but “how can we improve markup on existing pages that describe hotels?” (or Cars, Software, ...)
- Simplify publisher/webmaster experience
- Record agreements between search engines
- Central use case: augmented search results

**Hotel example:**

```html
<div itemscope itemtype="http://schema.org/Hotel">
  <span itemprop="name">Name of Hotel</span>
  <div itemprop="aggregateRating" itemscope itemtype="http://schema.org/AggregateRating">
    <span itemprop="ratingValue">4</span> stars - based on <span itemprop="reviewCount">321</span> reviews
  </div>
  <div itemprop="address" itemscope itemtype="http://schema.org/PostalAddress">
    <span itemprop="streetAddress">123 Fake Street</span>, <span itemprop="addressLocality">Seattle</span>, <span itemprop="addressRegion">Washington</span>, <span itemprop="postalCode">98146</span>
  </div>
  <span itemprop="telephone">(206) 123-4321</span>
  <a href="http://mapsurl.com/23452345" itemprop="maps">URL of Map</a>
  Price Range: <span itemprop="priceRange">$$</span>
</div>
```
Semantic Channels: Vocabularies

Implementations – Rich Snippets

• Three steps to rich snippets

1. Pick a markup format.
   Google suggests using microdata, but any of the three formats below are acceptable.
   • Microdata (recommended)
   • Microformats
   • RDFa

2. Mark up your content.
   Google supports rich snippets for these content types:
   • Reviews
   • People
   • Products
   • Businesses and organizations
   • Recipes
   • Events
   • Music
   • Google also recognizes markup for video content and uses it to improve our search results.
Semantic Channels: Implementations

Implementations – OWLIM

• OWLIM is a high-performance OWL repository
• Storage and Inference Layer (SAIL) for Sesame RDF database
• OWLIM performs OWL DLP reasoning
• It is uses the IRRE (Inductive Rule Reasoning Engine) for forward-chaining and “total materialization”
• In-memory reasoning and query evaluation
• OWLIM provides a reliable persistence, based on RDF N-Triples
• OWLIM can manage millions of statements on desktop hardware
• Extremely fast upload and query evaluation even for huge ontologies and knowledge bases
• OWLIM is developed by Ontotext

Semantic Channels: Implementations

Implementations – OWLIM

• OWLIM is available as a Storage and Inference Layer (SAIL) for Sesame RDF.

• Benefits:
  – Sesame’s infrastructure, documentation, user community, etc.
  – Support for multiple query language (RQL, RDQL, SeRQL)
  – Support for import and export formats (RDF/XML, N-Triples, N3)
Implementations – Jena

- Apache Jena™ is a Java framework for building Semantic Web applications.
- Jena provides a collection of tools and Java libraries to help you to develop semantic web and linked-data apps, tools and servers.
- The Jena Framework includes:
  - an API for reading, processing and writing RDF data in XML, N-triples and Turtle formats;
  - an ontology API for handling OWL and RDFS ontologies;
  - a rule-based inference engine for reasoning with RDF and OWL data sources;
  - stores to allow large numbers of RDF triples to be efficiently stored on disk;
  - a query engine compliant with the latest SPARQL specification
  - servers to allow RDF data to be published to other applications using a variety of protocols, including SPARQL.
Outline

1. Overview
2. Semantic Analysis (= Natural Language Processing)
3. Semantics as a channel (= Semantic vocabularies)
4. Semantic Content Modelling (=Ontologies)
5. Semantic Match Making (= Automatic distribution)
6. Summary

Social Media: upcoming challenges

- Social Networks are often compared to “Walled Gardens” [1]
- It becomes tremendously hard to share the right content with the right audience.
- Consumption is limited to few active channels but the image of a brand/company/individual is shaped by the community – and community does not happen in a single channel.

Semantic as End User Enabler

- Solve these obstacles by mechanizing important aspects of these tasks, and therefore offer a **scalable, cost-sensitive, and effective** online dissemination solution.

- Introduce a layer on top of the various internet based communication channels that is domain specific and **not** channel specific.

**Information model**
- defines the type of information items in the domain

**Weaver**
- mappings of information items to channels

**Channel model**
- describes the various channels, the interaction pattern, and their target groups

---

**Information model in action**

---

Same events
Separating Symbol and Knowledge Level

Analogy 1 (for senior people in the audience)

“I am about to propose the existence of something called the knowledge level, within which knowledge is to be defined.” [Newell, 1982]

- Knowledge is intimately linked with **rationality**. Systems of which rationality can be posited can be said to have knowledge.
- At the **knowledge level**, knowledge is described functionally in terms of goals and rationality.
- At the symbol level, knowledge is described operational in terms of achieving the goals through a certain sequence of activities.
- Obviously, there are various ways to encode knowledge at the symbol level.

Separating Content and Rendering

- Analogy 2 (for the juniors in the audience):
  - Content may be presented differently in different contexts.
  - Therefore, it should be modeled independent from a specific representation
  - Stylesheets connect content with a specific presentation
- Content:

```html
<html><head>
<link rel="stylesheet" type="text/css" href="tryit.css"/>
</head>
<body>
<div itemscope itemtype="http://schema.org/Person">
  <img src="http://www.fensel.com/dieter.jpg" itemprop="image" />
  <span itemprop="jobTitle">Prof. Dr.</span>
  <span itemprop="name">Dieter Fensel</span>
  <span itemprop="nationality">German</span>
  <span itemprop="birthdate">October 1960</span>
  <span itemprop="streetAddress">Technikerstr. 21a</span>,
  <span itemprop="postalCode">6020</span>,
  Innsbruck,
  Tirol,
  +43 512 507 6485,
  dieter.fensel@sti2.at,
  http://www.fensel.com/
</div></body></html>
```
Separating Content and Rendering

• Style Sheet 1:

```css
body {
  background-color: rgb(220,220,255);
  font-family: "Times New Roman";
  font-size: 20px;
}

img { float: right; }

span[id="property"]
{
  display: block;
  font-style: italic;
}

span[itemprop]
{
  font-weight: bold;
  font-style: normal;
}

a:link
{
  color: green;
  font-style: normal;
  font-weight: bold;
}
```

• Style Sheet 2:

```css
body {
  font-family: "Calibri";
  font-size: 25px;
}

img
{
  float: left;
  width: 120px;
  margin-right: 50px;
}

span[id="property"]
{
  margin-right: 40px;
  float: left;
}

span[itemprop] { font-style: italic; }

a:link
{
  font-style: italic;
  font-weight: bold;
}
```
Information model for organizations/projects
Information model for tourism

Using the information model for tourism

The hotel’s perspective:
Using the information model for tourism

The customer’s perspective:

Channel model

The channel model describes the different channels, their interaction patterns and the target groups.

- Channels can be
  - online or offline
  - for broadcasting or sharing
  - for group communication or collaboration
  - of static or dynamic information

- The number of different channels is growing constantly

- The target groups are very different from channel to channel
Offline and Online Channels

- Walk-in customer
- Telephone
- Email
- Fax
- Hotel website
- Review sites
- Booking sites
- Social network sites
- Blogs
- Fora & destination sites
- Chat
- Video & photo sharing

Weaver

Information Model

Branch specific concepts

Distribute content

Collect feedback + statistics

Channel Model

Web 3.0/Mobile/Other

Press releases + Ads

Web/Blog

Social Web

Typo3

myspace

Facebook

Twitter

LinkedIn

Flickr

XING

Google

jQuery

Web30
The **weaver** is responsible for mapping of information items to the appropriate channels.

- **Separation** of content and communication channels
- **Reuse** of the same content for various dissemination means
- **Explicit alignment** of information items and channels

---

**Weaver component**

**The details of the Weaver**

Elements 1 to 3 are about the content. They define the actual categories, the agent responsible for them, and the process of interacting with this agent.

1. **Information item**
   - It defines an information category that should be disseminated through various channels.

2. **Editor**
   - The editor defines the agent that is responsible for providing the content of an information item.

3. **Interaction protocol**
   - This defines the interaction protocol governing how an editor collects the content.
The details of the Weaver

Elements 4 to 9 are about the dissemination of these items.

4. **Information type**
   An instance of a concept, a set of instances of a concept (i.e., an extensional definition of the concept), or a concept description (i.e., an intentional definition of a concept).

5. **Processing rule**
   These rules govern how the content is processed to fit a channel. Often only a subset of the overall information item fits a certain channel.

6. **Channel**
   The media that is used to disseminate the information.

---

The details of the Weaver

7. **Scheduling information**
   Information on how often and in which intervals the dissemination will be performed which includes temporal constrains over multi-channel disseminations.

8. **Executor**
   It determines which agent or process is performing the update of a channel. Such an agent can be a human or a software solution.

9. **Executor interaction protocol**
   It governs the interaction protocol defining how an executor receives its content.
**Example: Weaver Storyline**

- The organization STI International has regular general assemblies
- A general assembly is described by the president
- The description refers to the general assembly as an event
- The description of a general assembly must refer to a future event
- The information is published on the content management system Drupal

---

**Example: Weaver Storyline**

This process forms a part of the weaver component:

- Item: sti2:general_assembly
- Editor: President
- Type: Concept Description
- Channel: homepage/event/past---event
- Schedule Constraint: date > current date
- Executor: Drupal
- Executor Interaction Protocol: none
Outline

1. Overview
2. Semantic Analysis (= Natural Language Processing)
3. Semantics as a channel (= Semantic vocabularies)
4. Semantic Content Modelling (= Ontologies)
5. Semantic Match Making (= Automatic distribution)
6. Summary

Semantic Match Making

- Branch specific concepts
- Distribute content
- Collect feedback + statistics
In a nutshell:

- Form concept groups through abstraction
- Use artificial intelligence to match new objects to one of the groups

Scientific aspects:

- Channels and content are matched automatically
- Texts are shortened/enlarged to an amount which fits the Blogpost/Tweet
- Pictures/Videos/Slides are attached where needed
Related Systems

- A Semantic Publish/Subscribe System for Selective Dissemination of the RSS Documents [1]

- Semantic Email Addressing: Sending Email to People, Not Strings [2]


A Semantic Publish/Subscribe System for Selective Dissemination of the RSS Documents

The presented system:

- System bases on the Semantic Web RDF and OWL standards and RDF

- Site Summaries (RSS) in order to introduce an efficient publish/subscribe mechanism that includes an event matching algorithm based on graph matching.


How do we relate to channels and content:

- Our approach, in contrast, matches information items rather than events to channels rather than users.

- Instead of graph matching, we use predefined weavers for channel selection
A Semantic Publish/Subscribe System for Selective Dissemination of the RSS Documents


Semantic Email Addressing: Sending Email to People, Not Strings

- SEA: Semantic E-Mail Addressing

- Main idea: send emails to concepts that dynamically change over time

- People describe their interests via foaf.

- Senders send emails to interest groups.

Semantic Email Addressing: Sending Email to People, Not Strings

Outline

1. Overview
2. Semantic Analysis (= Natural Language Processing)
3. Semantics as a channel (= Semantic vocabularies)
4. Semantic Content Modelling (=Ontologies)
5. Semantic Match Making (= Automatic distribution)
6. Summary
Summary

The four semantic pillars:

1. Semantic analysis
2. Semantic channels
3. Semantic content modelling
4. Semantic match making

Semantic Analysis

- Enables computers to interpret natural language

- A lot of different tasks (Topic detection, Named entity recognition, Sentiment detection, Opinion mining, etc.)

- Key task in analyzing content and automatically taking decisions about it.
Semantic Channels

• Distinguish between
  – languages (such as SPARQL, RDF, OWL, RDFa, Microformats, etc.)
  – vocabularies (such as GoodRelations, Schema.org, FOAF, SIOC, etc.)
  – Implementations (such as OWLIM,…)

• Dissemination in the right channels gains potential (see Google’s rich snippets)

Semantic Content Modelling

• Content modelling has to be shaped to individual domains.

• Content modelling only has to be done once for domains.

• Content is therefore reusable.

• A weaver brings channels and content together
Semantic Match Making

- Channels and content are matched automatically

- Texts are shortened/enlarged to an amount which fits the Blogpost/Tweet

- Pictures/Videos/Slides are attached where needed