OSM Lecture (14:45-16:15)
Takahira Yamaguchi

OSM Exercise (16:30-18:00)
Susumu Tamagawa
TBL 1st Proposal
Information Management: A Proposal (1989)

• Links have the following types:
  – depends on
  – is part of
  – made
  – refers to
  – uses
  – is an example of

• Users should do data analysis by WWW:
  – Automatic analysis such as comparison
  – Automatic generation of mailing lists
  – Deriving real organization structure
TBL 2nd Proposal (2001)
Tim Berners Lee, James Hendler, Ora Lassila: The Semantic Web,
http://www.sciam.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21&sc=I10032

• Semantic Web
  – Make Web real Hypertext
  – Data(instance) should have types (class)
  – Links(property) should have types
  – Links = relationship between resources

• Data Analysis
  – SW can make inference

“The Semantic Web will globalize KR,
just as the WWW globalized hypertext
Intelligence

• Machine Intelligence
  → Knowledge Representation
    (1980-: Expert Systems
     1990-: Knowledge Modeling
     -> Ontologies)

• Web Intelligence
  → (2000-: Semantic Web)
Expert System Overview

Major Components
- Knowledge Base (KB)
- Inference Engine (IE)
- Working Space (WS)
- Knowledge Acquisition (KA)

Other Components
- Explanation Facility (EF)
- Dialog Interface (DI)

(Domain Expert)
Technical systems in narrow domain knowledge

not been supplanted as knowledge providers

difficult to maintain or add knowledge (rules) (Authorizer’s Assistant ES for credit authorization
→ knowledge is stable and so still used in AMEX XCON →knowledge changes and so DEC stopped using it)

Why so hard to maintain rules?
→ other experts besides developer do not understand what rule primitives mean. So they should be specified.
→ ontologies (in the filed of knowledge engineering)
Ontologies in KE and Semantic Web

KE
- 90-: an explicit specification of conceptualization by Tom Gruber 93
- Ontolingua
- General Ontology CYC, WordNet, EDR...
- Ontology Development Methodology

Semantic Web
- 95-97: XML as arbitrary structures
- 97-98: RDF
- 98-99: RDFS
- 2001: DAML+OIL 2nd proposal by TBL
- 2004.2.10: OWL
- 2009.10 OWL2
- 2010.7 RIF
See Movie on Semantic Web
Introduction (4 minutes)
RDF
(Resource Description Framework)
RDF Basic (Data) Model

Resource: information resource identifiable by URI
Property: relationship between resources or resource’s attribute

Statement = (Subject, Predicate, Object)
(Tuple)
① (Resource, Property,(relationship), Resource)
② (Resource, Property(attribute), Value)
RDF Basic Model (2)

- RDF Data Model -> Arced Graph with Labels
  - Subject, Object: Node
  - Predicate: Arc (Property)
  - Resource: Ellipse (Subject, Object)
  - Literal: Rectangle (Object)

http://www.yamaguti.comp.ae.keio.ac.jp/

Takahira Yamaguchi

Statement
RDF Structured Model (1)

Putting together two or more statements
Some resource is object in statement 1 and subject in statement 2.

Ueda breed Tama
Ueda breed Tama

sex
male

color
white
http://www.bb2.com’s author is Ryou Imai. His mail address is webmaster@imai.com.
RDF Syntax (1)

RDF Syntax XML

```xml
<rdf:RDF
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:dc="http://purl.org/dc/elements/1.1">
   <rdf:Description rdf:about="http://kanzaki.com">
     <dc:creator>Takahira Yamaguchi</dc:creator>
   </rdf:Description>
</rdf:RDF>
```

RDF Syntax Notation3

```turtle
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
```

RDF Syntax N-Triple

```ntriple
<http://kanzaki.com>
```
RDF XML Syntax (1) Basic Model

rdf: RDF Model & Syntax name space
Description element: statement
About attribute: Subject resource’s URI
Description element: property(dc:creator) + value(Takahira Yamaguchi)

```
<rdf:Description rdf:about="http://yamaguchi.com">
  <dc:creator>Takahira Yamaguchi</dc:creator>
</rdf:Description>
```

Shorter Version

```
<rdf:Description rdf:about="http://yamaguchi.com" dc:creator="Takahira Yamaguchi"/>
```
RDF XML Syntax (2) Structured Model

```
<rdf:RDF ...>
    <dc:creator>Takahira Yamaguchi</dc:creator>
    <dc:publisher>
      <rdf:Description rdf:about=" http://www.ohmsha.co.jp ">
        <ex:webmaster>Someone’s name</ex:webmaster>
      </rdf:Description>
    </dc:publisher>
  </rdf:Description>
</rdf:RDF>
```
RDFS
RDF Scheme
If software does not make sense of the above names (symbols), there is no way to enable semantic-based intelligent processing. -> Ontologies (RDFS+OWL)
RDFS vocabulary

- RDFS 4 Basic Classes
  - rdfs:Resource:
  - rdfs:Class:
  - rdf:Property:
  - rdfs:Literal:

- RDFS 7 Basic Properties
  - rdf:type class-instance
  - rdfs:subClassOf superclass-subclass
  - rdfs:subPropertyOf super property-subproperty
  - rdfs:domain (what subject should be)
  - rdfs:range (what object should be)
    rdfs:label
    rdfs:comment

rdf: -> http://www.w3.org/1999/02/22-rdf-syntax-ns# prefix
rdfs: -> http://www.w3.org/2000/01/rdf-schema# prefix
How to make class

```xml
<rdf:Description rdf:ID="animal">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema #Class"/>
    or <rdf:type rdf:resource="&rdfs;Class"/>
    <rdfs:subClassOf rdf:resource="#living thing"/>
</rdf:Description>

<rdfs:Class rdf:ID="dog">
    <rdfs:subClassOf rdf:resource="#animal"/>
</rdfs:Class>
```
How to make “person” class

<rdf:Description rdf:ID="Person">
    <rdf:type rdf:resource="&rdfs;Class"/>
    <rdfs:subClassof rdf:resource="rdfs;Resource"/>
</rdf:Description>

<rdfs:Class ID="Person">
    <rdfs:subClassof rdf:resource="rdfs;Resource"/>
</rdfs:Class>
How to make property

- `rdfs:subPropertyOf` super property-subproperty
- `rdfs:domain` (what subject should be)
- `rdfs:range` (what object should be)

```xml
<rdf:Property rdf:ID="breed">
  <rdfs:subPropertyOf rdf:resource="..."/>
  <rdfs:domain rdf:resource="#person"/>
  <rdfs:range rdf:resource="#cat"/>
</rdf:Property>
```
How to make instance

**rdf:type** class-instance

```
<rdf:RDF
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#" >

   <rdf:Description rdf:about="#Tama">
       <rdf:type rdf:resource="#cat"/>
   </rdf:Description>

   <rdf:Description rdf:about="#Ueda">
       <rdf:type rdf:resource="#person"/>
       <breed rdf:resource="#tama"/>
   </rdf:Description>

</rdf:RDF>
```
RDF/RDFS

**RDF**

**Syntax**

```
<rdf:Description about="William">
  <hasFather>Charles</hasFather>
</rdf:Description>
```

**Model**

- Subject: William (Resource)
- Predicate: hasFather (Property)
- Object: Charles (Literal)

**RDFS**

**Syntax**

```
<rdf:Class rdf:about="Person">
  <rdfs:subClassOf resource="&rdfs;Resource"/>
</rdf:Class>
```

**Model**

- rdf:type: Person (Class)
- rdf:type: hasFather (Property)