An Overview and Underview of the Semantic Web

Tim Finin
University of Maryland
Baltimore County

Overview
- The Problem: building intelligent information systems
- The Semantic web as part of the solution
- Background on the semantic web
- Comments and Conclusions

The problem
- I’ve been engaged in research aimed at developing intelligent information systems for thirty years.
- The problem is hard, progress is slow, but the incremental results are worth it.
- It’s a task for many generations to come.
- Today’s environment is very different than that in 1972.

They way we were...
They way we will be...

## What’s new?
- **Internet.** Virtually of the computers in the world have been connected.
- **Scale.** Every day many more computing and communication devices are joining.
- **Power.** Raw computing power continues to climb.
- **Wireless.** New technologies (GSM, 802.11, Bluetooth, UWB?, IR, etc) are creating a pervasive, ubiquitous computing environment.
- **Web.** Anyone can publish content and provide services, powerful search engines support discovery, evolving standards enhance interoperability.

The way we will be...

People, agents, devices, & services need to
- Find others in their environment
- Describe the services they offer and seek
- Exchange APIs
- Negotiate for services, permissions, privacy, payment, ...
- Reason about services to create composite services
- Coordinate and cooperate as needed
- Sense their context and the activities of humans
- Deal with new entities never before encountered

And to do this dynamically

Information and Data Management Challenges

The environment makes new demands and offers new challenges, enough to keep all of us busy, e.g.:
- Very open environments
- Large and diverse community of service and content providers
- Lots of relative autonomy
- Dynamic ad hoc networks
- Systems with widely varying resources -- bandwidth, connectivity, cpu, memory, disk, power, software, knowledge, intelligence, etc.
Research topics

Concepts addressing these challenges include:

- Multiagent systems
- Information and knowledge sharing through common representation languages, ontologies and protocols
- Automatic service description, discovery, composition
- Negotiation for services and information
- Trust based models for authorization, credibility and security
- Social and norm governed behavior
- Delegation and degrees of autonomy
- Coordination and teamwork models

Semantic Web

- I’ll argue that the semantic web provides a good approach, language and tools to support the development of intelligent information systems in this environment.
- This isn’t obvious, since the SW seems grounded in the “traditional” hypertext on the wired web.
- But, the principles which drive it are the right ones for agents as well as pervasive computing.
- And, by grounding agents in web technology, they may make it out of the lab.
- Next: overview of Semantic Web

W3C’s Semantic Web Goals

- Focus on machine consumption:
  "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.”
- Whereas the Web has made people smarter, the SW will make machines smarter.
- The current Web stores things whereas the SW enables agents which do things.

Origins of the Semantic Web

Capsule history

- Tim Berners-Lee proposed WWW as a Web of relationships among named objects (89)
- Guha designed MCF (~94)
- XML+MCF= RDF (~96)
- RDF+OO= RDFS (~99)
- RDFS+KR= DAML+OIL (00)
- W3C’s SW activity (01)
- W3C’s OWL (02?)

http://www.w3.org/History/1989/proposal.html
Semantic Web does what?

- Concept-based search ≠ keyword-based search
- Semantic navigation ≠ link-based navigation
- Personalization ≠ one size fits all
- Query answering ≠ document retrieval
- Services ≠ CGI calls, but service-description languages, negotiation, service composition, etc

Why is this hard?

After Frank van Harmelen and Jim Hendler

What a web page looks like to a machine...

And understanding natural language is easier than images!

"Webscraping" is mostly done by hand crafted rules or rules generated by supervised learning

Either way, the rules can break when the page structure changes.

OK, so HTML is not helpful

Could we tell the machine what the different parts of the text represent?

title
speaker
time
location
abstract
biosketch
host

After Frank van Harmelen and Jim Hendler
XML to the rescue?

XML fans propose creating a XML tag set to use for each application. For talks, we can choose <title>, <speaker>, etc.

after Frank van Harmelen and Jim Hendler

XML ≠ machine accessible meaning

But, to your machine, the tags still look like this….

The tag names carry no meaning. XML DTDs and Schemas have little or no semantics.

after Frank van Harmelen and Jim Hendler

XML Schema helps

XML Schemas provide a simple mechanism to define shared vocabularies.

XML Schema file

XML Schema file 1

XML Schema file 42

But there are many schemas

after Frank van Harmelen and Jim Hendler
There's no way to relate schema:

XML Schema file 1

XML Schema file 42

Either manually or automatically -- XML Schema is very weak on semantics.

Ontologies can help ...

- An ontology defines the terms used to describe and represent an area of knowledge.
- Ontologies are used by people, databases, and applications that need to share domain information (a domain is just a specific subject area or area of knowledge, like medicine, tool manufacturing, real estate, automobile repair, financial management, etc.). Ontologies include computer-usable definitions of basic concepts in the domain and the relationships among them ...
- They encode knowledge in a domain and also knowledge that spans domains.

In this way, they make that knowledge reusable.

Ontologies can help ...

- Structure
- Constraint
- Sharing

We need a way to define ontologies in XML.

So we can relate them
So machines can understand (to some degree) their meaning.

An Ontology level is needed
Ontologies vary...

- Ontologies vary greatly in their
  - Scope
  - Complexity
  - Level of detail
  - Kind of knowledge encoded
- Two examples...

Dublin Core -- A Simple Ontology

- Developed by an OCLC sponsored workshop in Dublin ~'95 as a standard for metadata for digital library resources on web
  - Consists of 15 core attributes
    - http://dublincore.org/
  - Neutral on how DC should be represented
  - HTML found to be inadequate for representing complexities of structured use of DC
  - Available as an RDF schema.

Cyc -- a complex ontology

- Cyc is a large, general purpose ontology with associated reasoning tools developed over the past ~20 years by MCC and now Cycorp
  - Cyc KB has > 100k terms.
  - Terms are axiomatized by > 1M handcrafted assertions
  - Cyc inference engine has > 500 heuristic level modules
  - Goal is to encode knowledge for "common sense reasoning" needed by applications (e.g., NLP)
  - Available free in limited form from http://opencyc.org/

Today and tomorrow

- We are in a good position to use simple ontologies like DC today
  - This is happening (e.g., Adobe's XMP)
- We hope to be able to make effective use ontologies like Cyc in the coming decade
  - There are skeptics...
  - It's a great research topic...
"The Semantic Web will globalization KR, just as the WWW globalization hypertext" – Tim Berners-Lee

TBL’s semantic web vision

Semantic web languages today

- Today there are, IMHO, two semantic web languages
  - DAML+OIL – Darpa Agent Markup Language
    http://www.daml.org/
  - RDF – Resource Description Framework
    http://www.w3.org/RDF/
- and one under development by the W3C
  - OWL – Ontology Web Language
    http://www.w3.org/2001/sw/
- Topic maps (http://topicmaps.org/) are another breed
- with more to come….

Topic Maps

- A Topic Map is a collection of topics and (semantically meaningful) relationships between these topics
- Topic Maps link these topics with external references, such as resources behind URIs
- Topic Maps are a “superimposed semantic layer”
  - connection between topics and resources are URLs
- Topic Maps capture real-world subjects/objects but also concepts
  - these are defined not absolute but relative to each other

RDF is the first SW language

- RDF is the first SW language
**Simple RDF Example**

```
http://umbc.edu/~finin/talks/idm02/
dc:Title  "Intelligent Information Systems on the Web and in the Aether"
dc:Creator  bib:Name "Tim Finin" bib:Email "finin@umbc.edu"
```

---

**N triple representation**

- RDF can be encoded as a set of **triples**.
  - `<subject> <predicate> <object>`.

```
<http://umbc.edu/~finin/talks/idm02/> <http://purl.org/dc/elements/1.1/Title> "Intelligent Information Systems on the Web and in the Aether".
_:j10949 <http://daml.umbc.edu/ontologies/bib/Name> "Tim Finin".
_:j10949 <http://daml.umbc.edu/ontologies/bib/Email> "finin@umbc.edu".
_:j10949 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <Description>.
```

- Note the gensym for the anonymous node (`_:j10949`).

---

**XML encoding for RDF**

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:bib="http://daml.umbc.edu/ontologies/bib/">
  <description about="http://umbc.edu/~finin/talks/idm02/">
    <dc:title>Intelligent Information Systems on the Web and in the Aether</dc:title>
    <dc:creator>
      <description>
        <bib:Name>Tim Finin</bib:Name>
        <bib:Email>finin@umbc.edu</bib:Email>
        <bib:Aff resource="http://umbc.edu/"/>
      </description>
    </dc:Creator>
  </description>
</rdf:RDF>
```

---

**Triple Notes**

- RDF triples have one of two forms:
  - `<subject> <predicate> <object>`
  - `<subject> <predicate> <quoted string>`
- Triples are also easily mapped into logic:
  - `<subject> <predicate> <object>`
- `<predicate>(<subject>,<object>)`
- With type `<S>,<O>` becoming `<O>(<S>)`
- Examples:
  - `subclass(man,person)`  ; Note: we're not showing the actual URLS for clarity
  - `sex(man,male)`
  - `domain(sex,animal)`
  - `man(adam)`
  - `age(adam,100)`
- Triples are easily stored and managed in a DBMS.
N3 notation for RDF

N3 is a compact notation for triples which is easier for people to read and edit.

Example:
```
@prefix log: <http://www.w3.org/2000/10/swap/log#> .
:Person a rdfs:Class.
:Woman a rdfs:Class; rdfs:subClassOf :Person.
:Eve a :Woman; age "100".
:sister a rdf:Property.
```

RDF Schema (RDFS)

- RDF Schema adds taxonomies for classes & properties
  - subclass and subProperty
  - and some metadata.
  - domain and range constraints on properties
- Several widely used KB tools can import and export in RDFS

Stanford Protégé KB editor
- Java, open sourced
- extensible, lots of plug-ins
- provides reasoning & server capabilities

RDFS supports simple inferences

- An RDF ontology plus some RDF statements may imply additional RDF statements.
- This is not true of XML.
- Example:
  ```
  domain(parent,person) range(parent,person)
  subproperty(mother,parent) range(mother,woman)
  mother(eve,cain)
  ```
- This is part of the data model and not of the accessing/processing code
- Implies:
  ```
  subclass(woman,person) parent(eve,cain) person(eve) person(cain) woman(eve)
  ```

RDF is being already in use

- RDF has a solid specification
  - See the RDF model theory spec - http://www.w3.org/TR/rdf-mt/
- RDF is being used in a number of W3C specifications
  - CC/PP (Composite Capabilities/Preference Profiles) http://www.w3.org/Mobile/CCPP/
  - P3P (Platform for Privacy Preferences Project) http://www.w3.org/P3P/
- And in other web standards
  - RSS 1.0 (RDF Site Summary)
  - RDF calendar (~ iCalendar in RDF)
- And in other systems
  - Netscape's Mozilla web browser
  - Open directory (http://dmoz.org/)
  - Adobe products via XMP (eXtensible Metadata Platform)
RDF is not enough, but is a good foundation
- RDF lacks expressive adequacy for many tasks
  - Only range/domain constraints (on properties)
  - No properties of properties (transitive, inverse etc.)
  - No equivalence, disjointness, coverings, etc.
  - No necessary and sufficient conditions
  - No rules, axioms, logical constraints
- DAML+OIL extends RDF
  - Layering makes partial knowledge available to applications which only understand RDF
  - NB: Building on RDF has some disadvantages

We’re going down a familiar road

KR trends
- 55-65: arbitrary data structures
- 65-75: semantic networks
- 75-85: simple frame systems
- 85-95: description logics
- 95-?: logic?, rules?

Web trends
- 95-97: XML as arbitrary structures
- 97-98: RDF
- 98-99: RDFS (schema) as a frame-like system
- 00-01: DAML+OIL
- 02-?: OWL, ...??...

Only much faster!

DAML+OIL as a Semantic Web Language
- DAML = Darpa Agent Markup Language
  - DARPA program with 17 projects & an integrator developing language spec, tools, applications for SW.
- OIL = Ontology Inference Layer
  - An EU effort aimed at developing a layered approach to representing knowledge on the web.
- Process
  - Joint Committee: US DAML and EU Semantic Web Technologies participants
  - DAML+OIL specs released 01/01 & 03/01
  - See http://www.daml.org/
  - Includes model theoretic and axiomatic semantics

A Simple DAML Example

```xml
<rdfs:Class about="#Animal"/>
<rdfs:Class about="#Plant">
  <daml:disjointFrom resource="#Animal"/>
</rdfs:Class>
```

- Note the mixture of RDF (plant & animal are classes) and DAML (plant & animal are disjoint)
- If your cell phone only does RDF, it still understands some of this...
DAML in One Slide

DAML is built on top of XML and RDF

It allows the definition, sharing, composition and use of ontologies

DAML is ~= a frame based knowledge representation language

It can be used to add metadata about anything which has a URI

URLs are a W3C standard generalizing URLs

everything has URI

DAML-S is an ontology for describing properties and capabilities of web services

Desiderata:
- Ease of expressiveness
- Enables automation of service use by agents
- Enables reasoning about service properties and capabilities
- Also appropriate for describing services in a mobile/pervasive computing environment

See http://daml.org/services/

DAML Components

Service profile (what it does)
- For service registration, discovery and matching
  - High level description of service and provider with a (human readable) description of service, a specification of functionality provided and other functional attributes.
  - Functional properties support composition, inputs, outputs, pre conditions and effects.

Service model (how it works)
- For service invocation, composition, interoperation, monitoring, ...
  - Composite processes are build using sequence, if-then-else, fork, etc.

Service grounding (how to access)
- Specification of service access information (communication protocols, transport mechanisms, etc.) which could be via SOAP, HTTP forms, Java RMI, RPC, etc.

UMBC
The WOWG is working on a recommendation for the "Web Ontology Language", **OWL**.

- **56 Members from 30 W3C Organizations**
  - **Companies**: Agfa, Daimler-Chrysler, EDS, Fujitsu, Hewlett-Packard, IBM, Intel, IVIS, Lucent, Network Inference, Nisus, Nokia, Philips, Sito, Sun, Unisys
  - **Public Sector**: DISA, Electricite de France, Intelink, INTAP, MITRE, NIST
  - **Research projects/Labs**: DFKI, FZI, Ibrow group, Stanford, U. Bristol, U. Maryland, U. Southampton
  - **Invited Experts**: Medical, Digital Library, Defense, Technical
- **CoChairs**: Jim Hendler, University of Maryland/MIND; Guus Schreiber, Univ of Amsterdam/Ibrow
- **http://www.w3.org/2001/sw/WebOnt/**

**OWL Goals**

- The WOWG has identified the following goals in developing OWL:
  - Shared ontologies
  - Ontology evolution
  - Ontology interoperability
  - Inconsistency detection
  - Balance of expressivity and scalability
  - Ease of use
  - XML syntax
  - Internationalization

**OWL status and publications**

- **OWL is roughly equivalent to DAML with some renaming of properties**
- **Current plan is to have three compliance levels**: OWL lite, OWL, OWL plus
- **WebOnt has published**
  - **Requirements** for a Web Ontology Language
  - **Feature Synopsis** for OWL Lite and OWL
  - **OWL Web Ontology Language 1.0 Reference**
  - **OWL Web Ontology Language 1.0 Abstract Syntax**
  - (forthcoming) **OWL Guide**
**OWL Lite**

**RDF Schema Features:**
- Class
- rdf:Property
- rdfs:subClassOf
- rdfs:subPropertyOf
- rdfs:domain
- rdfs:range
- Individual

**Equality properties:**
- sameClassAs
- samePropertyAs
- sameIndividualAs
- differentIndividualFrom

**Properties of properties:**
- inverseOf
- transitiveProperty
- symmetricProperty
- functionalProperty
- inverseFunctionalProperty
- allValuesFrom
- someValuesFrom
- minCardinality (0/1)
- maxCardinality (0/1)
- cardinality (0/1)

**Header Information:**
- imports
- Dublin Core Metadata
- versionInfo

**Motivation:**
- easier to implement and to learn, lower reasoning complexity

**Missing:**
- enumerated classes, disjointness, unionOf, intersectionOf, complementOf, full cardinality, ...

---

**KR meets the Web**

- One way to think about the semantic web is that we are creating a knowledge representation language for the Web.
- This is more than just selecting an appropriate KR language and selecting an XML encoding.
- The Web as an information system has many significant properties.
  - Highly distributed
  - Subject to disconnections and other failures
  - Many content providers
  - Partial and inconsistent information
  - Not all info and services can be trusted
  - Dynamic
  - Evolving

---

**Semantic Web Principles**

- Everything is on the web
  - People, places, times, things all have URIs
- Partial information is assumed
  - The web privileges scalability over integrity and there's always more and new stuff to find
- Trust models are critical
  - It's not all true
- Support information evolution
  - Content and consensus is dynamic
- Minimalist design
  - Make the simple things simple, and the complex things possible. Standardize no more than is necessary.
- Common data model
  - To support interoperability and knowledge sharing

---

**SW is work in progress**

- There are important language aspects which need more work: rules, queries, etc.
- Many tools need to be created, e.g.,
  - Protégé plug-in for DAML+OIL
  - Annotation tools
- Applications need to be explored
- The W3C is developing a new SW language
  - OWL: Ontology Web Language
- SW ideas will migrate into other standards (e.g., basic XML, WSDL, .NET)
**DAML+OIL usage**

- DAML+OIL is already the most used ontology/KR language in history
  - Daml.org ~ 5.4M hits; avg. 24,300/day in Oct 02.
  - 1.8 x 10^15 (180,000 Gb) downloaded
  - Oct 16: Crawler finds 5.9M DAML statements on 20K web pages
- Doesn't include many instance KBs tied to ontologies or many very large RDFS-based KBs that include some OWL
- OWL is moving it towards the commercial world
  - Web tool developer labs: IBM, HP, Sun, Intel, Fujitsu
  - Content providers/users: Daimler-Chrysler, Nokia, Motorola, EDS, Agfa
- Starting to be noticed by thesaurus distributors -- e.g., National Cancer Institute metathesaurus to be released in OWL

**Lots of Open Issues**

- How expressive should the KR language be?
- What kind of KR/reasoning system
  - F.O. logic, fuzzy...
- On Web Ontologies
  - One (e.g. CYC) or many (DAML)
  - If many, composable (IEEE IFF) or monolithic (IEEE SUMO)
  - Will general “upper ontologies” (e.g., IEEE SUO) be useful?
- Will industry buy in?
  - Or continue to explore ad hoc XML based solutions
- How will it be used?
  - As markup? As alternative content? Just both machines and people?
  => Only experimentation will yield answers.

**Conclusions and final thoughts**

- SW might be a chance for us to get some AI out of the lab
- Solving the symbol grounding problem
  - Rethinking agent communication
  - How do we get there

**The symbol grounding problem**

- An argument against human-like AI is that it's impossible unless machines share our perception of the world.
- A solution to this “symbol grounding problem” is to give robots with human inspired senses.
- But the world we experience is determined by our senses, and human and machine bodies may lead to different conceptions of the world (e.g. Nagel's *What Is It Like To Be a Bat?*).
- Maybe the Semantic Web is a way out of this problem?
Solving the symbol grounding problem

- The web may become a common world that both humans and machines can understand.
- Confession: the web is more familiar and real to me than much of the real world.
- Physical objects can be tagged with low cost (e.g., $0.05) transponders or RFIDs encoding their URIs
  - See HP’s Cooltown project
  http://cooltown.com/

Rethinking the agent communication paradigm

- Much multi-agent systems work is grounded in Agent Communication Languages (e.g., KQML, FIPA) and associated software infrastructure.
  - This paradigm was articulated ~1990, about the same time as the WWW was developed.
  - Our MAS approach has not yet left the laboratory yet the Web has changed the world.
  - Maybe we should try something different?
    - The communication MAS paradigm has been peer-to-peer message oriented communication mediated by brokers and facilitators -- an approach inherited from client-server systems.

Rethinking the agent communication paradigm

- A possible new paradigm?
  - Agents “publish” beliefs, requests, and other “speech acts” on web pages.
  - Brokers “search” for and “index” published content
  - Agents “discover” what peers have published on the web and browse for more details
  - Agents “speak for” content on web pages by
    - Answering queries about them
    - Accepting comments and assertions about them

How do we get there from here?

- This semantic web emphasizes ontologies – their development, use, mediation, evolution, etc.
- It will take some time to really deliver on the agent paradigm, either on the Internet or in a pervasive computing environment.
- The development of complex systems is basically an evolutionary process.
- Random search carried out by tens of thousands of researchers, developers and graduate students.
"The sheer height of the peak doesn’t matter as long as you don’t try to scale it in a single bound. Locate the middle step on path and, if you have unlimited time, the ascent is only as formidable as the next step.” — Richard Dawkins, Climbing Mount Improbable, Penguin Books, 1996.

The Evolution of Useful Things

- Prior to the 1890’s, papers were held together with straight pens.
- The development of “spring steel” allowed the invention of the paper clip in 1899.
- It took about 25 years (!) for the evolution of the modern “gem paperclip”, considered to be optimal for general use.

So, we should …

- Start with the simple and move toward the complex
  E.g., from vocabularies to FOL theories
- Develop new capabilities
  E.g., rules, trust, negotiation, automatic markup, …
- Allow many ontologies to bloom
  Let natural evolutionary processes select consensus ontologies.
- Support diversity in ontologies
  Monocultures are unstable, there should be no "The ontology for X".
- The evolution of powerful, machine readable ontologies will take many years, maybe generations
  But incremental benefits will easily justify the effort

For more information

- RDF
  http://www.w3.org/RDF/
- DAML+OIL
  http://www.daml.org/
- OWL & W3C’s semantic web activity
  http://www.w3.org/2001/sw/
- Semantic web links
  http://semanticweb.org/
- Next Semantic Web Conference
  http://iswc.semanticweb.org/
  October 2003, Sanibel Island, SC.