F12: The Semantic Web
RDF & RDFS
OWL

ATHENA: SAP, SINTEF
INTEROP: AIDIMA, SINTEF

The Tree of Knowledge Technologies
(fra Top Quadrant)
Popularity of Semantic Web

- Standardization of OWL
  - increases interest and acceptance
- Disambiguation
  - unique identifiers called URIrefs

The **cardinal** was bright red

church:cardinal

church:cardinal is short for http://www.example.org/terms#cardinal
The general vision

Serious Problems in information:
• finding
• extracting
• representing
• interpreting
• and maintaining

Bringing the web to its full potential: data and semantic interoperability

WWW
URI, HTML, HTTP

Semantics Web
RDF, RDF(S), OWL

The general vision

• IT, and especially Internet/WWW, have boosted potential for knowledge acquisition and sharing.
• BUT information resources are heterogeneous, distributed, semi-structured, & enormous in size.
• HENCE: need for tools for selective semantic (meaning-oriented) access:
  • Move from keyword search to query answering
  • Move upwards in the data-info-knowledge chain
• Vision: toward the next-generation “semantic” (Tim Berners-Lee, W3C) or Semantic Web
What is the Semantic Web?

The Semantic Web is a major research initiative of the World Wide Web Consortium (W3C) to create a metadata-rich Web of resources that can describe themselves not only by how they should be displayed (HTML) or syntactically (XML), but also by the meaning of the metadata.

From W3C Semantic Web Activity

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.


The first step is putting data on the Web in a form that machines can naturally understand, or converting it to that form. This creates what I call a Semantic Web—a web of data that can be processed directly or indirectly by machines.

Tim Berners-Lee, Weaving the Web, Harper San Francisco, 1999

Most of the Web's content today is designed for humans to read, not for computer programs to manipulate meaningfully.

What is the Semantic Web good for?

- 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.

The Semantic Web layers

What lies beneath the Semantic Web?

- Self-desc. doc.
- Data
- Rules
- Ontology vocabulary
- RDF + rdfschema
- XML + NS + xsmlschema
- Unicode
- URI
- Trust
- Proof
- Digital Signature
- Logic
Components of Semantic Web Architecture

- Unicode, URI
- XML, Namespaces
- RDF Core, RDF Schema,
- OWL, Description Logic Programming bit of OWL/Rules
- SparQL
- Rules, Logic Framework, Proof
- Trust, Signature, Encryption

“If HTML and the Web made all the online documents look like one huge book, RDF, schema, and inference languages will make all the data in the world look like one huge database.”

- Tim Berners-Lee, Weaving the Web, 1999
RDF: Resource Description Framework

RDF is the simplest of the semantic languages. At the simplest level, the Resource Description Framework is an XML-based language to describe resources.

• Basic Idea #1: **RFD uses triples**
  - RDF is based on a subject-verb-object statement structure.
  - RDF subjects are called resources (classes).
  - Verbs (predicates) are called properties.
  - Objects (values) may be simple literals or other resources.

• Basic Idea #2: **Everything is a resource that is named with a URI**
  - RDF nouns, verbs, and objects are all labeled with URIs.
  - A URI is just a name for a resource.
  - It may be a URL, but not necessarily.
  - A URI can name anything that can be described.
  - Web pages, telephone numbers, concepts, creators of web pages, organizations that the creator works for….
Resource Description Framework (RDF)

- A language for making simple statements about things (resources)
- Statements: Subject Predicate Object (triples)
  - Item1 isOrderFor Product1
  - Item1 is-a Item
  - Product1 hasName “Lawnmower”

LinEItem database table:

<table>
<thead>
<tr>
<th>partNum</th>
<th>productName</th>
<th>quantity</th>
<th>USPrice</th>
<th>comment</th>
<th>shipDate</th>
</tr>
</thead>
<tbody>
<tr>
<td>572-AA</td>
<td>Lawnmower</td>
<td>1</td>
<td>148.95</td>
<td>Confirm this is electric</td>
<td>21.06.1999</td>
</tr>
<tr>
<td>102-AA</td>
<td>Baby Monitor</td>
<td>1</td>
<td>39.95</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

RDF and URIrefs

- Things are identified by Uniform Resource Identifiers (URI, URIref)
  - Avoids naming clashes
    - http://www.co.uk/vocabulary#Item1 (v:)
    - http://www.w3.org/1999/02/22-rdf-syntax-ns#type (rdf:)
- Same example using namespace prefixes
  - v:Item1 v:isOrderFor v:Product1
  - v:Item1 rdf:type v:Item
  - v:Product1 v:hasName “Lawnmower”
- Subject and Predicate are always resources
- Objects can be either resources or literals (see 3rd triple)
RDF data model

- RDF statements can be expressed using XML syntax
- But, the RDF data model is a graph of nodes and directed arcs
  - Subjects and objects are nodes
  - Predicates (also called Properties) are directed arcs from the subject to the object.
  - Properties relate individuals to individuals (or values)

Instance diagram

- Properties relate individuals to individuals (or values)
RDF Schema

- Defines the terms (vocabulary) to use in RDF models
  - v:Item, v:isOrderFor
- Can use XML Schema data types like xsd:string, xsd:date
- A simple ontology language
  - Class
  - Subclass
  - Property Sub-property
  - Domain and range restrictions on properties

```
<table>
<thead>
<tr>
<th>a:Pizza</th>
<th>a:Topping</th>
</tr>
</thead>
<tbody>
<tr>
<td>a:hasTopping</td>
<td></td>
</tr>
</tbody>
</table>
```

RDF basic ontology

- Class  Vehicle
  - “a group, set, or kind sharing common attributes”
    source: Merriam-Webster
- Subclass  Car
  - “a primary division of a class”
    source: Merriam-Webster
  - If Car is a subclass of Vehicle, then every Car is a Vehicle
- Property  has-parent
  - an attribute (common to all members of a class)
    source: Merriam-Webster
  - in an ontology properties are global in scope
- Subproperty  has-mother (sub-property of has-parent,)
  - If Dick has-mother Jane, then Dick has-parent Jane
RDF and reasoning

• Has a formal model-theoretic semantics
  – Can be mapped to first-order predicate logic
• Can be used for simple reasoning

{organism,…}
{animal,…}
{bird,…}
{robin…}

If the domain of hasTopping is restricted to Pizza
and a mudslide is defined as a thing that has a ChocolateTopping, which is a
kind of Topping,
then we can conclude that a mudslide is a Pizza.

However, a mudslide is a type of ice cream cone.

RDF Schema

Some RDFS Classes (Subjects and Values)

<table>
<thead>
<tr>
<th>RDFS: Resource</th>
<th>The RDFS root element. All other tags derive from Resource.</th>
</tr>
</thead>
<tbody>
<tr>
<td>RDFS: Class</td>
<td>The Class class. Literals and Datatypes are example classes. Classes consist of entities that share properties.</td>
</tr>
<tr>
<td>RDFS: Literal</td>
<td>The class for holding Strings and integers. Literals are dead ends in RDF graphs.</td>
</tr>
<tr>
<td>RDFS: Datatype</td>
<td>A type of data, a member of the Literal class.</td>
</tr>
<tr>
<td>RDFS: XMLLiteral</td>
<td>A datatype for holding XML data.</td>
</tr>
<tr>
<td>RDFS: Property</td>
<td>This is the base class for all properties (that is, verbs, ad</td>
</tr>
</tbody>
</table>
RDF Schema

Some RDFS Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>subClassOf</td>
<td>Indicates the subject is a subclass of the object in a statement.</td>
</tr>
<tr>
<td>subPropertyOf</td>
<td>The subject is a subProperty of the property (masquerading as an object).</td>
</tr>
<tr>
<td>Domain</td>
<td>Restricts a property to only apply to certain classes of subjects.</td>
</tr>
<tr>
<td>Range</td>
<td>Restricts the values of a property to be members of an indicated class or one of its subclasses.</td>
</tr>
<tr>
<td>type</td>
<td>Denotes an instance of a particular class. Actually from RDF, not RDFS.</td>
</tr>
</tbody>
</table>

RDF Schema compared to XML

- Has a formal model-theoretic semantics
- By contrast, there is no formal semantics for XML documents like po.xml
  - po.xsd can be turned into an ontology and po.xml into an instance of it
  - But, there is no standard algorithm to perform that transformation
    - no single interpretation

```xml
<?xml version="1.0"?>
<purchaseOrder orderDate="1999-10-20">
  <shipTo country="US">
   <name>Alice Smith</name>
   <street>123 Maple Street</street>
  </shipTo>
  <billTo country="US">
   <name>Robert Smith</name>
  </billTo>
  <comment>Hurry, my lawn is going wild!</comment>
  <items>
    <item partNum="872-AA">
     <productName>Lawnmower</productName>
     <quantity>1</quantity>
     <USPrice>148.95</USPrice>
     <comment>Confirm this is electric</comment>
    </item>
    <item partNum="926-AA">
    </item>
  </items>
</purchaseOrder>
```
Ontology Web Language (OWL)

- A more expressive ontology language
- Concepts (classes) can be described or defined
  - described – necessary conditions given
  - defined – necessary and sufficient conditions given
- Builds on RDF and can be expressed in several ways:
  - RDF XML-based syntax
  - abstract syntax
  - graphic UML-like
- Has three sub-languages:
  - OWL Full
  - OWL Description Logic (DL) – maps to a DL, a subset of predicate logic
  - OWL lite – for simple taxonomies (class hierarchies)
Logical languages for the Semantic Web

DAML+OIL

DARPA (Defense Advanced Research Program) Agent Markup Language-Ontology Inference Layer: These are two XML- and Web-based languages to support the Semantic Web, which have recently fused.

DAML originated from a US DARPA-sponsored program; OIL originated from a European Union-sponsored program. Together they constitute the most semantically expressive language available for WWW documents. The combined language is now supported by the W3C Web standards consortium.

Logical languages for the Semantic Web

DARPA’s DAML/W3C’s OWL Language

Web Languages
RDF/S
XML

DAML-ONT

DAML+OIL

OWL

OIL

Formal Foundations
Description Logics

FACT, CLASSIC, DLP, ...

Frame Systems
### Logical languages for the Semantic Web

**OWL**

Web Ontology Language (sometimes called Ontology Web Language) language developed by the W3C's Web Ontology Working Group and intended to be the successor of DAML+OIL. OWL is the most expressive knowledge representation for the Semantic Web so far.

OWL has three levels of language: OWL Lite, OWL DL (for description logic), and OWL Full. These three levels are in increasing order of expressivity.

---

**RDF/RDFS**

The first language (RDF) expresses instance-level semantic relations phrased in terms of a triple: `<subject, verb, object>`, i.e., `<object1, relation1, object2>`. The second (RDFS) expresses class-level relations describing acceptable instance-level relations.

RDF/RDFS is considerably less expressive than OWL and DAML+OIL.
Logical languages for the Semantic Web

An example of the reasoning possibilities of the logical languages

- The head of an organization is also a member of it
- A member of a terror organization is a terrorist
- Therefore, the head of a terror organization is a terrorist

```
<owl:Property rdf:ID="head">
  <rdf:subPropertyOf rdfs:resource="member" />
</owl:Property>
<owl:Class rdf:ID="Terrorist">
  <owl:sameClassAs>
    <owl:Restriction>
      <owl:onProperty rdf:resource="member" />
      <owl:someValuesFrom rdf:resource="TerroristOrg" />
    </owl:Restriction>
  </owl:sameClassAs>
</owl:Class>
```

Henri Parot --- type --- Terrorist

ETA --- type --- TerrorOrg
OWL Classes

- OWLClass
  - complete : Boolean
  - deprecated : Boolean
  - OWLdisjointWith
    - 0..*
  - OWLintersectionOf
    - 0..*
    - 0..*
    - +OWLdisjointWith
  - OWLequivalentClass
    - 0..*
  - +OWLoneOf
    - 1..*
  - Individual

- RDFSClass
- OWLRestriction
- EnumeratedClass
- ComplementClass
- IntersectionClass
- UnionClass

- OWLRestrictions
- RDFProperty
  - +OWLonProperty
    - 1
  - OWLRestriction

- RDFSResource
  - +RDFSresource
    - 1

- RDFSLiteral
  - +RDFSLiteral
    - 1

- RDFSProperty
  - +RDFSProperty
    - 1

- HasValueRestriction
  - +OWLhasValue
    - 1

- AllValuesFromRestriction
  - +OWLallValuesFrom
    - 1

- SomeValuesFromRestriction
  - +OWLsomeValuesFrom
    - 1

- CardinalityRestriction
  - +OWLcardinality
    - 1

- MaxCardinalityRestriction
  - +OWLmaxCardinality
    - 1

- MinCardinalityRestriction
  - +OWLminCardinality
    - 1

- BICardinalityRestriction
  - +OWLbCardinality
    - 1
OWL Properties

OWL DatatypeProperty
- deprecated : Boolean
- functional : Boolean

OWL ObjectProperty
- deprecated : Boolean
- functional : Boolean
- inverseFunctional : Boolean
- symmetric : Boolean
- transitive : Boolean

OWL Individuals

RDFS Resource (from RDFS)

RDFS Literal (from RDFS)

DatatypeSlot
- property : OWL DatatypeProperty
- content

ObjectSlot
- property : OWL ObjectProperty
- content

OWL Individuals
- OWL Different
- OWL sameAs
- OWL differentFrom
- OWL allDifferent
- OWL disjointWith
- OWL equivalentClass
- OWL hasValue
- OWL inDefinedWith
- OWL inRestrictionWith
- OWL memberOf
- OWL someValuesFrom
- OWL symmetricObjectProperty
- OWL transitiveObjectProperty

RDF Property (from RDFS)

+ OWL inverseOf
+ OWL equivalentProperty

OWL EquivalentProperty

OWL InverseOf

0..1

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *

0.. *
OWL PurchaseOrder

- <rdf:RDF
  - xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  - xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  - xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  - xmlns:owl="http://www.w3.org/2002/07/owl#"
  - <owl:Class rdf:ID="PurchaseOrderLine">
    - <rdfs:subClassOf>
      - <owl:Class rdf:ID="PricedLine"/>
    - </rdfs:subClassOf>
    - <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
      Inherits from Line and contains information related to delivery.
    - </rdfs:comment>
    - </owl:Class>
  - <owl:ObjectProperty rdf:ID="changesLineValue">
    - <rdfs:range rdf:resource="#Amount"/>
    - <rdfs:domain rdf:resource="#PricedLine"/>
  - </owl:ObjectProperty>
- </rdf:RDF>

OWL versus UML

<table>
<thead>
<tr>
<th>In OWL and not in UML</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thing, global properties, autonomous individual</td>
<td>In OWL, instances as well as some relations (in owl, relations are called properties), can exist without being attached to certain class. This is due to the fact that OWL is based on sets while UML is based on types. Instances and relations in OWL can be a subset of the universal class Thing or binary relation between two Things.</td>
</tr>
<tr>
<td>Class-specific cardinality redefinition</td>
<td>As OWL properties can be declared independent of classes, they can have different cardinality definitions when applied to different classes.</td>
</tr>
<tr>
<td>allValuesFrom, some ValuesFrom</td>
<td>&quot;In OWL, property can have its range restricted when applied to particular class, either that the range is limited to a class (subclass of range if declared) (allValuesFrom) or that range must intersect a class (someValuesFrom).&quot; [28]</td>
</tr>
<tr>
<td>SymmetricProperty, TransitiveProperty</td>
<td>OWL allows properties to be declared symmetric or transitive. In both cases the domain and range must be type compatible.</td>
</tr>
<tr>
<td>Classes as instances</td>
<td>In UML or MOF defined languages, there is a strict separation of metalevels so that population of M1 classes is distinct from the population of M2 classes. In OWL full, one class can be an instance of another class, a characteristic inherited form RDF. In OWL DL, this usage is restricted.</td>
</tr>
</tbody>
</table>
Protege: Good subclass of Beautiful

premises:
Eros is lacking in what is beautiful
What is good is beautiful

conclusion:
Eros is lacking in what is good
Protege: Eros described

**Premises:**
- Eros is lacking in what is beautiful
- What is good is beautiful

**Conclusion:**
- Eros is lacking in what is good

---

**Question 3.4**

Why did I say that Eros was described rather than defined?
Answer 3.4

Why did I say that Eros was described rather than defined? because the conditions are only necessary.

Protege: NonGood defined

premises: Eros is lacking in what is beautiful What is good is beautiful

conclusion: Eros is lacking in what is good
Protege: Eros classified by reasoner

premises:
Eros is lacking in what is beautiful
What is good is beautiful

conclusion:
Eros is lacking in what is good

Term. Mgmt. compared to ontology creation

- Business importance of terms – use same term for same concept
  - To avoid misunderstandings that cost money, time, quality, reputation. e.g., use of standard terms and signs in the chemical industry
  - Use terminology database to support technical writing and translation, e.g., English to Spanish
- Similarity of process activities:

  **Ontology Creation (use in programs)**
  from book: *Semantic Web Primer*
  1. Determine Scope - purpose
  2. Consider reuse
  3. Enumerate terms
  4. Define taxonomy
  5. etc.

  **Terminology Management (use by people)**
  1. Determine boundaries of subject
  2. With the help of experts locate **artifacts**
  3. Extract terms from artifacts
  4. Write definitions
  5. Organize terms
Standard stages to develop an ontology (by hand)

- Determine Scope - purpose
- Consider reuse of existing ontologies
- Enumerate terms  Define Terms (this was left out of the book)
  - Statement from member of EU project on semantic-web services: a major barrier to re-use is poor documentation
  - Need definitions so that users and ontology developers can understand the ontology
- Define taxonomy (class hierarchy)
- Define properties
- Define facets
- Define instances
- Check for anomalies, debug

Trust and credibility

URI variable

Alan:
1) If X is AC rep of Y, X can delegate W3C member access rights in Y.
2) Kari is AC rep of Elisa.

Kari:
1) If X is employee of Elisa, X has W3C member access rights.
2) Tiina is employee of Elisa.

Tiina: I have W3C member access rights
Proof: Alan 1, Alan 2, Kari 1, Kari 2
Conclusions

• To make data machine processable, we need:
  – Syntactic interoperability between data: **Unicode** and **XML**.
  – Unambiguous names for resources (that may also bind data to real world objects): **URIs**.
  – A common data model to access, connect and describe the resources: **RDF**.
  – To define common vocabularies: **RDFS, OWL**.
  – Reasoning logics: **OWL, Rules**.

• The semantic Web is an extension of the current Web, providing an infrastructure for the **interoperability of data** on the Web
Bruksområder

• **RDF/ OWL**
  Utvide dagens web til en web av data.

• **Topic Maps**
  Strukturering av og navigering i portaler, IT-støtte kunnskapsforvaltning.

• **Core Components**
  B2B transaksjoner mellom forretningskritiske systemer. Ex ordre, faktura, transportdokumenter

• **ISO 15926**

• **UML**
  Design av programwaresystemer.

Andre karakteristika

<table>
<thead>
<tr>
<th>Adopsjonsgrad</th>
<th>RDF/ OWL</th>
<th>Topic Maps</th>
<th>Core Components</th>
<th>ISO 15926</th>
<th>UML</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tilgjengelighet på kompetanse</td>
<td>Noe</td>
<td>Noe</td>
<td>Liten – UML og XML hjelper godt.</td>
<td>Liten</td>
<td>Meget god.</td>
</tr>
<tr>
<td>Verktøystøtte</td>
<td>Noe</td>
<td>Noe</td>
<td>Noe</td>
<td>Liten</td>
<td>Meget god.</td>
</tr>
</tbody>
</table>
Representasjon av klassebegreper

**UML:**
- Class
- Attribute
- Association

**OWL:**
- owl:Class
  - objectProperty
  - valueProperty
- value

**Topic Maps:**
- Topic
- Association

**Core Components:**
- Aggregate CC / BIE
- Basic CC / BIE
- Association CC / BIE

**ISO 15926:**
- Class
  - ID
- Association
- Class
  - ID

CC=Core Component   BIE=Business Information Entity (CC+kontekst)

Ordrescenario

**Modellere produkttkatalog**
**Produktsøk/ Navigering**
**Info modell ordredokument**
**Utvekslings-format**
**Validering**

Interoperability Research for Networked Enterprises Applications and Software
### ST – verdibidrag i et ordrescenario

<table>
<thead>
<tr>
<th>Modell av</th>
<th>Produktet-katalog</th>
<th>Informasjons-modell for ordre</th>
<th>Utvekslings-format på instansnivå</th>
<th>Validering</th>
</tr>
</thead>
<tbody>
<tr>
<td>XML</td>
<td>XSD brukes for å definere struktur i katalog.</td>
<td>Bygger datamodell i XSD.</td>
<td>XML</td>
<td>XSD</td>
</tr>
<tr>
<td>UML</td>
<td>Standard visualisering av klassemodell for katalogstruktur og innhold.</td>
<td>Bygger informasjonsmodell i UML.</td>
<td>XML basert på XML.</td>
<td>Validering basert på XML/ XML.</td>
</tr>
<tr>
<td>TM</td>
<td>Emnekart gir intuitivt forståelig katalog-struktur. Lätt utvändbar, lett å finna det kunden vil ha.</td>
<td>TMQL er felles spørrespråk för sök och navigering.</td>
<td>Bygger/gjenbruker informasjonsmodell i TMCL.</td>
<td>XTM (standard XML-format för emnekart). URI-er identifierar modell- og instans- elementer.</td>
</tr>
</tbody>
</table>

### Metamodeller

![Diagram med metamodeller](image-url)
Kommentarer metamodeller

• Store metamodeller
  + Gir et presist rammeverk for hvordan modeller skal bygges og tolkes. Fordel for integrasjon av store ontologier med høye presisjonskrav.
  - Meget høy brukterskel for bruk av metamodell for små ontologier. Trenger verktøystøtte.

• Små metamodeller
  + Stor fleksibilitet ved bygging av modeller
  - Uten presise retningslinjer for modellering er det vanskelig å integrere modeller (ontology alignment).

Standardformatene – bruk og typer

• Domenemodell: Denne kolonnen beskriver det formatet som selve informasjonsmodellen representeres i.

• Utvekslingsformat Instansdata: Dette er formatet som benyttes for utveksling av instansdata som er definert på generelt nivå i modell. XML generert fra UML er det tradisjonelle eksempelet.

• Utveksling av skjema: Dette er formatet som brukes for å utveksle modellene. Det vil si utveksling av informasjonsstrukturene uten instansdata.

• Runtime Instansdata: Dette er standarden som brukes i et run-time verktøy. Noen av teknologiene har ikke noe forhold til dette.

• Metodikk for utvikling av modeller: Denne kolonnen angir om det finnes noen felles prinsipper og retningslinjer for utvikling av skjema/modeller, slik at modellene blir så konsistente som mulig på tvers av brukergrupper.

• Konseptuell (teknologinøytral) domenemodell: Denne kolonnen angir om standarden inkluderer ferdig gjennombrukt domeneelementer for gjenvunnet.

• Teknologispesifikke domenevokabular: Denne kolonnen angir om standarden brukes som basis for å beskrive mer tekniske domenevokabular, som for eksempel at XML brukes for å beskrive markup av websider (XHTML).
### Områder for standardisering

<table>
<thead>
<tr>
<th>Representering av domene-</th>
<th>Utviklingsformat</th>
<th>Utvikling av domene-modeller</th>
<th>Runtime Instansdata</th>
<th>Metodikk (for utvikling av modeller)</th>
<th>Konseptuell (teknologinøytral) domenemodell/ komponentbibliotek</th>
<th>Teknologiske vokabular basert på standard</th>
</tr>
</thead>
<tbody>
<tr>
<td>RDF/OWL</td>
<td>RDFS + OWL</td>
<td>RDF/XML og N3</td>
<td>RDF</td>
<td>Ulike</td>
<td>SKOS/ DC- RDF, RSS, FOAF</td>
<td></td>
</tr>
<tr>
<td>Topic Maps</td>
<td>TM + TMCL</td>
<td>XTM</td>
<td>XTM</td>
<td>Under utvikling</td>
<td>DCTM</td>
<td></td>
</tr>
<tr>
<td>Core Component s</td>
<td>UML profil (BCSS), CEF ACT</td>
<td>XML</td>
<td>XMI</td>
<td>UMM – Unified Modelling Methodology</td>
<td>CC/BIE-bibliotek</td>
<td></td>
</tr>
<tr>
<td>ISO 15926</td>
<td>Part 4 Core Classes + Part 6 Domain Classes:</td>
<td>Part 7: Implementation methods in XML</td>
<td>Part 6: Tillegg av nye klasser i ontologi.</td>
<td>Metamodell (Upper Ontology/ Core Classes (Middle Ontology/ Domeneklasse (Ontology))</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UML</td>
<td>UML (klasse dia gram ++)</td>
<td>XMI</td>
<td>RUP (ikke standard), m.fl…</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XML</td>
<td>XSD</td>
<td>XML</td>
<td>XML</td>
<td>XHTML</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>