Languages and Semantic Web Architecture

- The Semantic Web Tower
  - what is the semantic web
- Problems Layering the Semantic Web
  - The problem in detail and suggested approaches

Øyvind Evensen

What is the semantic web

- Semantic web: "an extended web of machine-readable information and automated services that extend far beyond current capabilities".
  - knowledge based web
  - automated services
  - extremely knowledgeable system that futures specialized reasoning services to support us in our daily life

Semantic Web Tower

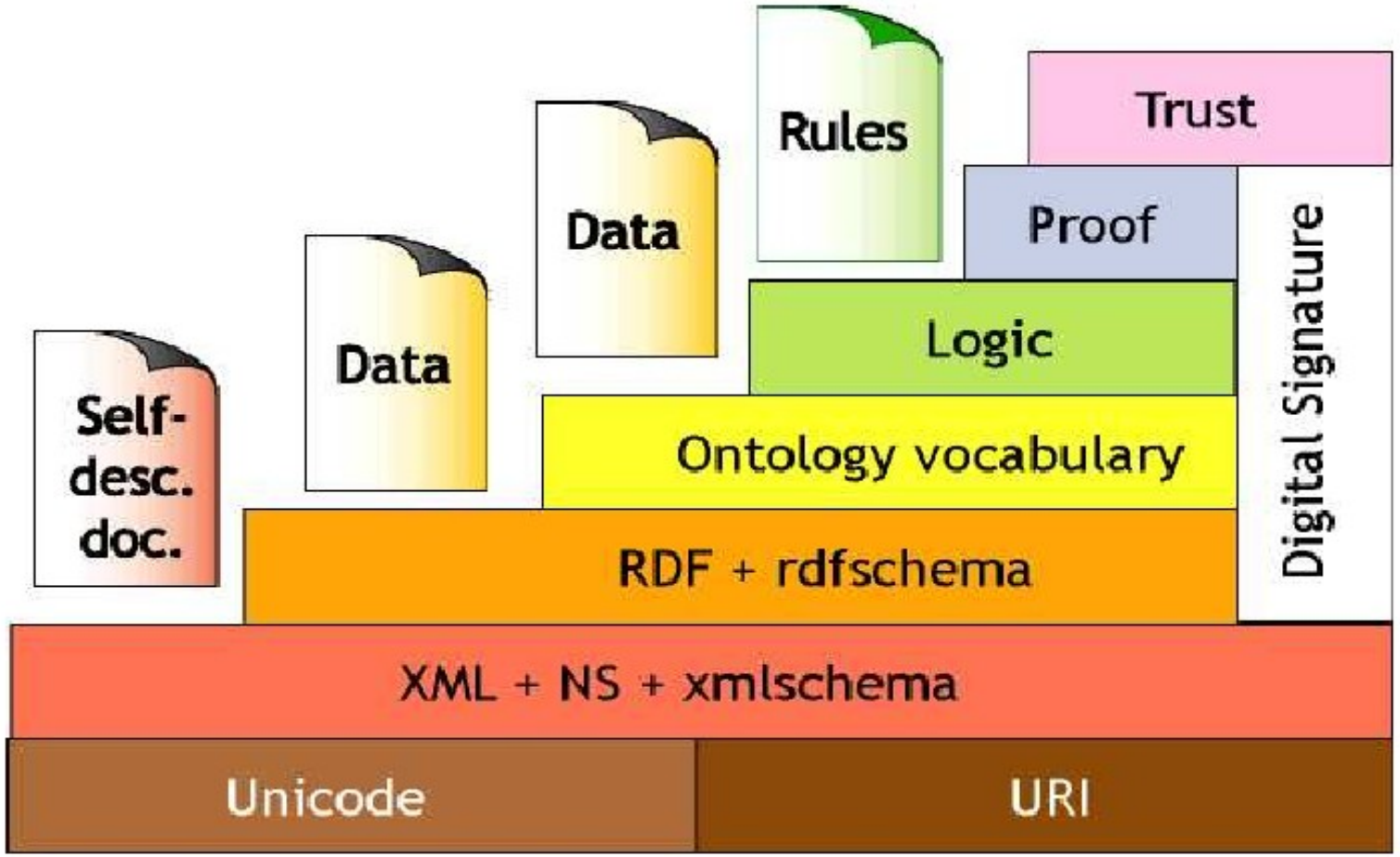

Fig. 1. The Semantic Web Tower

Different layers part 1

- URI & Unicode
  - Standard way to to define references to entities and exchange symbols
- XML + NS + XML Schema
  - XML provides a standard to represent labelled trees
  - XML Schema provides a mechanism for define grammars for legal XML documents
  - Namespace mechanism allows the combination of documents with heterogeneous vocabulary

Different layers part 2

- RDF(Resource Description Framework)
  - data model for describing machine processable semantics
  - Consist of 3 types: resources, properties, statements
- RDF Schema
  - defines a simple modelling language on top of RDF
  - introduce modelling primitives for: classes, is-a relationship between classes and properties, and domain and range restrictions for properties
  - Semantic extension of RDF with RDF syntax
- RDF is mainly used as a standard syntax for the semantic web. RDF Schema is expected to be the lowest layer to provide semantics for the semantic web

RDF(S)

- RDF(S) is a short form of RDF and RDF Schema

Different layers part 3

- Ontology Vocabulary
  - an ontology language is restricted to a subset of logic to define terminology
  - OWL developed for this purpose. OWL is designed to provide a richer theory of classes and properties, allowing for defined classes and more relationships between classes
  - relationship between OWL and RDF(S) is both semantic and syntactic
  - predecessors of OWL: OIL, DAML-OIL, DAML-ONT

Layering in the Semantic Web

- semantic meaning for one layer should be the base for the next
- each layer should be an extension of the next
- only semantic layering currently in the Semantic web is between RDF and RDFS approved by W3C

Universal meaning

The OIL definition:

```
<rdfs:Class rdf:ID="Herbivore">
  <rdfs:type rdf:resource="oil:DefinedClass">
  <rdfs:subClassOf rdf:resource="Animal" />
  <rdfs:subClassOf>
    <oil:NOT>
      <oil:hasOperand rdf:resource="Carnivore" />
    </oil:NOT>
  </rdfs:subClassOf>
</rdfs:Class>
```

What an RDF Schema system understands:

```
<rdfs:Class rdf:ID="Herbivore">
  <rdfs:type rdf:resource="unknown:DefinedClass">
  <rdfs:subClassOf rdf:resource="Animal" />
  <rdfs:subClassOf>
    <rdf:Description />
  </rdfs:subClassOf>
</rdfs:Class>
```

Fig. 2. OIL versus RDF Schema

Layering OWL on top of RDF(S)

Suggested approaches:

- Same syntax, semantic extension
  - Same syntax for OWL and RDF(S)
  - Semantics an extension of OWL
  - Most natural proposal, but leads to ill-defined modal theory of OWL
- Syntax and semantic extension
  - avoids paradoxical situations
  - new parsers required for OWL and RDF(S) agent aware programs would get very limited understanding of an OWL ontology
- Same syntax but diverging semantics
  - RDF treated as syntax carrier(as with RDF and XML)
  - some constructs with different meaning, but still with some overlap with RDF(S)
- Different syntax and semantics
  - OWL differs from RDF and RDF Schema in both syntax and semantics
  - diverge from the idea of semantic web tower

The problem when layering OWL on top of RDF(S) part 1

- Most attractive solution:
  - same syntax as RDF
  - semantics an extension of RDFS
  - Expected to be used by the designers of RDFS
- Not possible to extend the layering relationship between RDF(S) and OWL as it leads to semantic paradoxes
- Causes all situations to be ill-defined resulting in a collapse in the logical formalism

The problem when layering OWL on top of RDF(S) part 2

- Russel's paradox
  - a language include a large collection of built-in sets
  - all models for sets include these built-in sets
  - these built-in sets include the set consisting of those sets that do not contain themselves
  - is this set a member of itself? because it contains exactly those sets that do not contain themselves, and it does contains itself
- OWL on top of RDF(S) need a large collection of built-in classes to make the logical foundation needed.
- RDFS extending RDF does not fall into this paradox as it does not need a large collection of built-in classes

The problem in detail

- RDF has a simple theory for types of resources and property between resources
- RDFS extends this theory by creating classes and properties
- RDFS theory is very weak, as it use modeling primitives in both specifying the language and in describing ontologies
- The intent of OWL is to provide an even richer theory of classes and properties, but it is restricted by the semantic constraints of RDF(S)

Example

- An OWL interpretation including:
  - RDFS class Person, a property child and an object John(with no outgoing child relationship)
  - an OWL restriction of a property to a type i.e., given a property child and a class Person. The restriction corresponds to: all children belongs to Person
  - John is a part of this restriction because John has no children and thus all John's children belong to Person

- OWL interpretations must include resources for many restrictions, essentially all the combination of restrictions that can be built out of the classes and properties in the interpretation

Solutions for layering OWL on top of RDF(S) part 1

- Limiting Entailment
  - Give up on entailments that require the presence of the problematic restrictions in all interpretations

  If John was a member of Student and Employee it would not follow that John is a member of the intersection of Student and Employee

Solutions for layering OWL on top of RDF(S) part 2

- Syntactic and Semantic Extension
  - OWL defines new syntactic constructs that does not exist in RDF(S)
  - An OWL document would consist of three parts, a RDF part setting up base facts, a RDFS part creating classes and relations, and an OWL part giving meaning to some of the classes
  - An RDF(S) system would still be able to understand considerable parts of OWL ontologies
  - supports the view of semantic tower with more powerful languages for each layer with a common semantic core

```
<owl:cardinality maximum=''0'' property=''friend''>
Person
</owl:cardinality>
```

Solutions for layering OWL on top of RDF(S) part 3

- Same Syntax but Diverging Semantics
  - Keep RDF(S) syntax but not abide by all of RDF(S) meaning(same was as RDF and XML)
  - RDF viewed as a syntactic mechanism for defining ontology languages instead of viewed as a basic ontology language
    - a relation r as an attribute of a class c versus
    - as a global property r with c as its domain

Solutions for layering OWL on top of RDF(S) part 4

- Diverging Syntax and Diverging Semantics
  - RDF is a terrible syntax for complex constructs
  - OWL syntax is more complex than RDF(S) and often needs to be encoded as several triplets
  - In this view OWL takes the useful portions of RDF syntax and semantics and replace the rest, class and property theory from RDFS is not used
  - diverges with the semantic web tower idea

Conclusion

- The biggest problem with layering OWL on top of RDF(S) is not RDF(S) itself but instead from the vision of the developers
- RDF encompasses the whole of the semantic web and all information is represented with RDF syntax
- RDF(S) have been made too comprehensive,with little thought on possible extensions
- More significant extensions to RDF(S) require either an extension to the syntax of RDF or some deviation of the semantics of RDF(S)

New Semantic Web Stack

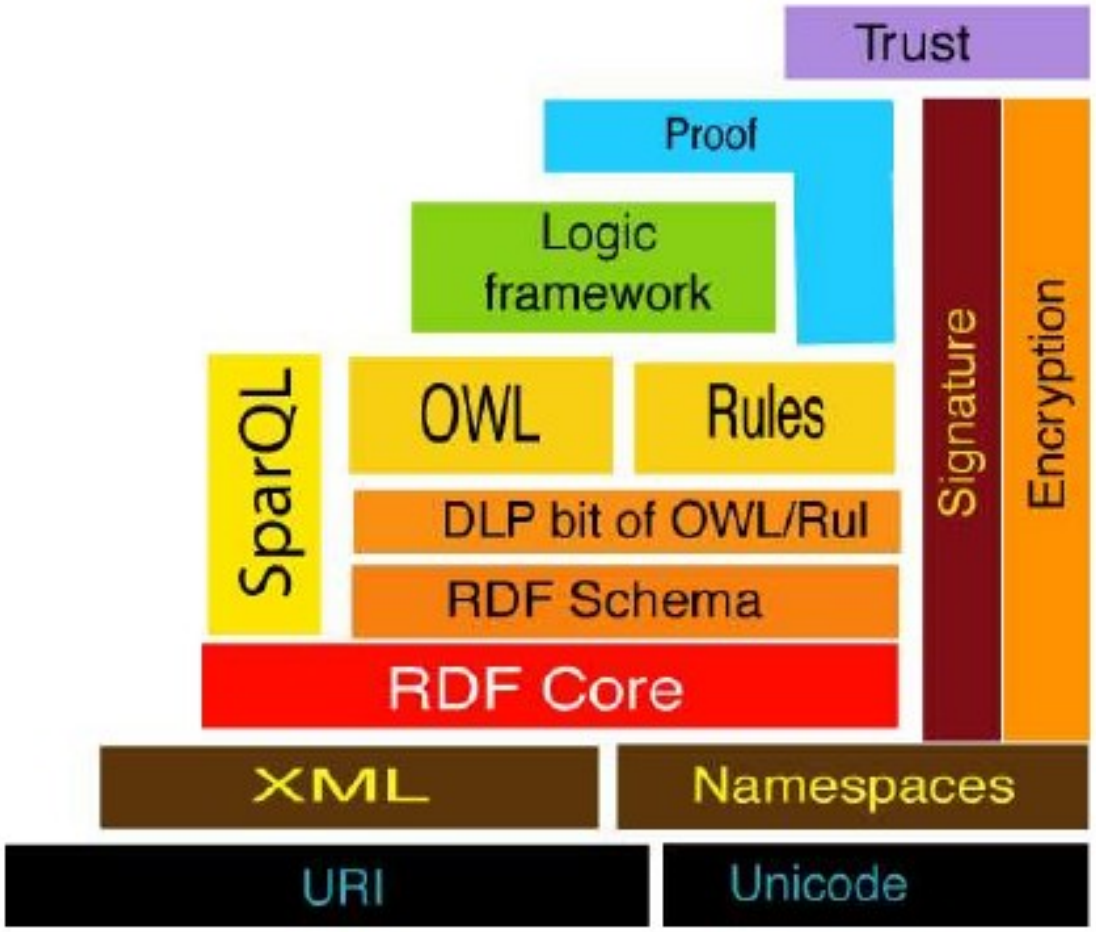

Fig. 2. New Semantic Web Stack