The world must learn to work together, or finally it will not work at all......

---- Dwight D. Eisenhower

W3C's Semantic Web standards

WEB 2.0?
WEB 3.0?
HOW'S A GIRL TO STAY AHEAD?
Robert H.P. Engels

• Born in Amsterdam (1968)
• Studied at University of Amsterdam, Stockholm and Karlsruhe. MSc in Psy, PhD in AI & ML
• ML systems, Econometric Systems, KM systems, Computational Linguistics, Semantic Web
• Company owner, Products, EU expert, courses, consultancy etc.
• Also builds motorcycles ;-)
• robert.engels@esis.no / 99544481
Why are we here today?

- Discuss the (im-)possibilities of ISO15926 and possibilities for its implementation in KRL's with a strong(er) semantics
  - Motivate universities/research institutions in Norway to increase focus on semantics and ontologies
  - Strengthen the relation between universities/research institutions and the oil & gas industry
  - Create an academic network on ontology within research and teaching
Spectrum of KR and Reasoning

Source: Dr. Leo Obrst, Mitre; Mills Davis, Project10X
“Semantic” standards to choose from...

- Unified Modeling Language (UML)
  - well known and often used standard

- UN/CEFACT Core Components
  - context independent: Core Component Technical Specification (CCTS)
  - context specific: Business Information Entities

- ISO Topic Maps
  - weak semantics, drawback in reasoning and wrt interoperability

- ISO 15926
  - extensive metamodel, predefines much of domain specific vocabulary

- W3C RDF/RDFSchema/OWL
  - well defined semantics, several flavours
  - widespread use on internet
  - enabler for data interoperability
  - complexity problems in specific problem domains
Implications of choosing

• Is support for the standard global?
  » amateur content, professional content, sharing of digital assets

• Does the standard have enough international momentum?
  » availability of human resources, development, tool support?
  » pricing/licences – open source tools and standards?

• Does the standard support your needs for digital publication?
  » can you actually represent what you need and use it in the ways you need?
  » human readability (do you really facilitate people?)
  » machine interoperability (can intelligent programs automatically interact with your information? can you automatically import information from other places?)
  » will it be embedded in your digital formats?

• Does one need to choose at all?
  » what is the choice actually?
  » possibility for a heterogenous environment?
Semantic Web (W3C)

• W3C
  – Founded by Tim Berners Lee (the “initiator” of the Internet) in 1994
  – Creates Web Standards and Guidelines
  – Involved in education, outreach and software development
  – Started the Web (1991) and coined the Semantic Web (1999)
  – Coordinated by MIT (USA), European Research Centre (ERCIM, France) and Keio University (Japan) + World Offices all around the globe

• Semantic Web standard and technology
  – Set of standards for the “next generation in Internet” (1999)
  – Query Language for RDF (SPARQL – W3C recommendation)
  – Web Ontology Language (OWL – W3C recommendation)
  – RDF Schema (ontology definition)
  – Resource Description Framework
Current Status
Semantic Web
Under the hood

A little help from Ivan Herman's intro course to the semantic web
RDF triples (cont.)

- An RDF Triple \((s, p, o)\) is such that:
  - “s”, “p” are URI-s, ie, resources on the Web; “o” is a URI or a literal
    - “s”, “p”, and “o” stand for “subject”, “property”, and “object”
  - here is the complete triple:

\[
<\text{http://...isbn...6682}>, \ <\text{http://.../original}>, \ <\text{http://...isbn...409X}>
\]

- **RDF** is a general model for such triples (with machine readable formats like RDF/XML, Turtle, N3, RXR, …)
A simple RDF example (in Turtle)

```
<http://.../isbn/2020386682>
  f:titre "Le palais des mirroirs"@fr ;
  f:original <http://.../isbn/000651409X> .
```
“Internal” nodes

• Consider the following statement:
  – “the publisher is a «thing» that has a name and an address”
• nodes were identified with a URI. But…
• …what is the URI of «thing»?
• Use the concept of blank nodes
  » but be carefull when merging
Semantic Web stack
Need for RDF schemas

• We need “extra knowledge”, so let's:
  – define the terms we can use
  – what restrictions apply
  – what extra relationships are there?

• This is where RDF Schemas come in
  – officially: “RDF Vocabulary Description Language”; the term “Schema” is retained for historical reasons…
Classes, resources, … (cont.)

- Relationships are defined among classes/resources:
  - “typing”: an individual belongs to a specific class
    - “«The Glass Palace» is a novel”
    - to be more precise: “«http://.../000651409X» is a novel”
  - “subclassing”: all instances of one are also the instances of the other (“every novel is a fiction”)
- **RDFS formalizes these notions in RDF**
RDFS defines the meaning of these terms
  - (these are all special URI-s, we just use the namespace abbreviation)
Current Status

- User Interface & applications
- Trust
- Proof
- Unifying Logic
- Query: SPARQL
- ontology: OWL
- Rules: RIF
- RDF-S
- Data interchange: RDF
- XML
- URI
- Unicode
- Crypto
What can we do with RDF/S?

Interoperability
Interchange
Sharing

An example
The rough structure of data integration

1. Map the various data onto an **abstract data representation**  
   – make the data independent of its internal representation…

2. **Merge** the resulting representations

3. **Start querying** on the whole!  
   – queries not possible on the individual data sets
**A simplified bookstore data**
(dataset “A”)

<table>
<thead>
<tr>
<th>KEYFIELD: ID</th>
<th>Author</th>
<th>Title</th>
<th>Publisher</th>
<th>Year</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Home Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>id_xyz</td>
<td>Ghosh, Amitav</td>
<td><a href="http://www.amitavghosh.com">http://www.amitavghosh.com</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ID</th>
<th>Publ. Name</th>
<th>City</th>
</tr>
</thead>
<tbody>
<tr>
<td>id_qpr</td>
<td>Harpers Collins</td>
<td>London</td>
</tr>
</tbody>
</table>
1st: export your data as a set of **relations**
Some notes on the exporting the data

- Data export does *not* necessarily mean physical conversion of the data
  - relations can be generated on-the-fly at query time
    - via SQL “bridges”
    - scraping HTML pages
    - extracting data from Excel sheets
    - etc.

- One can export *part* of the data
Another bookstore data
(dataset “F”)

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ID</td>
<td>Titre</td>
<td>Auteur</td>
<td>Traducteur</td>
<td>Original</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>Nom</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>Ghosh, Amitav</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>Besse, Christianne</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
2nd: export your second set of data
3rd: start merging your data
3rd: start merging your data

Same URI = Same Resources
**3rd**: merge identical resources
Start making queries…

• User of data “F” can now ask queries like:
  – “give me the title of the original”
• This information is not in the dataset “F”…
• …but can be retrieved by merging with dataset “A”!
However, more can be achieved...

- We “feel” that `a:author` and `f:auteur` should be the same
- But an automatic merge does not know that!
- Let us add some extra information to the merged data:
  - `a:author` same as `f:auteur`
  - both identify a “Person”
  - a term that a community may have already defined:
    - a “Person” is uniquely identified by his/her name and, say, homepage
    - it can be used as a “category” for certain type of resources
3rd revisited: use the extra knowledge
Start making richer queries!

• User of dataset “F” can now query:
  – “give me the home page of the original’s author”
• The information is not in datasets “F” or “A”…
• …but was made available by:
  – merging datasets “A” and datasets “F”
  – adding three simple extra statements as an extra “glue”
Combine with different datasets

• Via, e.g., the “Person”, the dataset can be combined with other sources

• For example, data in Wikipedia can be extracted using dedicated tools
Merge with Wikipedia data
Merge with Wikipedia data

KEYWORD: SPARQL
SPARQL as a unifying point

Diagram:
- SPARQL Endpoint
- Triple Store
- Data in RDF
- SPARQL Processor
- Application
- SQL = SPARQL
- GRDDL
- RDFa
- XTIONL
- Unstructured Text
- XML
So now we have:
Ontologies

- RDFS is useful, but does not solve all possible requirements
- Complex applications may want more possibilities:
  - characterization of properties
  - identification of objects with different URI-s
  - disjointness or equivalence of classes
  - construct classes, not only name them
  - more complex classification schemes
  - can a program reason about some terms? E.g.:
    - “if «Person» resources «A» and «B» have the same «foaf:email» property, then «A» and «B» are identical”
  - etc.
$\textbf{Ontologies (cont.)}$

- The term \textit{ontologies} is used in this respect:

$\textit{"defines the concepts and relationships used to describe and represent an area of knowledge"}$

- I.e., there is a need for Web Ontology Language(s)
  - RDFS can be considered as a simple ontology language
- Languages should be a compromise between
  - rich semantics for meaningful applications
  - feasibility, implementability
Web Ontology Language = OWL

• OWL is an extra layer, a bit like RDF Schemas
  – own namespace, own terms
  – it relies on RDF Schemas
• It is a separate recommendation
• There is an active W3C Working Group working on extensions of the current standards
  – the new version will be called “OWL 2”
  – in what follows, some features will be referred to as “may come in future”, i.e., under consideration by that group
Term equivalence

- For classes:
  - `owl:equivalentClass`: two classes have the same individuals
  - `owl:disjointWith`: no individuals in common

- For properties:
  - `owl:equivalentProperty`
    - remember the `a:author` vs. `f:auteur`?

- For individuals:
  - `owl:sameAs`: two URIs refer to the same concept (“individual”)
  - `owl:differentFrom`: negation of `owl:sameAs`
Typical usage of `owl:sameAs`

- Linking Kolkata from one data set (DBpedia) to another (Geonames):

  `<http://dbpedia.org/resource/Kolkata> owl:sameAs <http://sws.geonames.org/1275004/>;`

- This is the main mechanism of “Linking” in the Linking Open Data project
Property characterization

- In OWL, one can characterize the behaviour of properties (symmetric, transitive, functional, inverse functional…)
- OWL also separates *data* and *object* properties
  - “datatype property” means that its range are typed literals
Characterization example

- "foaf:email" is inverse functional (i.e., two different subjects cannot have identical objects)
What this means is…

• If the following holds in our triples:

:email rdf:type owl:InverseFunctionalProperty.
<A> :email "mailto:a@b.c".
<B> :email "mailto:a@b.c".

• then the following holds, too:

<A> owl:sameAs <B>.

• I.e., *new relationships* were discovered again (beyond what RDFS could do)
Other property characterizations

- Functional property ("owl:FunctionalProperty")
- Transitive property ("owl:TransitiveProperty")
- Symmetric property ("owl:SymmetricProperty")
- Inverse of another property ("owl:inverseOf")
- May come in future:
  - reflexive and irreflexive object properties
  - specify that properties are “disjoint”
Classes in OWL

• In RDFS, you can subclass existing classes… that’s all
• In OWL, you can \textit{construct} classes from existing ones:
  – enumerate its content
  – through intersection, union, complement
  – etc
• OWL makes a stronger distinction between \textit{classes} and \textit{individuals}
  – referring to its own \textit{Class} and to “\textit{Thing}”, respectively
    • of course, \texttt{owl:Class} is a subclass of \texttt{rdfs:Class}, i.e., it is a refinement
OWL classes can be “enumerated”

- The OWL solution, where possible content is explicitly listed:
Union of classes

- Essentially, like a set-theoretical union:
What we have so far…

• The OWL features listed so far are already fairly powerful
• E.g., various databases can be linked via `owl:sameAs`, functional or inverse functional properties, etc.
• It is still possible to find all inferred relationship using a traditional rule engine
  – (more or less… there are some restrictions on details)
However… that may not be enough

• Very large vocabularies might require even more complex features
  – typical example: definition of all concepts in a health care environment

• One major issue is the way classes (i.e., “concepts”) are defined

• OWL includes those extra features but… the inference engines become (much) more complex
The term OWL “profiles” comes to the fore:

- restricting *which* terms can be used and *under what circumstances (restrictions)*
- if one abides to those restrictions, then simpler inference engines can be used
• In the **current OWL standard**, three such “profiles” are defined:
  – **OWL Full**: no restrictions whatsoever
  – **OWL DL**
  – (and its “sub profile” **OWL Lite**): major restrictions to ensure implementability

• The **OWL 2** work will add new profiles
  – profiles that are simple enough to be implementable with simple rule engines (like the first few examples we had)
  – profiles that are optimized to a small number of class and property definition but a large amount of data
  – etc.
OWL Full

- No constraints on the various constructs
  - `owl:Class` is equivalent to `rdfs:Class`
  - `owl:Thing` is equivalent to `rdfs:Resource`
  - This means that:
    - Class can also be an individual, a URI can denote a property as well as a Class
      - e.g., it is possible to talk about class of classes, etc.
    - One can make statements on RDFS constructs (e.g., declare `rdf:type` to be functional...)
    - etc.
- But: *an OWL Full ontology may be undecidable!*
Note on OWL profiles

- OWL profiles are defined to reflect compromises:
  - expressibility vs. implementability
- Some application just need to express and interchange terms (with possible scruffiness): OWL Full is fine
  - they may build application-specific reasoning instead of using a general one
- Some applications need rigour, but only a simple set of statements: a rule engine based profile might be o.k.
- Some applications need rigour and complex term classification; then OWL DL/Lite might be the right choice
Some of the issues we will not manage to talk about

- GRDDL
- RDFa
- OWL2
- QCR (Qual.Card.Restr.)
- POWDER
- N3/Turtle
- SKOS
- RIF

- Temporal Logic!!
- Trust, Proof
- Linking Open Data Project
- Uncertainty & Probabilistic approaches
- SWBP
**Main Message**

*Publication (Web 1.0), Participation & Interaction (Web 2.0) and **Interoperability** (Web 3.0) are key!***
A 14 year's old map of the world
- Sep 2008
So, how are you too choose?

Robert Engels
ESIS Norge AS
Vestlandsforsking

+47 99544481
robert.engels@esis.no

with many thanks to Ivan Herman (W3C) for letting me use slides from his tutorial