VAASAN YLIOPISTO

Semantic web: tutorial

Kimmo Salmenjoki
Department of Computer Science
University of Vaasa, Finland

Contents

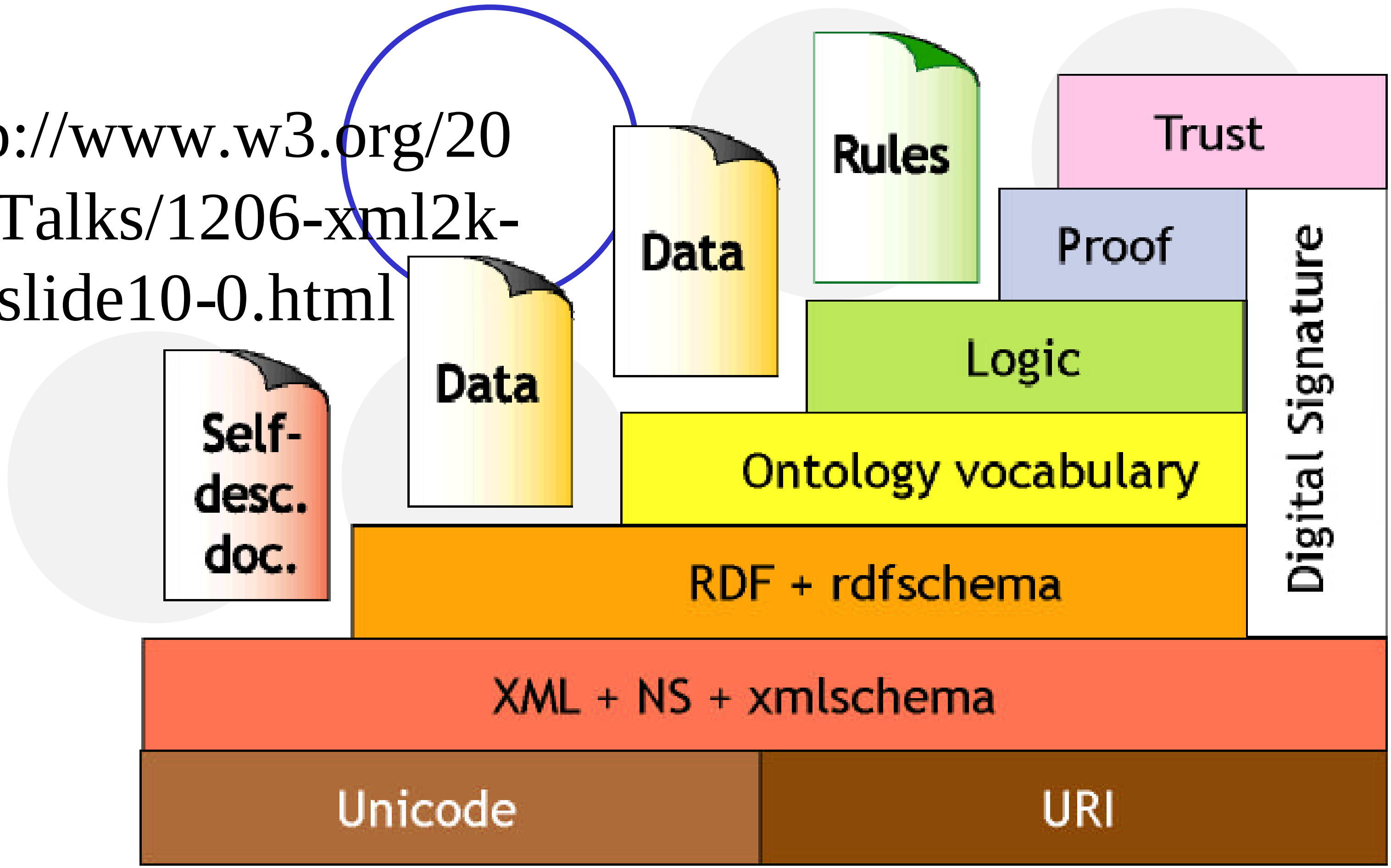
VAASAN YLIOPISTO

Background

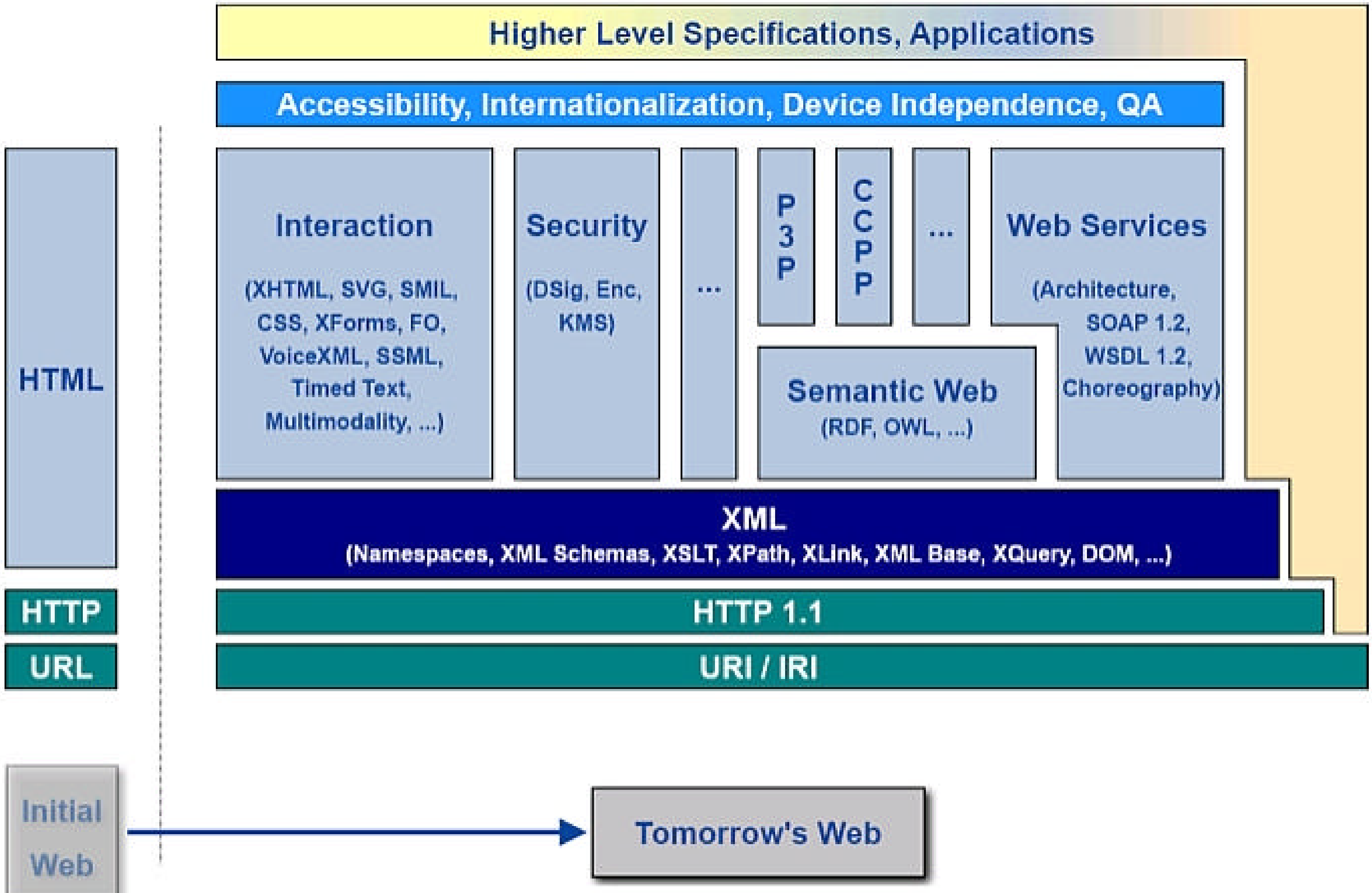
- fast and continuous web evolution with
- (XML technologies)
- (web services)
- (personal and mobile needs)

(R)evolution of XML

- http://www.w3.org/2000/Talks/1206-xml2k-tbl/slide10-0.html



Web with information and methods



http://www2004.org

II Different aspects of the semantic web SW

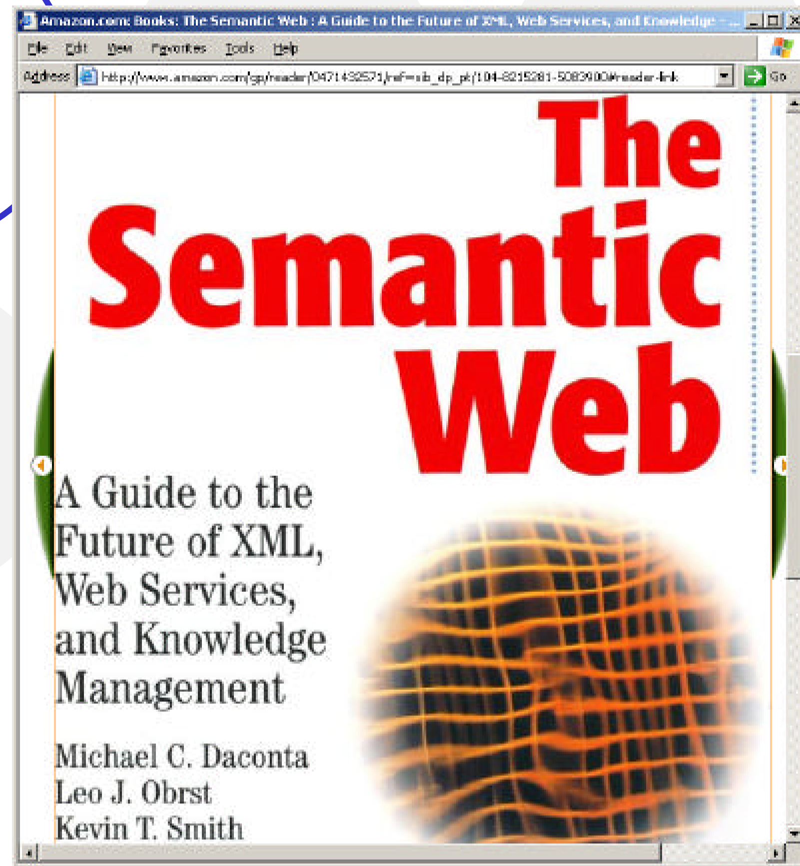
- Semantic web ”WHAT”: knowledge management
- Semantic web ”WHY”: web based knowledge integration and applications
- Semantic web ”HOW”: technologies and tools for SW

Semantic web ”WHAT”

- knowledge management
- what is the SW?
- why do we need SW?
- SW and web services
- what’s after SW?

Knowledge management

- Problem: how to find and organize information on/with web?
- The active management of information that turns into it into knowledge by selection, addition, sequence, correlation, and annotation
- [2] Michael C. Daconta, Leo J. Obrst, Kevin T. Smith: The Semantic Web: A guide to the future of XML, Web Services and Knowledge Management, John Wiley, 2003, http://www.wiley.com/legacy/compbooks/daconta/sw/



What is the SW?

- The explicit representation of the semantics underlying data, programs, pages and other web resources will enable a knowledge-based web that provides a qualitatively new level of service
- Automated services will improve in their capacity to assist humans in achieving their goals by “understanding” more of the content on the web, and thus providing more accurate filtering, categorizing, and searching of these information sources
- This process will ultimately lead to an extremely knowledgeable system that features various specialized reasoning services

VAASAN YLIOPISTO

Why do we need SW?

- the four stages progress from data with minimal smarts to data embodied with enough semantic information for machines to make inferences about it
- **Text and databases (pre-XML):** the “smarts” are in the application and not in the data
- **XML documents for a single domain:** data smart enough to move between applications in a single domain, for ex. XML in the healthcare industry
- **Taxonomies and documents with mixed vocabularies:** dat can be classified in a hierarchical taxonomy. Simple relationships between categories in the taxonomy can be used to relate and thus combine data. Thus, data is now smart enough to be easily discovered and sensibly combined with other data
- **Ontologies and rules:** new data can be inferred from existing data by following logical rules [mechanically!], for ex. Automatic translation of a document in one domain to the equivalent (or as close as possible) document in another domain

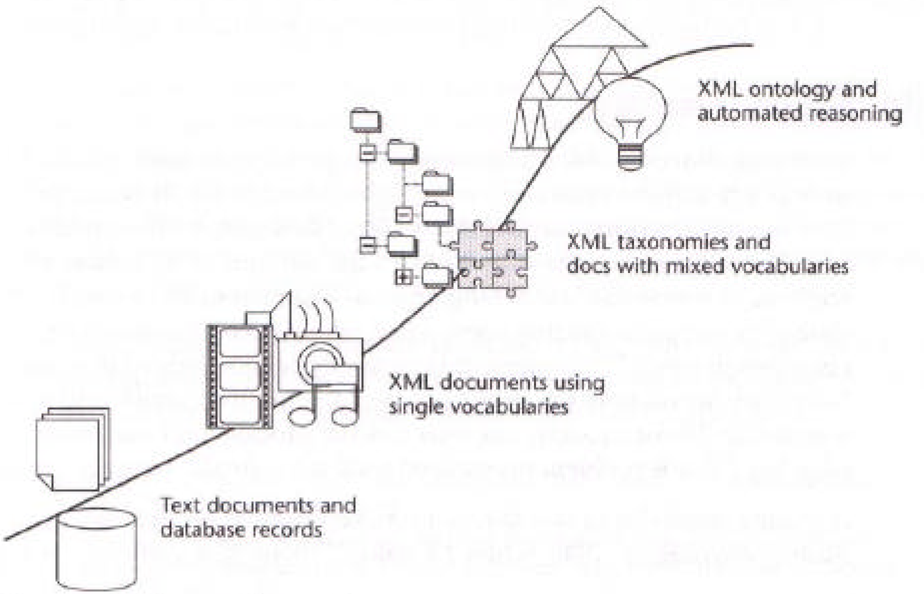


Figure 1.2 The smart data continuum.

SW, XML and web services

- **With this semantic web (SW): a machine-processable web of smart data**
- **http://www.w3.org/2001/sw/**
- XML only provides syntactic interoperability. In other words, sharing an XML document adds meaning to the content; however, only when both parties know and understand the element names

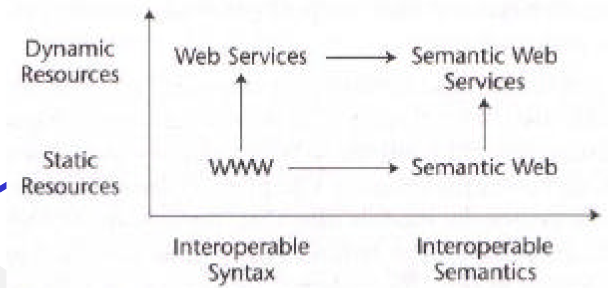


Figure 1.3 Semantic Web services.

Derived in part from two separate presentations at the Web Services One Conference 2002 by Dieter Fer and Dragan Sretenovic.

What's after SW?

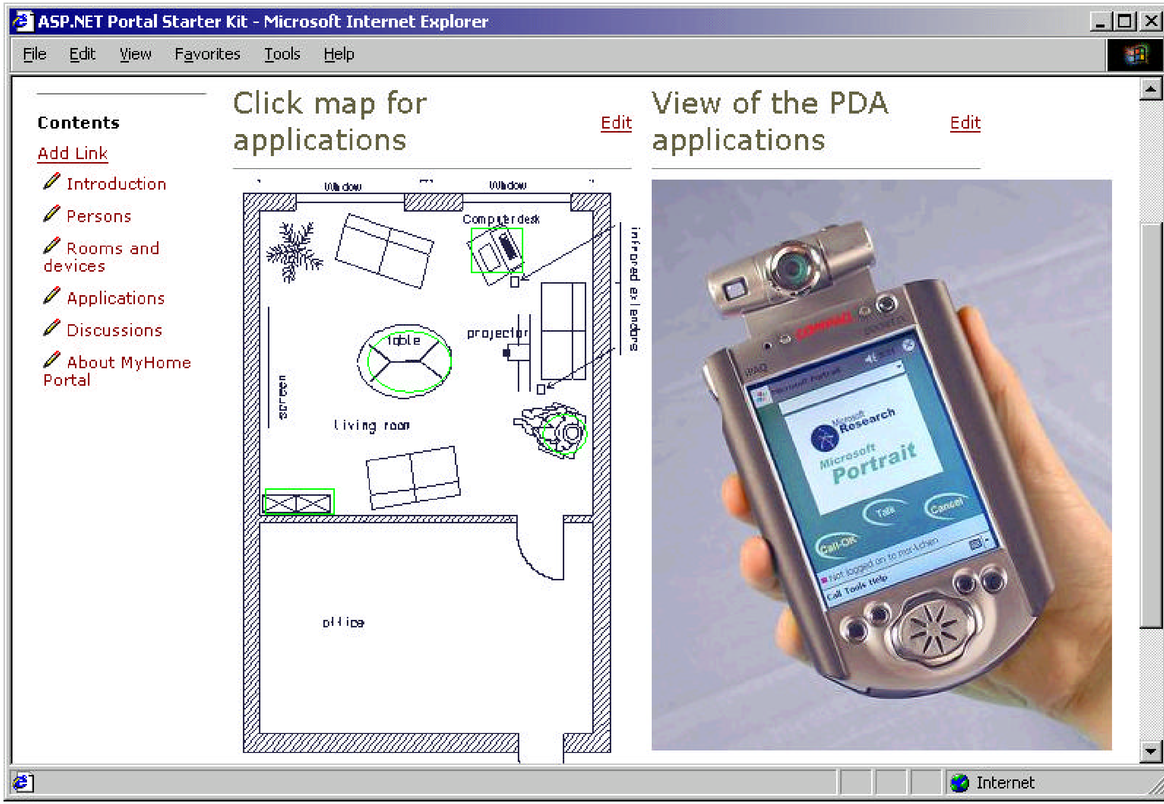
- Formal class models: Ontologies are used to represent formal class hierarchies, constrained properties, and relations between classes
- W3C's OWL Web Ontology Language (against WOL the owl in Winnie the Pooh!)
- Trust: verifying the sourse of statements is a key part of the semantic web
- SW will be practical, in terms of computing power, within three years, [2]

III Personalized services with context

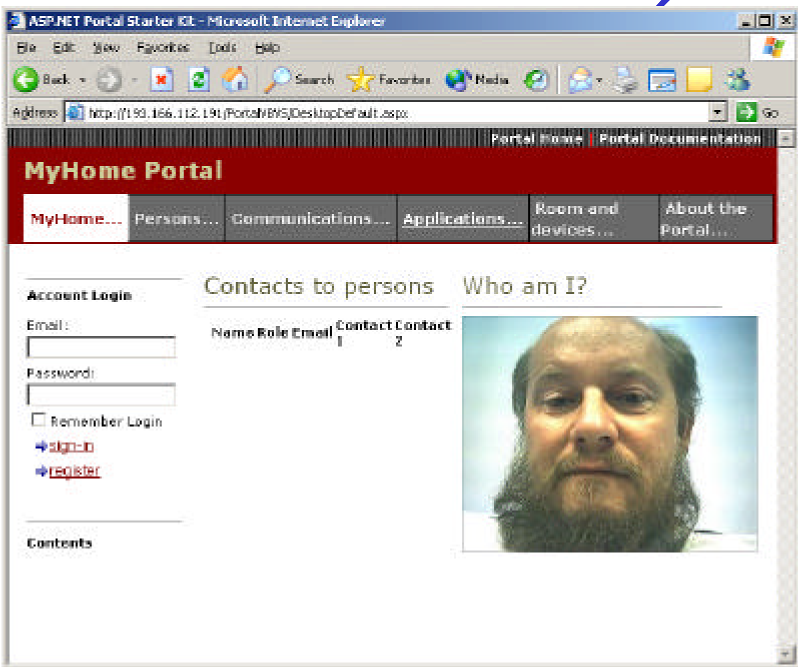
- Towards semantic information with…
- ubiquitous computing
- context
- DLR the Digital Living Room lab
- Dilemma: how to combine the general and personal knowledge management and information organization needs?

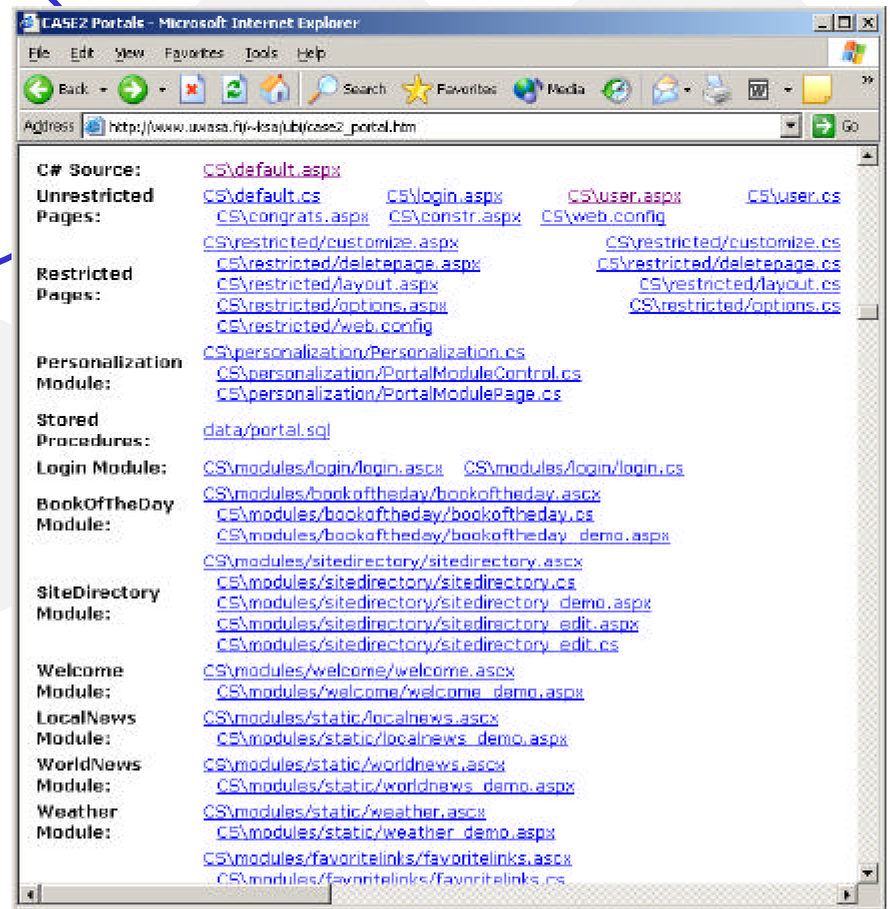
Personalized services with context

- Web as personal space
- Context handling with events and actions in the MyHome portal
- Microsoft Passport and Alert usage with SDK
- MyServices



From web content to context based services





using .NET, XML, web services with RDF and OWL

Digital livingroom (DLR)

evices in DLR lab of Technobothnia

Server / PC

Webcams

Entertainment

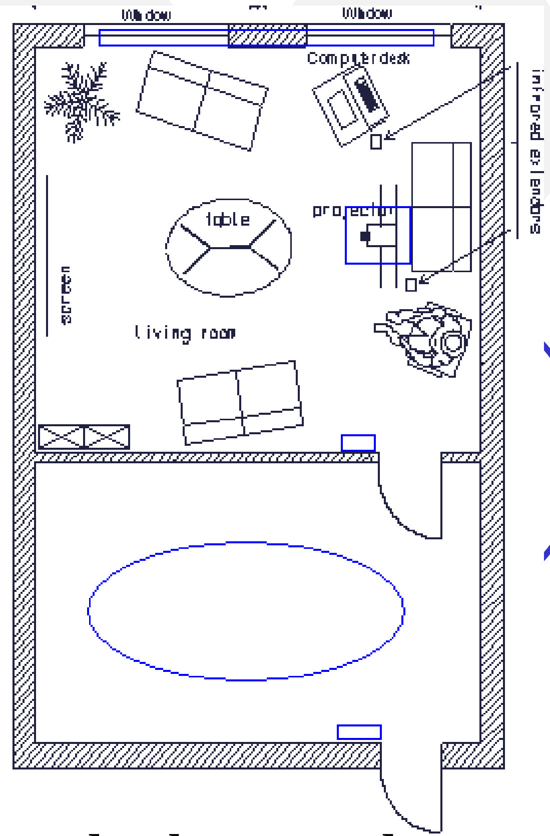
TV card

Digital & Analog TV and Radio

Speaker

Mobile devices: PDA, Smart phone

LAN, WLAN, Bluetooth



The communication through through PDAs and other wireless devices inside or outside the DLR

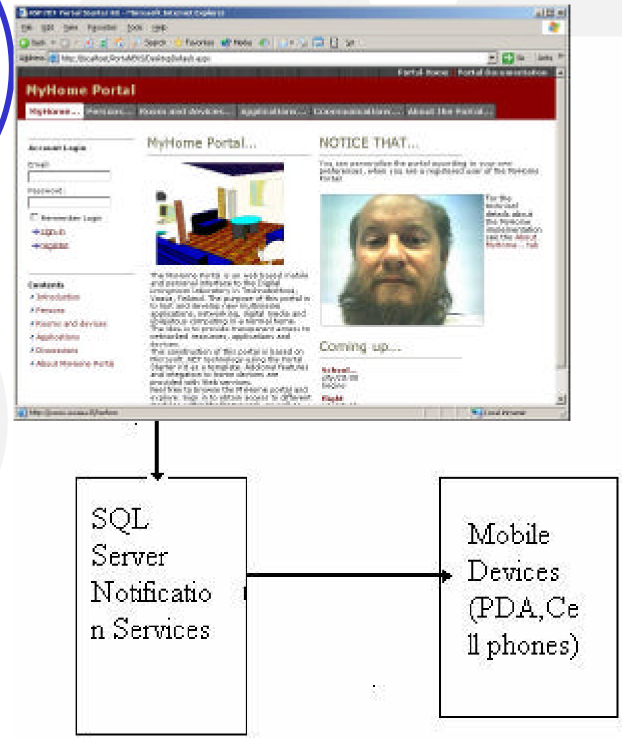
VAASAN YLIOPISTO

IV Semantic web

- Semantic web ”WHAT”: personal knowledge management
- Semantic web ”WHY”: personal web based knowledge integration and applications
- Semantic web ”HOW”: technical approaches for SW

Semantic web ”WHAT”: personal knowledge management

- make the previously general knowledge management of the SW personal



Semantic web ”WHY”:

- business case for the SW
- status of the SW
- metadata with XML technologies
- web service evolution
- web based knowledge integration and applications
- personal…

VAASAN YLIOPISTO

Business case for the SW

- By 2005 the Gartner Group reports, “lightweight ontologies will be part of 75 percent of application integration projects” (J. Jacobs, A. Linden, G G, GG Research Note T-17-5338, 20-Aug, 2002)
- The organization, that has the best information, knows where to find it, and can utilize it the quickest wins

Status of the SW

- The next big trend in web services will be semantic-enabled web services, where we can use information from web services from different organizations to perform correlation, aggregation, and orchestration
- Adobe is reorganizing its software meta data around RDF. Because of this change, “the information in PDF files can be understood by other software even if the software doesn’t know what a PDF document is or how to display it”
- Company Ontoprise sells (and buys?) ontologies, http://www.ontoprise.de/home

VAASAN YLIOPISTO

Metadata with XML technologies

- Meta data increases the fidelity and granularity of our data. The way to think of about the current state of meta data is that we attach words (or labels) to our data values to describe it. How could we attach sentences? What about paragraphs? The motivation for providing richer data description is to move data processing from being tediously preplanned and mechanistic to dynamic, just-in-time, and adaptive
- Inference engines: CWM Closed World Machine, http://infomesh.net/2001/cwm/
- RSS Resourse Description Framework Site Summary, http://www.xml.com/pub/a/2002/12/18/dive-into-xml.html
- IMS http://www.imsproject.org for interoperable learning technology
- OAG Best Practices and XML Content for Everywhere-to-Everywhere Integration, http://www.openapplications.org

Web service…

- Because a web service does not need to focus on presenting styling, the focus for creating them is purely on business logic, making it easier to reuse web services as software components in your enterprise
- MVC Model- View- Controller paradigm

VAASAN YLIOPISTO

… evolution

- XACML eXtensible Access Control Markup Language by OASIS, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml or http://xml.coverpages.org/xacml.html
- The idea of XACML is the XML documents (or SOAP messages themselves) can describe the policy of who can access them
- DAML-S is an ontology for web services, http://www.daml.org/services/
- In addition, Semantic Web Enabled Web Services (SWWS) is a comprehensive web service description framework and discovery framework to provide a scalable web service mediation, http://swws.semanticweb.org/
- Together both these technologies have the potential to increase automated usability of web services

Web based knowledge integration and applications

- possibilities:
- using XML with context (data structure)
- integration of information sources (coordination)
- automate the information production and access (methods for data)
- knowledge aware applications (like the ImageBlog)

Semantic web ”HOW”: technical approaches for SW

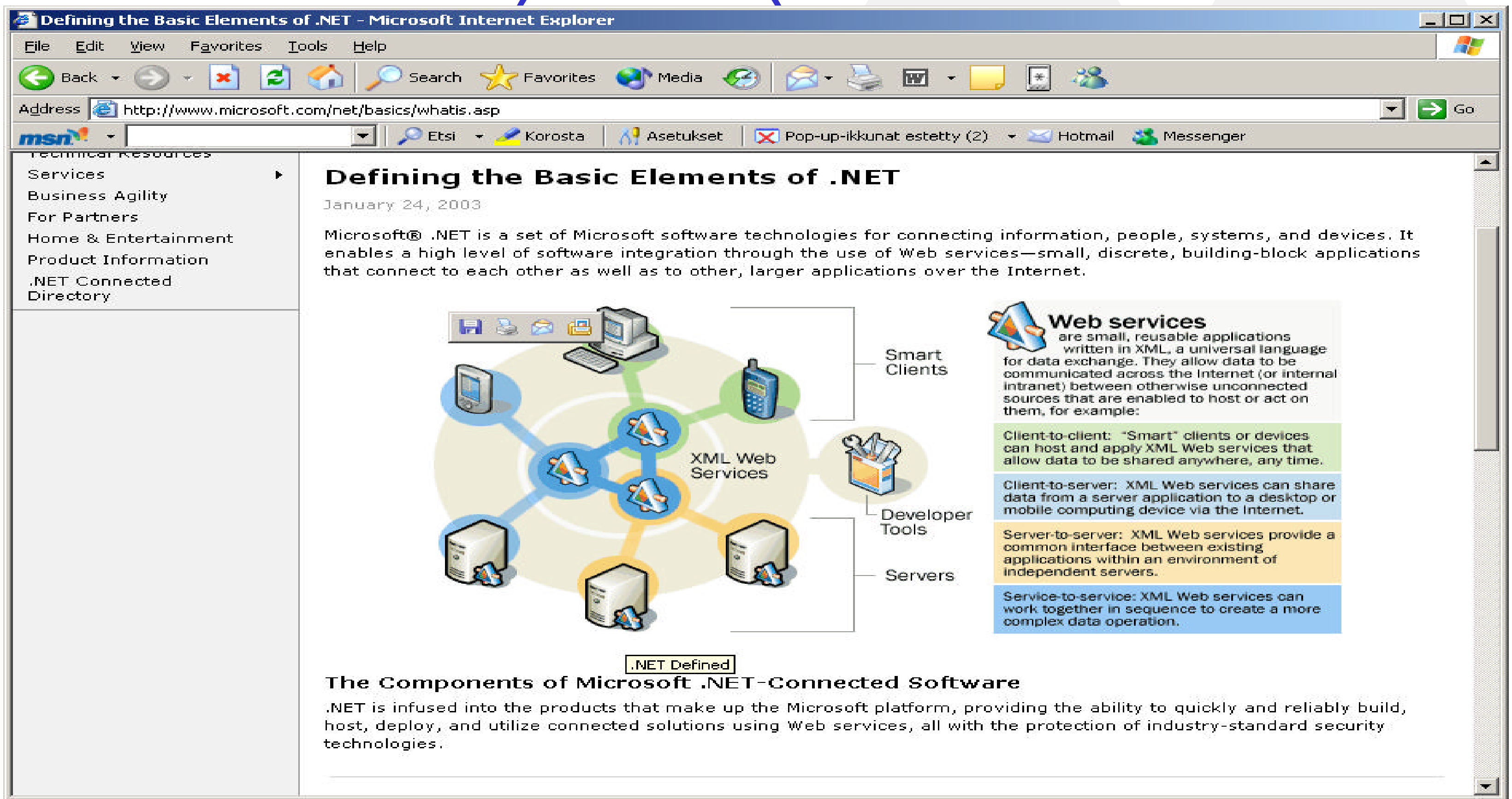
- use XML data: namespaces, Schemas, RSS, and
- web services: SOAP, WSDL, UDDI together
- and/or core SW technologies (RDF, RDFS, OWL)
- within applications (or on the whole web!)

V .NET based software engineering

- .NET web service platform
- service systems with context
- MS.NET service examples: Alert, Passport, Notification services
- context and semantic information

Online video lecture " Understanding the Framework " at http://msdn.microsoft.com/theshow/ and more technical .Net material at http://docs.msdnaa.net/ark_new/Webfiles/courseware3.htm

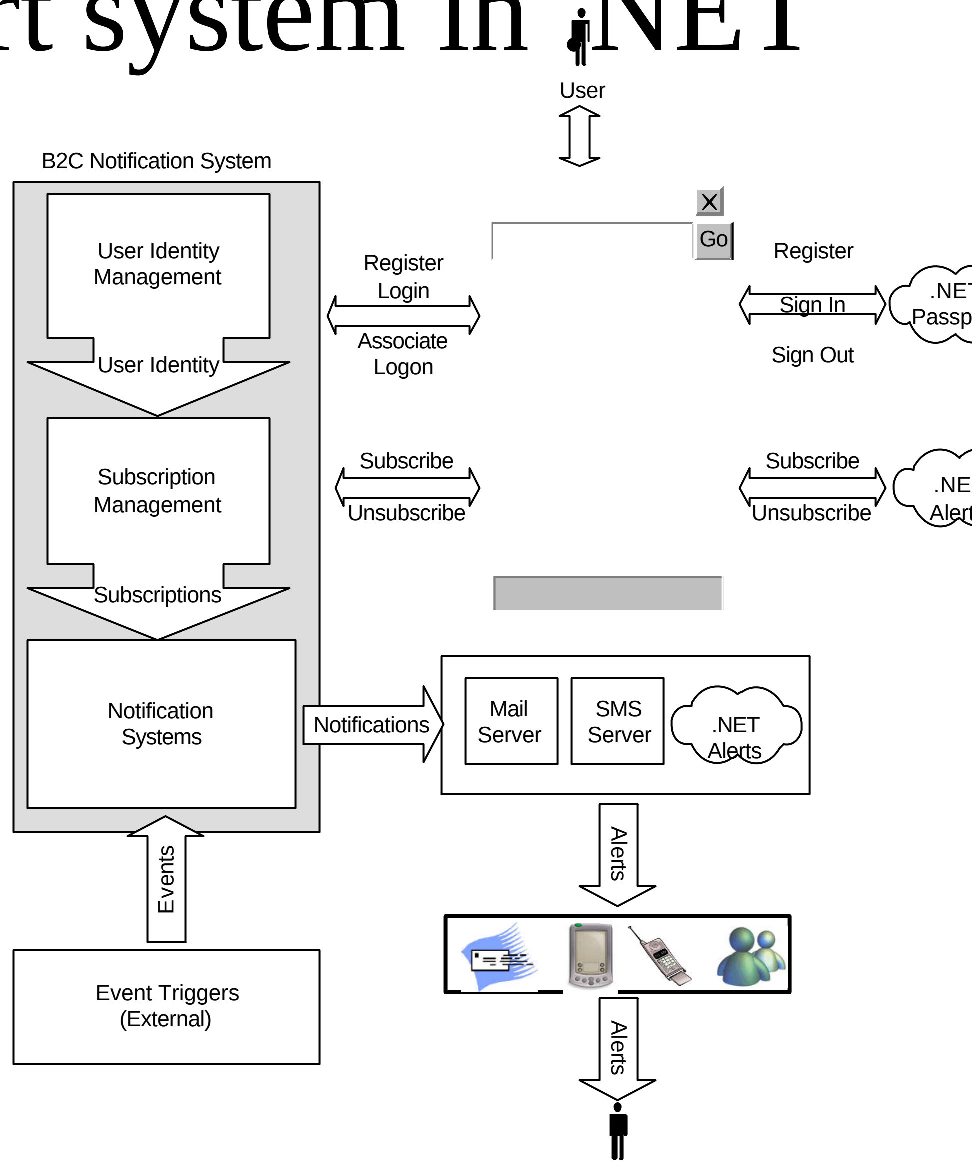
.NET web service platform



VAASAN YLIOPISTO

Microsoft Alert system in .NET

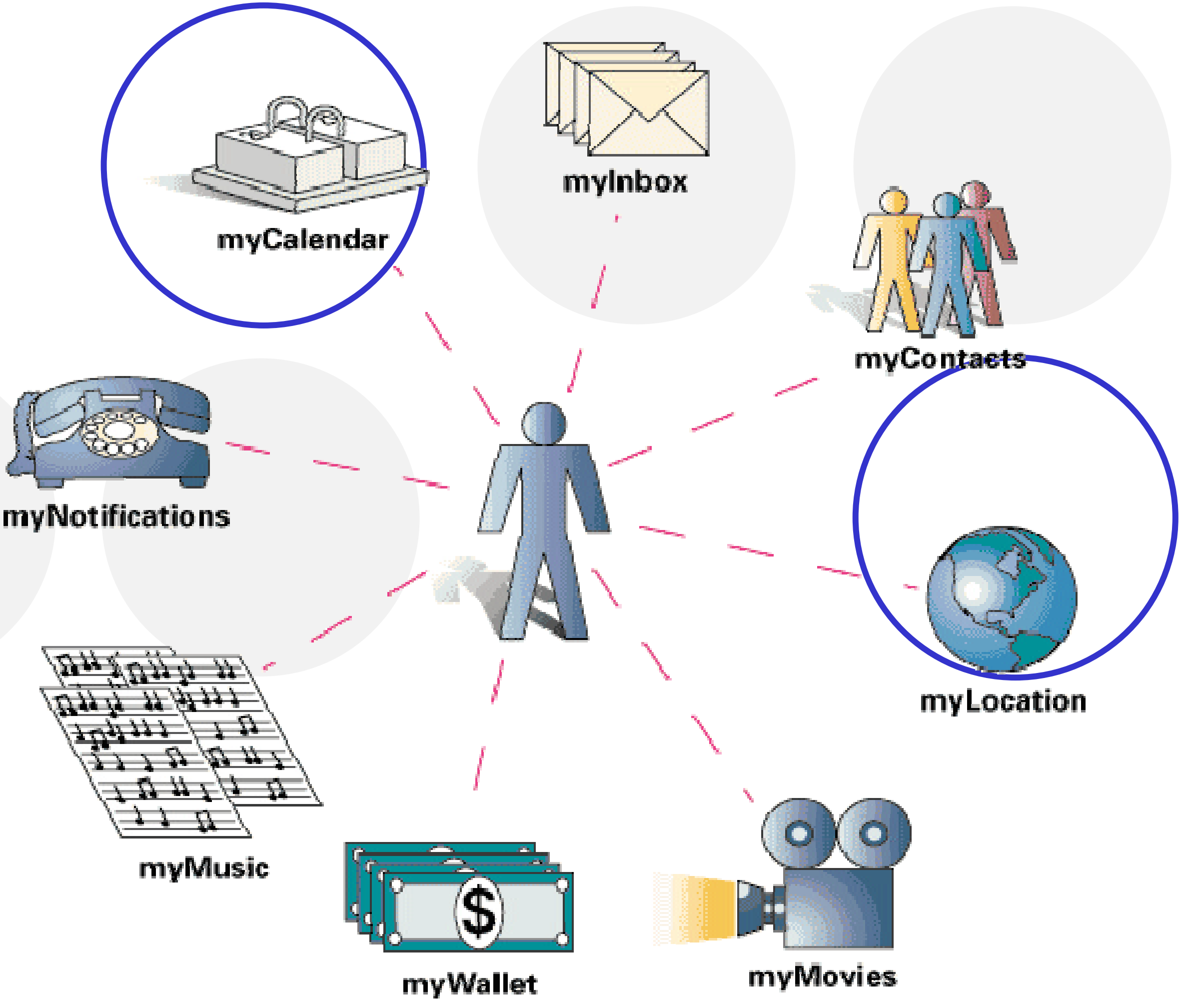
- Passport authenticated users subscribe to alert or notification services
- DLR context and it's changes create automated alerts



VAASAN YLIOPISTO

MyServices

- various Microsoft tools like
- .NET platform and Compact Framework for mobile applications
- utilizing W3C's XML, web service and semantic web standards



http://www.uwasa.fi/~ksa/ubi/case2_portal.htm

(Present) Tools for RDF

- IsaViz: A Visual Authoring Tool for RDF, http://www.w3.org/2001/11/IsaViz/
- Sesame server at http://www.openrdf.org/
- Jena – A Semantic Web Framework for Java, http://jena.sourceforge.net/
- HP SW research at http://www.hpl.hp.com/semweb/

VAASAN YLIOPISTO

IV Semantic web: CORE HOW

- technologies
- knowledge presentation
- knowledge usage
- knowledge-centric organization

VAASAN YLIOPISTO

SW technologies

- RDF
- RDF containers
- N3, reification, tools
- RDF Schema
- DAML+OIL to OWL
- non-contextual modelling

RDF

- RDF = Resource Description Framework

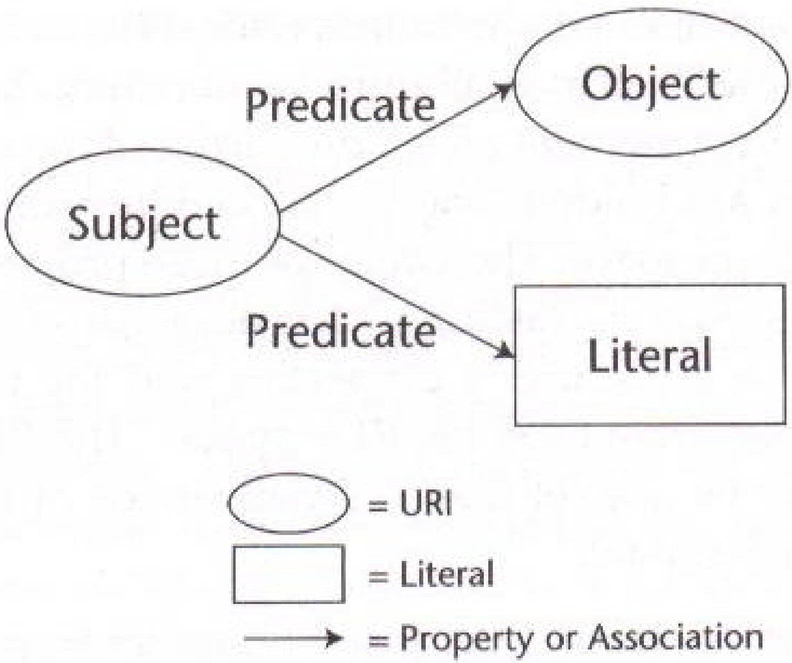


Figure 5.2 The RDF triple.

Ideas for RDF usage

- An RDF resource stands for either electronic resources, like files, or concepts, like “person”. One way to think of an RDF resource is as “anything that has identity”
- The resources in RDF must be identified by resource Ids, which are URIs with optional anchor Ids. This is important so that a unique concept can be unambiguously identified via a globally unique ID. This is a key difference between relying on semantics over syntax
- Capturing statements in a formal way allows slow aggregation of a corporate knowledge based in which you capture processes and best practices, as well as spot trends. This is knowledge management via a bottom-up approach instead of a top-down approach

RDF (XML serialisation)

- RDF Primer http://www.w3.org/TR/rdf-primer/

listing5-1.rdf - Notepad

File Edit Format View Help


```xml
<?xml version='1.0' encoding='ISO-8859-1'?>
  <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
      xmlns:rdfs="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#"
      xmlns:s0="http://www.w3.org/2000/PhotoRDF/dc-1-0#"
      xmlns:s1="http://sophia.inria.fr/~enerbonn/rdfpiclang#"
      xmlns:s2="http://www.w3.org/2000/PhotoRDF/technical-1-0#">
    <rdf:Description rdf:about="shop1.jpg">
      <s0:relation>part-of Store Front</s0:relation>
      <s0:type>image</s0:type>
      <s0:format>image/jpeg</s0:format>
      <s1:xmllang>en</s1:xmllang>
      <s0:description>Buddy Belden's work bench for TV/VCR repair</s0:description>
      <s2:camera>Kodak EasyShare</s2:camera>
      <s0:title>TV Shop repair bench</s0:title>
    </rdf:Description>
  </rdf:RDF>
```

RDF containers

- Three types of RDF containers are available to group resources or literals:
- **Bag:** An rdf:bag element is used to denote an unordered collection
- **Sequence:** an rdf:seq element is used to denote an ordered collection (a “sequence” of elements)
- **Alternate:** An rdf:alt element is used to demote a choice of multiple values or resources

VAASAN YLIOPISTO

Writing RDF: N3 notation

- N3 example of reification (Jane has tested Mary's web page and asserts that it passes the accessibility tests)
- http://www.w3.org/DesignIssues/Notation3.html
- http://infomesh.net/2001/05/notation3/

```
@prefix : <http://example.org/onto#>.
@prefix earl: <http://www.w3.org/2001/03/earl/0.95#>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix dc: <http://purl.org/dc/elements/1.1/>.

:Jane earl:asserts
     [ rdf:subject :MyPage;
       rdf:predicate earl:passes;
       rdf:object "Accessibility Tests" ];
   earl:email <mailto:Jane@example.org>;
   earl:name "Jane Jones".

:MyPage
   a earl:WebContent;
   dc:creator <http://example.org/onto/person/Mary/>.
```

Listing 5.4 N3 example of reification.

Reification “making statements about statements”

- The method of reifying statements in RDF is to model the statement as a resource via explicitely specifying the subject, predicate, object and type of the statement. Once the statement is modelled, you can make statements about the modeled statement
- The reification is akin to statements as arguments instead of statements as facts, which is useful in cases where the trustworthiness of the source is carefully tracked. This is important to understand, as reification is not applicable to all data modelling tasks. It is easier to treat statements as facts (!)
- Some current SW applications explicitly eliminate reification from their knowledge bases to reduce the complexity

Why is RDF not in the mainstream?

- **RDF doesn't yet play well with XML documents**
- There is a fairly esoteric issue regarding a difference between how XML Schema and RDF process namespaces. This has led many people to view RDF and XML documents as two separate paths for meta data
- (This is not true!: RDF is serialized as XML means that both XML Schema and RDF share common syntax, W3C works to embed RDF in XHTML and XML documents, for more see RDF in HTML: Approaches, http://infomesh.net/2002/rdfinhtml/ and SMORE tool)
- **Parts of RDF are complex:**
- More complex than XML, because of mixing metaphors (Table 5.1 next page), the serialization syntax (RDF syntax allows the RDF graph to be serialized via attributes or elements), and reification (another level of abstraction, matches natural language, but a foreign concept to all the other data communities!, with reification everything is just an assertion (and you must potentially follow a potentially infinite chain of assertions…))

RDF metaphors for its modeling primitives

- **Early RDF examples are weak**
- don’t highlight the unique characteristics of RDF
- Dublin Core DC, RSS even highlighted in the RDF Primer

METAPHOR	PART1	PART2	PART3
Language	Subject	Predicate	Object
Object-oriented	Class	Property	Value
Graph	Node	Edge	Node
Web link	Source	Link	Destination
Database	Entity	Relation	Entity

Table 5.1 RDF metaphors for modelling

How to see the real points of RDF (beyond the syntax!)?

- **Most RDF authors write their RDF assertions in N3 format and then convert the N3 to RDF/XML syntax via a conversion tool (like Jena's n3 program)**
- RDF literals can be types via XML Schema data types, or RDF/XML document integration in an RDF schema for DC at http://dublincore.org/documents/dcmes-xml/
- Another way to solve the validation problem is to have the namespace URI point to a document, which describes it as proposed by the Resource Directory Description Language (RDDL), http://RDDL.org
- For ideas see Make Your XML RDF-Friendly by Bob DuCharme, John Cowan, http://www.xml.com/pub/a/2002/10/30/rdf-friendly.html
- RDF Schema is a lightweight ontology vocabulary layer on RDF
- Noncontextual modelling makes RDF the perfect glue between systems and fixed data models

RDF Schema

- If we use the triple to denote class, class property, and value, we can create class hierarchies for the classification and description of objects. This is the goal of RDFSchema
- RDFSchema is a simple set of standard RDF resources and properties to enable people to create their own RDF vocabularies. The data model expressed by RDFSchema is the same data model used by object-oriented programming languages like Java. The data model for RDF Schema allows you to create classes of data

VAASAN YLIOPISTO

Key components of RDF Schema

- **rdfs:Class:** an element that defines a group of related things that share a set of properties
- **rdfs:label**
- **rdfs:subclassOf**
- **rdfs:Property:** In OOP, you define a class and everything it contains. In RDFS, you define properties and state what class they belong to

```
<?xml version='1.0' encoding='ISO-8859-1'?>
<!DOCTYPE rdf:RDF [
    <!ENTITY rdf 'http://www.w3.org/1999/02/22-rdf-syntax-ns#'>
    <!ENTITY example_chp5
'http://protege.stanford.edu/example-chp5#'>
    <!ENTITY rdfs 'http://www.w3.org/TR/1999/PR-rdf-schema-19990303#'>
]>
<rdf:RDF xmlns:rdf="&rdf;"
    xmlns:example_chp5="&example_chp5;"
    xmlns:rdfs="&rdfs;">
<rdfs:Class rdf:about="&example_chp5;Artifacts"
    rdfs:label="Artifacts">
  <rdfs:subClassOf rdf:resource="&rdfs;Resource"/>
</rdfs:Class>
<rdfs:Class rdf:about="&example_chp5;DesignDocument"
    rdfs:label="DesignDocument">
  <rdfs:subClassOf rdf:resource="&example_chp5;Artifacts"/>
</rdfs:Class>
<rdfs:Class rdf:about="&example_chp5;Employee"
    rdfs:label="Employee">
  <rdfs:subClassOf rdf:resource="&rdfs;Resource"/>
</rdfs:Class>
<rdfs:Class rdf:about="&example_chp5;Software-Engineer"
    rdfs:label="Software-Engineer">
  <rdfs:subClassOf rdf:resource="&example_chp5;Employee"/>
</rdfs:Class>

<!-- Classes SourceCode, System-Analyst, Technology, and Topic omitted
     for brevity.  They are similar to the above Classes. -->
```

Listing 5.6 RDF schema for Figure 5.9. *(continued)*

RDF/S

- **rdfs:domain**
- **rdfs:range**
- **rdfs:type**
- **rdfs:subPropertyof**
- **rdfs:seeAlso**
- **rdfs:DefinedBy**
- **rdfs:comment**
- **rdfs:Literal**
- **rdfs:XMLLiteral**

```
<rdf:Property rdf:about="&example_chp5;knows"
    rdfs:label="knows">
  <rdfs:domain rdf:resource="&example_chp5;Employee"/>
  <rdfs:range rdf:resource="&example_chp5;Topic"/>
</rdf:Property>
<rdf:Property rdf:about="&example_chp5;writes"
    rdfs:label="writes">
  <rdfs:range rdf:resource="&example_chp5;Artifacts"/>
  <rdfs:domain rdf:resource="&example_chp5;Employee"/>
</rdf:Property>
</rdf:RDF>
```

Listing 5.6 *(continued)*

in OOP we are going down from the class to the properties. In RDFS, we are going up from the properties to the class

Editing RDF/S

- Protégé open source ontology editor at http://protege.stanford.edu/
- After modelling the classes, Protégé allows you to generate both the RDF Schema and an RDF document if you create instances of the Schema (tab labelled “Instances” in the Protégé window)

SMORE editor

- SMORE Semantic Markup, Ontology and RDF Editor, http://www.mindswap.org/~aditkal/editor.shtml
- SMORE allows to embed RDF markup inside of HTML documents during the HTML authoring process

```
<html>
  <head>
  <script type="application/rdf+xml">
   <?xml version="1.0"?>
   <rdf:RDF
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:general1.0="http://www.cs.umd.edu/projects/plus/DAML/onts/gen-
eral1.0.daml#"
   xmlns:personOnt="http://www.wam.umd.edu/~mhgrove/personOnt.rdf#">
     <general1.0:Organization rdf:ID="Virtual_Knowledge_Base_">
          <general1.0:subOrganizationOf>JIVA</general1.0:subOrganiza-
tionOf>
     </general1.0:Organization>
     <general1.0:Organization rdf:ID="JIVA">
          <general1.0:subOrganizationOf>DIA</general1.0:subOrganiza-
tionOf>
     </general1.0:Organization>
     <personOnt:Person rdf:ID="Ted_Wiatrak"></personOnt:Person>
     <personOnt:Person rdf:ID="Danny_Proko"></personOnt:Person>
   </rdf:RDF>
  </script>
  </head>
  <body>
    <p>
     <b>Virtual Knowledge Base (VKB) </b>
    </p>
    <!-- omitted for brevity. -->
  </body>
</html>
```

Listing 5.9 RDF embedded in HTML (via SMORE).

VAASAN YLIOPISTO

What is Non-contextual Modeling

- Two key aspects of noncontextual modelling:
- **Non-contextual modelling uses explicit versus implicit relationships:** XML ducments create a hierarchy of name/value pairs. XML does not state the relationship between the name and the value (except implicitly!). On the contrary, RDF uses an explicit relationship between the name and the value with the triple structure: subject, predicate, and object
- **A graph is less brittle than a tree:** RDF graphs can be robust in the face of change and suffer less from the bridle data problem and need for versioning and compatibility issues that can plague XML documents
- Why RDF model is different from the XML model by T. Berners-Lee, http://www.w3.org/DesignIssues/RDF-XML.html
- Order us often very important in a document but not important to an RDF graph

to contextualize or not?

- The question is weather your specific application is better served by fixing the context or not fixing the context. In some ways this is the classical trade-off between flexibility in the face of change versus reliable execution via static processes. When the environment is stable and the volume is high, it is both easier and more efficient to strictly fix the context of documents and messages to reduce the errors and increase throughput. In the opposite case flexibility and noncontextual modelling are the best choice
- RDF takes the trend toward composable context to its logical conclusion. How does RDF implement noncontextual modelling? RDF creates a collection of statements and not a document. Therefore, the context of a set of RDF statements cannot be determined beforehand; instead, it is wholly dependant on the statements themselves and the relationships between the sentences. In a sense, this disconnection between a list of statements and a hierarchical tree is the root cause of the difficulty in encolding RDF in RDF/XML syntax, because it attempts to marry a list of statements with a hierarchical tree structure

VAASAN YLIOPISTO

RDF, TAP etc.

- In this example the RDF captures statements about the organizations, suborganizations, and people discussed in the HTML page
- TAP project at Stanford, http://tap.stanford.edu/ for coherent semantic web
- TAPache is a module for the Apache HTTP server that enables you to publish RDF data via a standard web service called getData(). This allows easy integration of distributed RDF data
- What the Semantic Web is not - answering some FAQs of the unconvinced by T. Berners-Lee, http://www.w3.org/DesignIssues/RDFnot.html

VAASAN YLIOPISTO

Knowledge presentation

- ontology spectrum
- taxonomies
- ontologies
- syntax, structure, semantics and pragmatics
- logic and logics

Ontology spectrum

Ontology can be…

a Taxonomy

a Thesaurus (words and synonyms)

a Conceptual Model (with more complex knowledge)

a Logical Theory (with very rich, complex, consitent, meaningful knowledge)

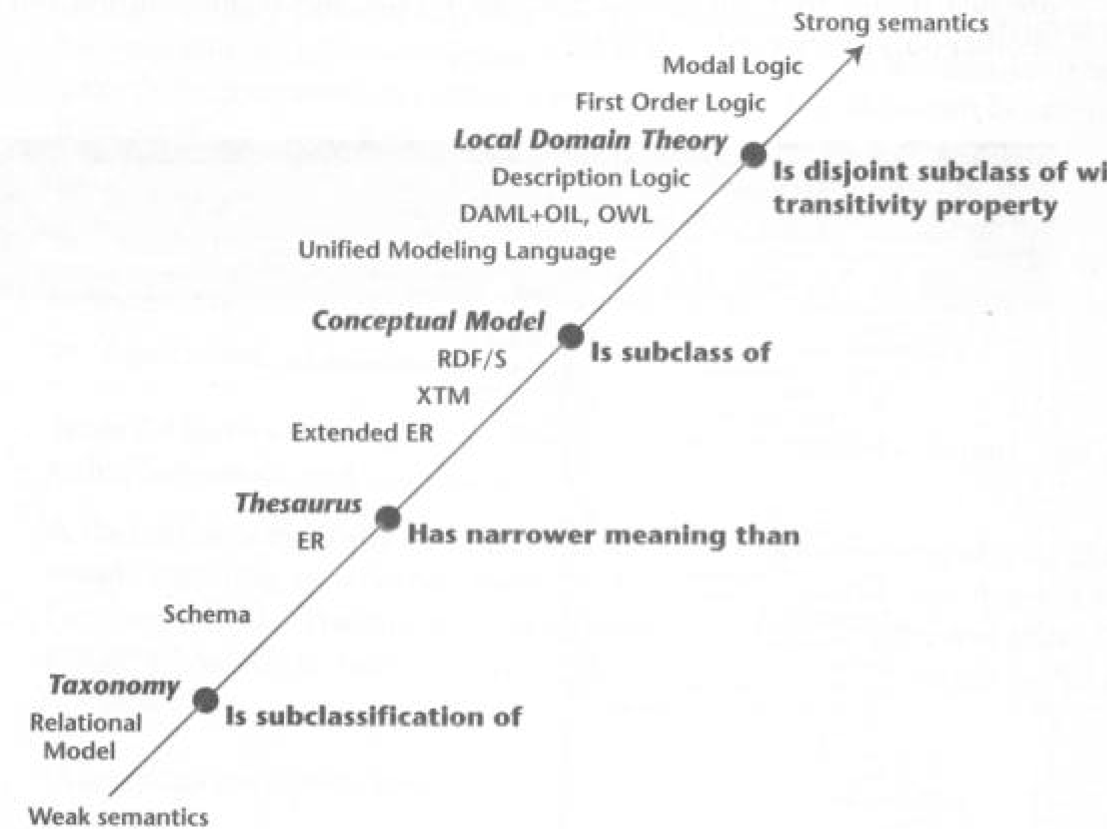


Figure 7.5 The ontology spectrum: Weak to strong semantics.

VAASAN YLIOPISTO

Taxonomies

- knowledge with minimal hierarchic or parent/child structure
- definition of a taxonomy: the classification of information entities in the form of a hierarchy, according to the presumed relationships of the real-world entities that they present

Ontologies

- An ontology defines the common words and concepts (the meaning) used to describe and represent an area of knowledge
- An ontology is an engineering product consisting of “a special vocabulary used to describe [a part of] reality, plus a set of explicit assumptions regarding the intended meaning of that vocabulary”- in other words, the specification of a conceptualisation
- When describing an are of knowledge- a domain- we describe the important things in the domain, their properties, and the relationships among the things. If we were to elaborate our description, we may even include rules about the domain

DAML+OIL to OWL

- DARPA Agent Markup Language DAML+OIL by DARPA, http://www.daml.org/
- http://www.w3.org/TR/daml+oil-reference
- Web Ontology Language OWL, http://www.w3.org/2001/sw/WebOnt/ W3C standard
- Both DAML+OIL and OWL also directly use XML Schema data types
- Feature comparison of RDF/S, DAML+OIL and portions of OWL, see http://www.daml.org/language/features.html

VAASAN YLIOPISTO

OWL

- Ontologies like OWL are layered on top of RDF
- Many see ontologies as the killer application for the semantic web and thus believe they will drive the adoption of RDF
- OWL has classes (and subclasses), properties (and subproperties), property restrictions, and both class and property individuals
- Class constructs such as subClassOf, disjointWith, intersectionOf, unionOf, complementOf

VAASAN YLIOPISTO

OWL: Ontology representation levels

LANGUAGE LEVEL	DESCRIPTION
OWL Full	The complete OWL. For example, a class can be considered,both as a collection of individuals and an individual itself
OWL DL (description logic)	Slightly constrained OWL. Properties cannot be individuals, for example. More expressive cardinality constraints
OWL Lite	A simple language, but one that is more expressive than RDF/S. Simple cardinality constraints only (0 or 1)

- Level 1- The knowledge representation level
- Level 2- The ontology concept level
- Level 3- The ontology instance level

Ontology tools

- Ontoedit ontology editor, http://www.ontoknowledge.org/tools/ontoedit.shtml
- OilEd, http://oiled.man.ac.uk/

Accessing SW by machines

- DAML+OIL and OWL have a logic behind them, a logic that is almost but not quite as complicated as first-order predicate logic (description logics explicitly try to achieve a good trade-off between semantic richness and machine tractability)
- ontologies modelled in those languages can be machine-interpretable: the machine knows exactly what the model means and how the model works logically, and can infer in a step-by-step fashion those inferences a human would make
- But you need not worry about the formal logic behind those languages. You just use the languages like OWL to create your ontologies, and the OWL interpreter will do the right thing

Syntax, structure, semantics …

- Ontologies try to limit the possible formal models of interpretation (semantics) of those vocabularies to the set of meanings you intend
- Ontologists want to shift some if that "semantic interpretative burden" to machines and have them eventually mimic our sematics- that is, understand what we mean-and so bring the machine up to the human, not force the human to the machine level
- By machine semantic interpretation, we mean that by structuring (and constraining) in a logical, axiomatic language (i.e., a knowledge representation language) the symbols humans supply, the machine will conclude via an interference process (again, built by the human according to logical principles) roughly what a human would in comparable circumstances

…and pragmatics

- Pragmatics sits above semantics and has to do with the intent of the semantics and actual semantic usage
- Intelligent agents will have to deal with the pragmatics (think of pragmatics as the extension of the semantics) of ontologies
- Agent communication Language is based on speect act theory, which is a pragmatics theory about human discourse that states the human beings express their utterances in certain ways that qualify as acts, and that they have a specific intent for the meaning of those utterances
- Intelligent agents are sometimes formalized in a framework called BDI for Belief, Desire, and Intent

VAASAN YLIOPISTO

Extension and Intension: E & I

- In the database and formal/natural language worlds, the first type of knowledge is the intension and the second the extension
- In the database world, a schema is the intentional database, whereas the tuples of the database constitute the extensional database
- In the formal/natural language worlds, a description or specification is an intension, whereas the actual objects (instances/individuals) in the model (or world) for which the description is true are in the extension

E & I in ICT

- Now the various technical communities will call the intension the following: a taxonomy, a schema, a conceptual/object model, an intensional semantics, an ontology. They will call extension the following, respectively: leaves of the taxonomy, tuples, instances, the extension, instances/individuals
- so overall, in an ontology you describe a set of structured, generic properties that have a particular semantics (meaning). This is called a model, meaning that it defines and represents information about some aspects of the world that you (as the modeller) care to model
- If you are model-driven (meaning here ontology- or knowledge-driven), just means you can change your model, regenerate the implementation, or find the delta, and continue

Ontology mapping problem

- The Ontology or semantic mapping problem is an issue that affects everything in information technology that must confront semantics problems- that is, the problem of representing meaning for systems, applications, databases, and document collections
- You must always consider mappings between whatever representation of semantics you currently have (for system, application, database, document collection) and some other representation of semantics (within your own enterprise, within your community, across your market, or the world)
- And you must consider semantic mappings within your set of ontologies or whatever your semantic base representation is (if it's not ontologies, it's probably hard-coded in the procedural code that services your databases, and that means it's really a problem)

VAASAN YLIOPISTO

Logic and logics…

- Ontologies provide two kinds of knowledge:
- About the class or generic information that describes and models the problem, application, or, most usually, the domain
- About the instance information- that is, the specific instantiation of that description model

VAASAN YLIOPISTO

…with language…

- The representation is a means for both expressing and using information
- The language used for knowledge representation determines the kind of reasoning that can take place on the knowledge; the representation precedes reasoning

…in science and the world

- Logic is sometimes supposed to underlie all of mathematics and science. Some say that logic also underlies all of natural language. We will remain agnostic on these pronouncements and will just say that logic usually and definitely should underlie all models and modelling languages. Why?
- Because if we are serious about defining languages that can both represent the knowledge of the world according to the perspective of the human being and be machine-interpretable at semantic level (i.e. machines and their software can interpret human semantics and knowledge at our human level of understanding),
- then those knowledge representation languages and the knowledge they represent must be supported by formally powerful tools only representable by logic. Otherwise our knowledge- if represented in onlogically underpinned ways- will remain arbitrarily interpretable by our software, the condition that holds today, where the semantics of our data and systems are embedded indecipherably and inextricably in our imperative programming code

Ontologies today

- Upper ontology characterize very basic commonsense knowledge notions that humans know so well we typically don't know we know them i.e. common generic information that spans all ontologies

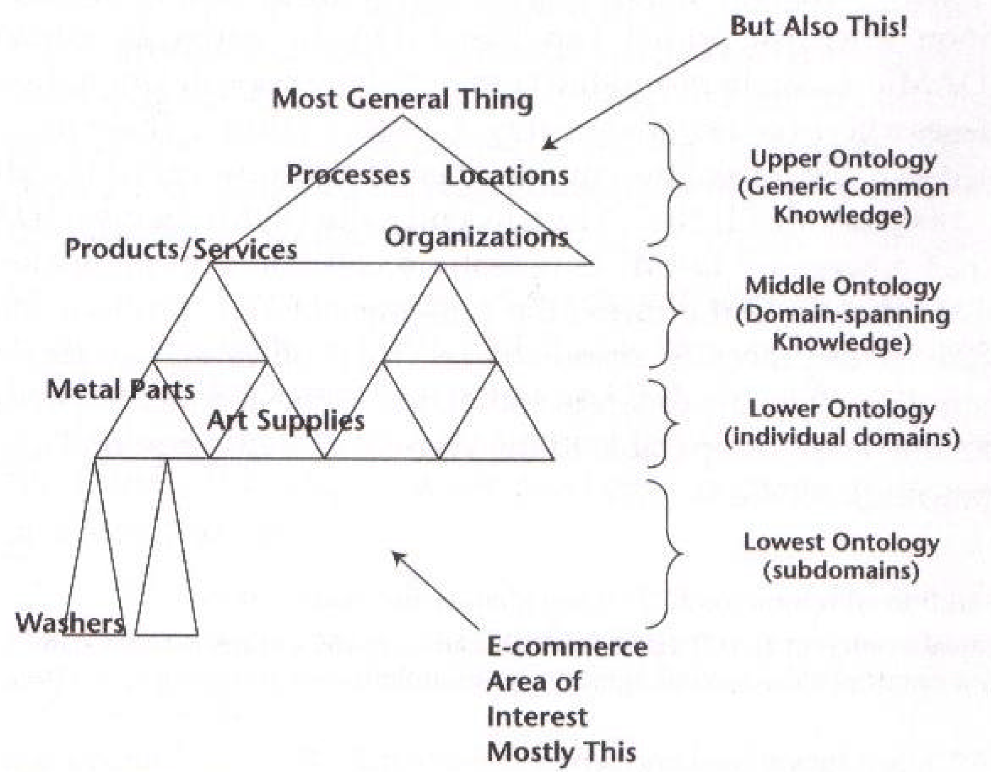
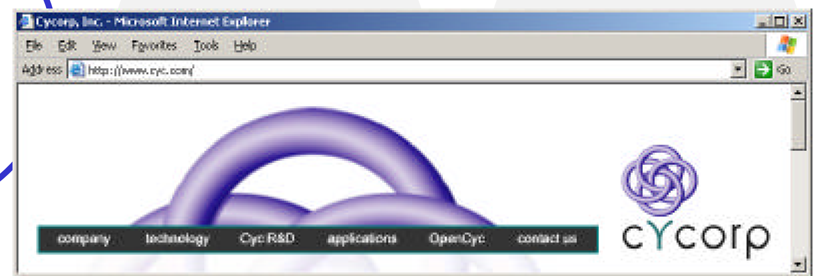


Figure 8.11 Ontology levels.

CYC Inc, http://www.cyc.com

- The middle ontology represents knowledge that spans domains and may not be as general as the knowledge of the upper level
- Finally, the lower levels represent ontologies at the domain or subdomain level



CYC open, http://www.cyc.com/cyc/opencyc

VAASAN YLIOPISTO

Knowledge-centric organization

- To establish enterprise- or community-wide common semantics does not require a common semantics or common model (a monolithic ontology) across the enterprise or community,
- but instead a set (or probably more accurately, a lattice) of integrated ontologies: upper, middle, and domain (or subdomain) levels integrated logically and thus not all in the same namespace and all contexts not the same,
- and all applications not using the same portions of the lattice of ontologies

VAASAN YLIOPISTO

Producing knowledge on the web

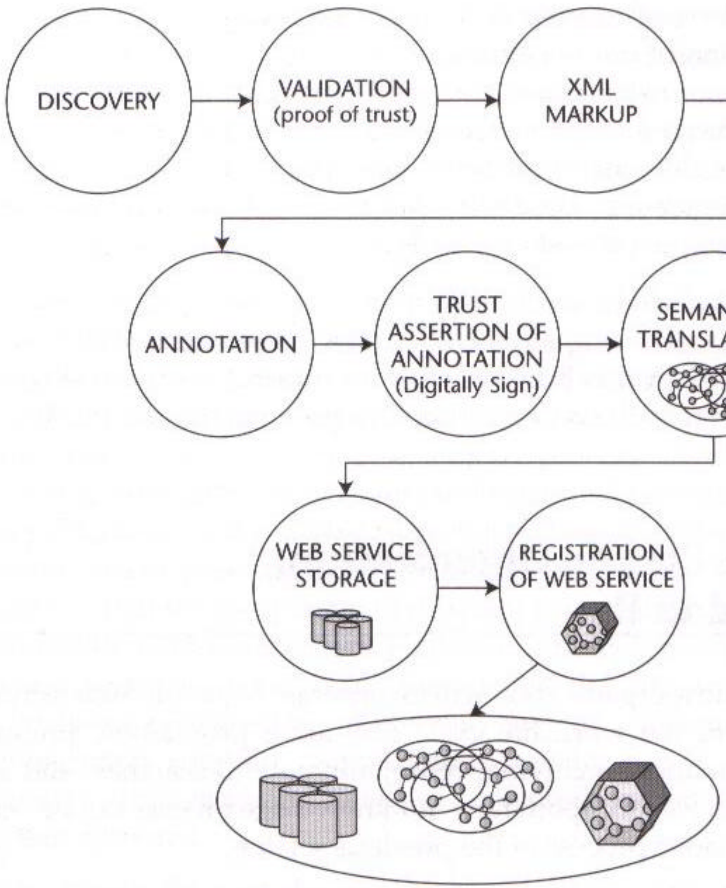


Figure 9.3 The discovery and production process.

Knowledge usage example: automated context handling with RDF and OWL

- work of Harry Chen, Tim Finin, Anupam Joshi: Semantic Web in the Context Broker Architecture
- further automation of context needs data that applications can understand and manipulate
- http://pervasive.semanticweb.org/ by UMBC

Communication example

- In a pervasive computing environment, sensors are often used to detect the presence of people in a building
- For example, RFID (Radio Frequency Identification) sensors can detect the presenc of Smart Tags and conclude th presence of people who wear them, and Bluetooth sensors can detect the proximity presence of the Bluetooth-enabled personal devices and conclude the presence of the device owners

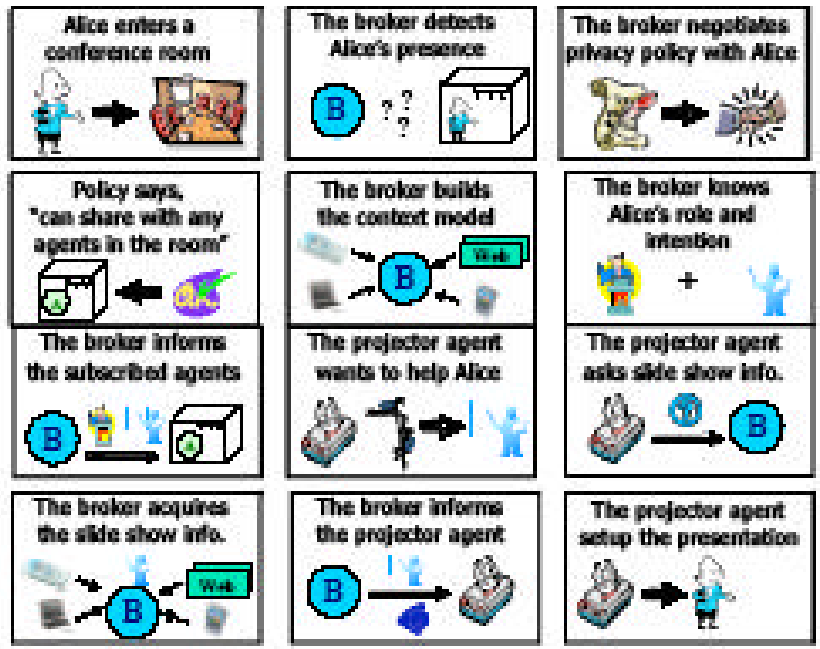


Figure 1. A typical scenario from an intelligent briefing

COBRA ontology and architecture

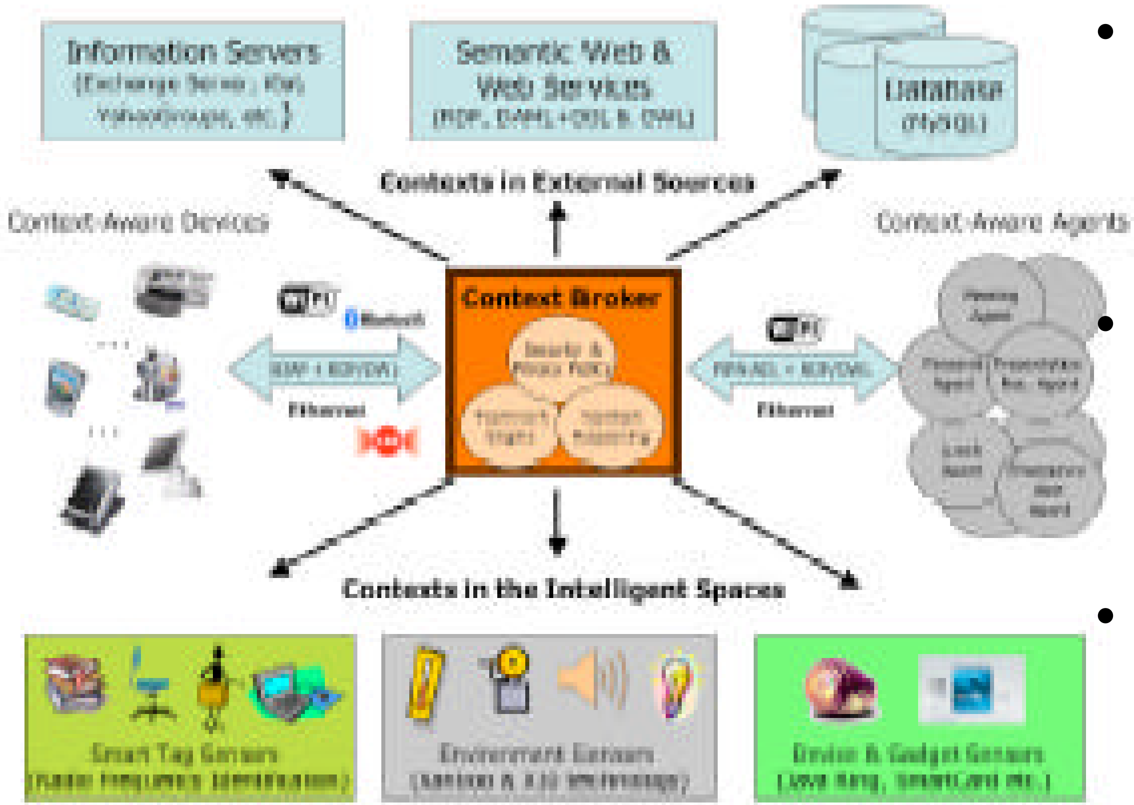


Figure 2. In our approach an intelligent context broker acquires context information from devices, agents and sensors in its environment and fuses it into a coherent context model, which is then shared with the devices and their agents.

- building context into the applications and environments
- based on extending the web information archirecture
- using independent agent (applications!) for information and user exchange needs

•http://cobra.umbc.edu

Use case: people presence sensors

- Using the CoBrA ontology, these people presence sensors can effectively share people presence information with the broker in the system and enable the broker to reason about the situational contexts of these people. For example,
- 1. Whether a person is in the building,
- 2. Whether a person is in school today, and
- 3. Whether a person is not in a room (e.g., in hallway or in a cafeteria).
- Figure 5 shows an example of the person presence information that is sent to the broker. Upon receiving this information, the broker will reason about Harry Chen’s context. The following three examples describe how the broker may reason about his contexts.

VAASAN YLIOPISTO

Example 1:To determine if Harry Chen is in the ECS Building

- A1: Person("Harry Chen") has property isCurrentlyIn("ECS210I").
- A2: For any person who has the property isCurrentlyIn() with rdfs:range limited to any Place that isPartOf Building, that person must be a type of PersonInBuilding (i.e., that person is in a building).

VAASAN YLIOPISTO

cont.

<cobra:Person rdf:about
="http://www.cs.umbc.edu/people/hchen4">
<cobra:isCurrentlyIn rdf:resource
="http://www.cs.umbc.edu/ECS210I"/>
</cobra:Person>

- Figure 5. When Harry Chen enters Room ECS210I and swipes his RFID badge at the door, the RFID sensor informs the broker of his presence in the room
- A3 <= A1+A2: Person("Harry Chen") is a type of the PersonInBuilding class (i.e., Harry is currently in a building). Furthermore, because Room("ECS210I") is-PartOf the Building("ECS"), the broker can deduce Harry is currently in the ECS building

Example 2: To determine if Harry Chen is in school today

- B1: Person("Harry Chen") is in Building("ECS"). (From Example 1: A3)
- B2: Building("ECS") isPartOf UniversityCampus("UMBC")
- B3 <= B1+B2: Person("Harry Chen") is in school today

Example 3: To determine if Harry Chen is NOT in any rooms in the ECS building.

- For example, he is talking to someone in the hallway or has just left the meeting
- C1: Person("Harry Chen") is in Room("ECS210I") &Building("ECS"). (From Example 1: A3)
- C2: If a person has property isCurrentlyIn with value that is a type of OtherPlaceInBuilding, then that person is not currently in a room. The class OtherPlaceInBuilding rdfs:disjointWith the class Room.
- C3 <= C1+C2: It is false that Person("Harry Chen") is NOT in a room in the ECS building

Use case: a room agent

- In an intelligent space, room agents will play an important role in maintaining and sharing room-specific contexts with devices and agents. Let’s assume in each room, there is a room agent maintains a set of specific contexts of the room, for example,
- 1. Whether the room is currently hosting a meeting
- 2. The temperature, noise level, and light intensity level in the room
- 3. The close/open states of the doors and windows in the room
- 4. The type of devices/services that are available in the room

VAASAN YLIOPISTO

cont.

- As the context of the room changes, the room agent will inform the broker of the updated contexts. Figure 6 shows an example of the information that is sent to the broker from the room agent. From this information, the broker can reason about additional context of the room and the context of people in the room. These contexts may include: 1) whether a person is currently in a meeting place, and 2) whether a person is a meeting participant of a particular meeting. The following two examples show how the broker may reason about these contexts.

Example 5:

- **To determine if Harry Chen is currently in a meeting place in the ECS building.**
- E1: Person("Harry Chen") is in Room("ECS210I") and Building("ECS") (from Example 1: A3)
- E2: For any room that has the property hostsMeeting() with rdfs:range limited to Meeting, the room must be a type of MeetingPlaceInBuilding (see cobra-ont.owl).
- E3: Room("ECS210I") has the property hostMeeting("me239").
- E4 <= E2+E3: Room("ECS210I") is a type of Meeting-PlaceInBuilding
- E5: If a person is currently in a room, and that room is a type of MeetingPlaceInBuilding, then that person is currently in a meeting place.
- E6 <= E1+E4+E5: Person("Harry Chen") is currently in a meeting place which is in the ECS building

VAASAN YLIOPISTO

Example 6:

- **To determine if Harry Chen is attending a meeting in ECS210I (i.e., is Harry Chen a meeting participant).**
- F1: Person(“Harry Chen”) is in Room(“ECS210I”) (From Example 1: A3)
- F2: Room(“ECS210I”) is a type of MeetingPlaceIn-Building. (From Example 5: E4)
- F3: If a person has the property isCurrentlyIn() with a value that is a type of Room class, then that person is a type of MeetingParticipant (i.e., that person is a meeting participant).
- F4 <= F1+F2+F3: Person(“Harry Chen”) is a meeting participant.

cont.

<cobra:Room rdf:about
="http://www.cs.umbc.edu/ECS210I"/>
<cobra:hostsMeeting rdf:resource
="http://www.ittalks.org/me293"/>
</cobra:Room>

- Figure. 6 A meeting is scheduled to take place in ECS210I at 11:00am. Few minutes before the meeting, the room agent of ECS210I informs the broker that the room is about to host a meeting.

VAASAN YLIOPISTO

Use case: a person agent

- Person agents are specialized agents that provide personalized services for individual users [9]. In intelligent spaces, these agents will keep track of users' profiles, preferences, desires and intentions. For example, the person agent of a speaker will automatically set up presentation slides when the speaker arrives at the meeting and adjust room lighting when the presentation starts. In order for the person agent to provide these services, it must acquire contextual knowledge about the person from the broker. This knowledge may include the following:
- **1.** The role of the person in the meeting
- **2.** The type of services that the person has access to
- **3.** The type of the devices that the person carries
- **4.** The type of non-computing objects the person's vicinity (e.g., the type of clothes the person wears & the type of objects that the person holds)
- **5.** The time at which the person enter the room or joins the meeting
- **6.** The identity of people whom the person is talking to

```
<cobra:Person rdf:about
  ="http://www.cs.umbc.edu/people/hchen4">
  <cobra:hasIntention rdf:resource
    ="&cobra;#GiveSlideShowPresentation"/>
</cobra:Person>
```

- One source from which person agents can acquire information about their users is through user behavior monitoring. For example, Harry Chen is scheduled to talk about ontology development at Wednesday's meeting. Days before the meeting, while Harry prepares his PowerPoint slides, his personal agent learns his intention to give presentation at the meeting. On the day of the meeting, as Harry enters the conference room, the personal agent informs the broker of Harry's intention and queries the broker for Harry's situational contexts.
- Figure 7 shows an example of the information that is sent to the broker from the person agent. Upon receiving information from a person agent, the broker will reason about the context of the user. Sometimes ontology reasoning may involve uncertainty. For example, knowledge about the context of a person may not always be completely accurate. The following examples show how reasoning about the role of a person can involve varied degree of certainty.

Example 7: To determine the role of a person

- (e.g., is Harry Chen is the speaker of meeting “me239”)
- G1: Person(“Harry Chen”) is the same person as MeetingParticipant(“Harry Chen”) (From Example 6: F4)
- G2: MeetingParticipant(“Harry Chen”) is associated with Meeting(“me239”) (From Example 5 & Example 6)
- G3: Person(“Harry Chen”) has the intention to GiveSlideShowPresentation.(Informed by Harry Chen’s person agent)
- G4: If a person is a type of MeetingParticipant and that person has owl:oneOf the SpeakerIntention, then that person is **LIKELY** to be a speaker.
- G5 <= G1+G2+G3: Person(“Harry Chen”) is likely to be a speaker.
- Now, let’s assume the broker has some prior knowledge about the invitations that are given to meeting
- participants. For example, from a talk announcement server (e.g., ITTalks.ORG [8]), the broker learns that some person who is a type of TalkEventHost has invited Harry Chen to the meeting “me239” (see Figure 8). This information can increase the certainty about the role of Harry Chen being a speaker

cont.

- G6: If G5 is true, and the person who in question is invited by some TalkEventHost, then that person **MUST BE** a speaker.
- G7: Person(“Harry Chen”) is invited by a TalkEventHost.
- G8 <= G6+G7: Person(“Harry Chen”) must be a speaker.
- the next version of the CoBrA ontology has additional concepts and vocabularies to model other detail aspects of meetings and potential services in the environment
- a prototype an ontology reasoning component for building a Context Broker. The prototype will exploit TRIPLE and Jess

VAASAN YLIOPISTO

Goals to implement in SW

- 1. set detailed technical goals
- mark up your documents in XML
- expose your applications as web services
- build web service orchestration tools
- establish corporate registry
- build ontologies
- use tools that will help your production process
- integrate search tools
- use an enterprise portal as a catalyst for knowledge engineering
- 2. develop a plan with a workflow change strategy
- 3. set appropriate staff in place
- 4. set a schedule

Conclusions

- In a nutshell, think of RDF as semantic glue to link your XML-marked-up documents to your taxonomy (directory tree) and ontology (formal class model showing relationships)
- a document will be XML inside, RDF outside, filed in a branch of the taxonomy and related to classes in the ontology

References

1. Semantic web http://www.w3.org/2001/sw/
2. M. C. Daconta, L. J. Obrst, K. T. Smith: The Semantic Web: A guide to the future of XML, Web Services and Knowledge Management, John Wiley, 2003
3. Microsoft .NET, http://www.microsoft.com/net/
4. Harry Chen, Tim Finin, Anupam Joshi: Semantic Web in the Context Broker Architecture, http://cobra.umbc.edu/
5. SW Working Symposium, http://www.semanticweb.org/SWWS/, 2001

…

- (Sampath Kumar (FH-Kiel): Zone based mobile communication using web based portals, project work at Technobothnia, 2004)
- Kontti project, http://akseli.tekes.fi/Resource.phx/tivi/nets/z-07335329.htx
- K. Salmenjoki and T. Welzer : Using Web Services and Semantic Web for Producing Intelligent Context-Aware Services, KES2004