Fundamental Terms of Knowledge Representation

**Knowledge**: Collection of multimodal content, skills, experiences and problem solving methods, providing the background for complex information processing.

**Knowledge Representation**: Operational as well as formal and therefore computer understandable description of knowledge.

**Knowledge Representation Language**: Formal language for systematic representation of knowledge.

**Representation Construction**: Subset of a knowledge representation language

**Knowledge Base**: Knowledge that can be used by AI System.

**Meta Knowledge**: Knowledge about knowledge in the knowledge base.

**Heterogeneous Knowledge**: Knowledge base using different knowledge representation languages to encode code knowledge units.

**Multiple Representation**: Representation of the same knowledge using different knowledge representation languages in the same knowledge bases.
Four Description Layers for Knowledge Representation Languages

1) Implementation Layer
- objects
- pointers
- ...

2) Logical Layer
- predicates
- quantifiers
- ...

3) Epistemological Layer
- inheritance relations
- structuring primitives
- ...

4) Ontological Layer
- primitive concepts
- primitive relations
Hierarchies of Ontologies

- Top ontology
- Domain ontology
- Utility ontologies (time, geospatial etc)
- Biological processes
- Annotations/metadata ontology (dublin core etc)

Test ontologie
Early Semantic Networks in HAM-RPM

Conceptual semantic network of HAM-RPM (v. Hahn, Wahlster)

Referential semantic network of HAM-RPM
The Terminological Box (TBox) and the Assertional Box (ABox) in Knowledge Representation

TBox
Father = (and Man Parent)

ABox
Father(a)
∀ x: Man(x) → ...
The TBox and ABox in Description Logics

• A **TBox (Terminological Box)** is a set of schema axioms (sentences), defining the vocabulary to describe situations in a domain

• An **ABox (Assertional Box)** is a set of data axioms (ground facts), describing a specific state of a domain e.g.:
  
  {John:HappyParent, John hasChild Mary}

• A **Knowledge Base (KB)** is just a TBox plus one or more ABoxes
Introduction to the Resource Description Framework

- RDF (Resource Description Framework)
  - from machine-readable to machine-understandable descriptions

- RDF covers a wide spectrum of applications
  - Semantic Web Foundation, Digital libraries, information extraction and integration, E-Commerce,...
  - Significant industrial support and W3C standard

- RDF consists of two parts:
  - RDF Model (a set of triples)
  - RDF Syntax (various XML syntactic constructs for serialization)

- RDF Schema allows the definition of vocabularies (simple ontologies for RDF (and in RDF))
The RDF Data Model

- **Resources**
  - A resource is an object, that one can refer to
  - Resources have URIs
  - RDF definitions are also resources

- **Properties**
  - Slots define relations to other resources or atomic values

- **Statements**
  - „Values are resources or atomic XML data“

- Similar to frame-based representation in AI (eg. FRL)
A Simple RDF Example

- Statement
  - "http://www.w3.org/Home/Lassila has the creator Ora Lassila"

- Structure
  - Resource (Subject) http://www.w3.org/Home/Lassila
  - Property (Predicate) http://www.schema.org/#Creator
  - Value (Object) "Ora Lassila"

- Directed Graph
A More Complex RDF Example

- In order to add another property to Creator, an intermediate resource is inserted.

```
http://www.w3.org/Home/Lasilla
```

```
s:Creator
```

```
Person://fi/654654635
```

```
Name
```

```
Ora Lasilla
```

```
Email
```

```
lassila@w3.org
```
Handling Collections in RDF

- Multiple occurrences of the same Property Types do not introduce a relationship between these values.
  - The Millers own a boat, a bike, and a TV set
  - The Millers need (a car or a truck)
  - (Sarah and Bob) bought a new car

- RDF defines three special Resources:
  - Bag unordered values rdf:Bag
  - Sequence ordered values rdf:Seq
  - Alternative single value rdf:Alt

- The core of RDF does not enforce a set-theoretic semantics for values
RDF Example for Bags

The students of course 6.001 are Amy, Tim, John, Mary und Sue
RDF Example for Alternative

- The source code for X11 can be found under ftp.x.org, ftp.cs.purdue.edu, or ftp.eu.net.
Statements about Statements

In order to make statements about *statements*, they must be transformed into resources:

- subject: the original referent resource
- predicate: the original Property Type
- object: the original value
- type: rdf:Statement
RDF Example for Reification

• Ralph Swick believes that Ora Lassila is the creator of the resource http://www.w3.org/Home/Lassila.
A Formal Model of RDF

• RDF has a very simple mathematical foundation:

  – Basic Definitions:
  
  • Resources
  • Properties ⊂ Resources
  • Literals
  
  • Statements = Properties × Resources × {Resources ∪ Literals}

  – Type System
  
  • rdf:type ∈ Properties
  
  • {rdf:type, sub, obj} ∈ Statements ⇒ obj ∈ Resources
A Formal Model of Reification in RDF

Reification

- \( \text{rdf:Statement} \in \text{Resource-Properties} \)
- \( \{ \text{rdf:predicate}, \text{rdf:subject}, \text{rdf:object} \} \subset \text{Properties} \)
- Reification of a triple \( \{ \text{pred}, \text{sub}, \text{obj} \} \) of statements is an element \( r \) of resources, which represent the reified triple and the elements \( s_1, s_2, s_3, \) and \( s_4 \) of statements, so that:

  - \( s_1: \{ \text{rdf:predicate}, r, \text{pred} \} \)
  - \( s_2: \{ \text{rdf:subject}, r, \text{subj} \} \)
  - \( s_3: \{ \text{rdf:object}, r, \text{obj} \} \)
  - \( s_4: \{ \text{rdf:type}, r, [\text{rdf:Statement}] \} \)

Collections

- \( \{ \text{rdf:Seq}, \text{rdf:Bag}, \text{and rdf:Alt} \} \subset \text{Resources-Properties} \)
- There is a subset of properties representing ordinal numbers (1, 2, 3, ...) which is called Ord. Elements of Ord are referred to by: \( \text{rdf:}_1, \text{rdf:}_2, \text{rdf:}_3, .. \)
RDF/XML as an XML-based Syntax for RDF

RDF also provides an XML-based syntax (called RDF/XML)

<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:contact="http://www.w3.org/2000/10/swap/pim/contact#">
    <contact:Person rdf:about="http://www.w3.org/People/EM/contact#me">
        <contact:fullName>Eric Miller</contact:fullName>
        <contact:mailbox rdf:resource="mailto:em@w3.org"/>
        <contact:personalTitle>Dr.</contact:personalTitle>
    </contact:Person>
</rdf:RDF>
Unlike hypertext, RDF URIs can refer to any identifiable thing, including things that may not be directly retrievable on the Web (such as the person Eric Miller).
RDF Syntax

- The data model does not enforce a specific syntax
- There are various syntax variants based on XML
- General form:

```xml
<rdf:rdf>
  <rdf:Description about="http://www.w3.org/Home/Lassila">
    <s:Creator>Ora Lassila</s:Creator>
    <s:createdWith rdf:resource="http://www.w3c.org/amaya"/>
  </rdf:Description>
</rdf:rdf>
```
Resulting RDF Graph

```
<rdf:rdf>
  <rdf:Description about="http://www.w3.org/Home/Lassila">
    <s:Creator>Ora Lassila</s:Creator>
    <s:createdWith rdf:resource="http://www.w3c.org/amaya"/>
  </rdf:Description>
</rdf:rdf>
```
RDF schema (RDFS) and Extensibility

• RDF solely defines the data-model.

• There is a need for the definition of vocabularies for such a data-model.

• RDF schemata are web resources (and have URIs). They can be described with the aid of RDF. Define an ontology of a language with RDF schemata

• Use the new vocabulary to describe instances of data

• Advantage: All languages use the same data model (simplifying interoperability)
An example RDF Schema

Annotation of WWW resources and *semantic* links

- **Writer**
- **hasWritten**
- **Book**

**FamousWriter**

**/twain/mark**

**../ISBN00010475**

**Schema(RDFS, Data(RDF))**

**T-Box**

**A-Box**
The most important modeling primitives

- Core classes
  - Root-Class rdfs:Resource
  - Meta-Class rdfs:Class
  - Literals rdfs:Literal
- rdfs:subclassOf-property
- taken over from rdf: properties (slots)
- rdfs:domain & rdfs:range
- rdfs:label, rdfs:comment, etc.
- taken over from rdf: InstanceOf (rdf:type)
RDF-schema: An example

s = rdfs:subClassOf

xyz:MotorVehicle

xyz:Truck

xyz:Van

xyz:PassengerVehicle

xyz:MiniVan

rdfs:Resource

rdfs:Class

t = rdf:type
Example: Defining RDF-schema using RDF-schema

- Namespace-URL:
  http://www.w3.org/2000/01/rdf-schema#

```xml
<rdfs:Class rdf:ID="Resource">
  <rdfs:label xml:lang="en">Resource</rdfs:label>
  <rdfs:label xml:lang="fr">Ressource</rdfs:label>
  <rdfs:comment>The most general class</rdfs:comment>
</rdfs:Class>

<rdfs:Class rdf:ID="Class">
  <rdfs:label xml:lang="en">Class</rdfs:label>
  <rdfs:label xml:lang="fr">Classe</rdfs:label>
  <rdfs:comment>The concept of Class</rdfs:comment>
  <rdfs:subClassOf rdf:resource="#Resource"/>
</rdfs:Class>

<rdf:Property ID="subClassOf">
  <rdfs:label xml:lang="en">subClassOf</rdfs:label>
  <rdfs:label xml:lang="fr">sousClasseDe</rdfs:label>
  <rdfs:comment>Indicates membership of a class</rdfs:comment>
  <rdfs:range rdf:resource="#Class"/>
  <rdfs:domain rdf:resource="#Class"/>
</rdf:Property>
```
The principle of ontology languages based on RDF

Legend

- Definition uses RDF data model
- Defined with
- Extension of

```
Instances
```

```
Ontology language
```

```
RDF Schema
```

```
RDF
```

RDF/RDFS “Liberality”

• No distinction between classes and instances (individuals)
  <Species,type,Class>
  <Lion,type,Species>
  <Leo,type,Lion>

• No distinction between language constructors and ontology vocabulary, so constructors can be applied to themselves/each other
  <type,range,Class>
  <Property,type,Class>
  <type,subPropertyOf,subClassOf>
Problems with RDFS as an Ontology Language

• RDFS too weak to describe resources in sufficient detail
  – No localised range and domain constraints
    • Can’t say that the range of hasChild is person when applied to persons and elephant when applied to elephants
  – No existence/cardinality constraints
    • Can’t say that all instances of person have a mother that is also a person, or that persons have exactly 2 parents
  – No transitive, inverse or symmetrical properties
    • Can’t say that isPartOf is a transitive property, that hasPart is the inverse of isPartOf or that touches is symmetrical

• Difficult to provide reasoning support
  – No “native” reasoners for RDF semantics
  – May be possible to reason via First Order Predicate Logic axiomatization – but does not scale