



Semantic Web

Hong-Gee Kim
Dankook University
June, 2003

Agenda

- Introduction
- Representations in the Semantic Web
- Languages
- Tools
- Applications
- Conclusions

Agenda

□ Introduction

- ⇒ Why do we need the Semantic Web?
- ⇒ What is the Semantic Web?
- ⇒ How the Semantic Web Will Be Possible?

□ Representations in the Semantic Web

□ Languages

□ Tools

□ Applications

□ Conclusions

Searching for Information

Google™ 고급 검색 환경설정 언어 도구 검색 도움말

sars 구글 검색

☉ 전체 검색 ☉ 한글 페이지 검색

웹 문서 이미지 뉴스그룹 디렉토리

sars에 대한 웹 검색 전체 370,000 중 결과 1 - 10 검색 소요 시간: 0.15 초

카테고리: [Regional](#) > [Africa](#) > [South Africa](#) > [Government](#)

[SARS \(South African Revenue Service\)](#)
[www.sars.gov.za/](#) - 3k - [저장된 페이지](#) - [비슷한 페이지](#)

[CDC | Severe Acute Respiratory Syndrome \(SARS\)](#)
[www.cdc.gov/ncidod/sars/](#) - 28k - [저장된 페이지](#) - [비슷한 페이지](#)

[Tomato Nation](#)
[www.earlygirl.com/](#) - 10k - [저장된 페이지](#) - [비슷한 페이지](#)

[Severe Acute Respiratory Syndrome \(SARS\)](#)
[www.who.int/csr/sars/en/](#) - 29k - [저장된 페이지](#) - [비슷한 페이지](#)

[The Safety and Reliability Society](#)
[www.sars.u-net.com/](#) - 1k - [저장된 페이지](#) - [비슷한 페이지](#)

[SARS](#)
[www.sarscapital.com/](#) - 8k - [저장된 페이지](#) - [비슷한 페이지](#)

[SARS Website](#)
[home.swipnet.se/amc/](#) - 2k - [저장된 페이지](#) - [비슷한 페이지](#)

[Sars Bergen](#)
[www.uib.no/fa/sars/](#) - 1k - [저장된 페이지](#) - [비슷한 페이지](#)

[Utah State Office of Education \(USOE\)/Services for At Risk ...](#)
[www.usoe.k12.ut.us/sars/](#) - 13k - [저장된 페이지](#) - [비슷한 페이지](#)

[Constance Sars Relatie-bemiddeling](#)
[www.constancesars.nl/](#) - 2k - [저장된 페이지](#) - [비슷한 페이지](#)

결과 페이지: 1 2 3 4 5 6 7 8 9 10 다음

Goooooooooogle ▶

Reality of Information Age

- We produce 1 ExaByte of Information per year
 - ⇒ **Source: Stanford University Study**

- 60% of programming time is wasted on extracting and moving data from point A to point B
 - ⇒ **Source: Gartner Group Study**

- 50% of world economy depends on Office Work manipulating data by hand
 - ⇒ **Source: MIT research**

The Need for the Semantic Web

□ Knowledge Management

- ⇒ **Searching information**
- ⇒ **Extracting information**
- ⇒ **Maintenance**
- ⇒ **Automatic document generation**

□ Web Commerce

- ⇒ **Limitation of traditional shopbots**
- ⇒ **Meta-online stores using standard representation formalisms**
- ⇒ **Semantic mappings translate different formats representing products**

□ E-Business

- ⇒ **Rich modeling primitives need to be defined for defining, mapping, and exchanging product data**
- ⇒ **Standard conceptualization of various business areas**
- ⇒ **Efficient translation services are needed for the communication between business partners**

The Semantic Web

- “The Semantic Web is not a separate Web but an extension of the current one, in which information is given well-defined meaning, better enabling computers and people to work in cooperation”



Source: Scientific American: The Semantic Web
TIM BERNERS-LEE, JAMES HENDLER and ORA LASSILA
<http://www.sciam.com/2001/0501issue/0501berners-lee.html>

The Semantic Web

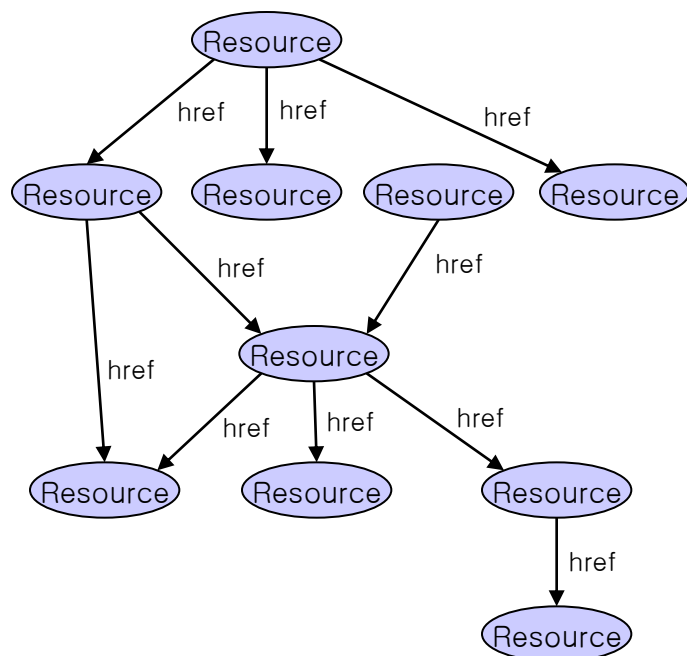
- “The Semantic Web will enable machines to COMPREHEND semantic documents and data, not human speech and writings.”



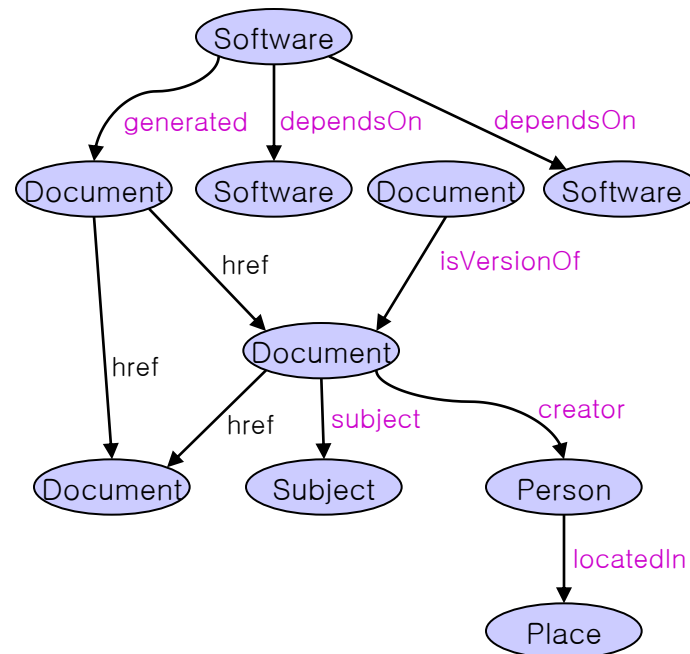
Source: Scientific American: The Semantic Web
TIM BERNERS-LEE, JAMES HENDLER and ORA LASSILA
<http://www.sciam.com/2001/0501issue/0501berners-lee.html>

The Evolving Web

- “The Semantic Web is a web of data, in some ways like a *global database*.” by Tim Berners-Lee



- Very little information available



- More information available



How the Semantic Web Will Be Possible

□ Languages

- ⇒ **Formal Syntax and Formal Semantics**
- ⇒ **Real world semantics → “Ontologies”**

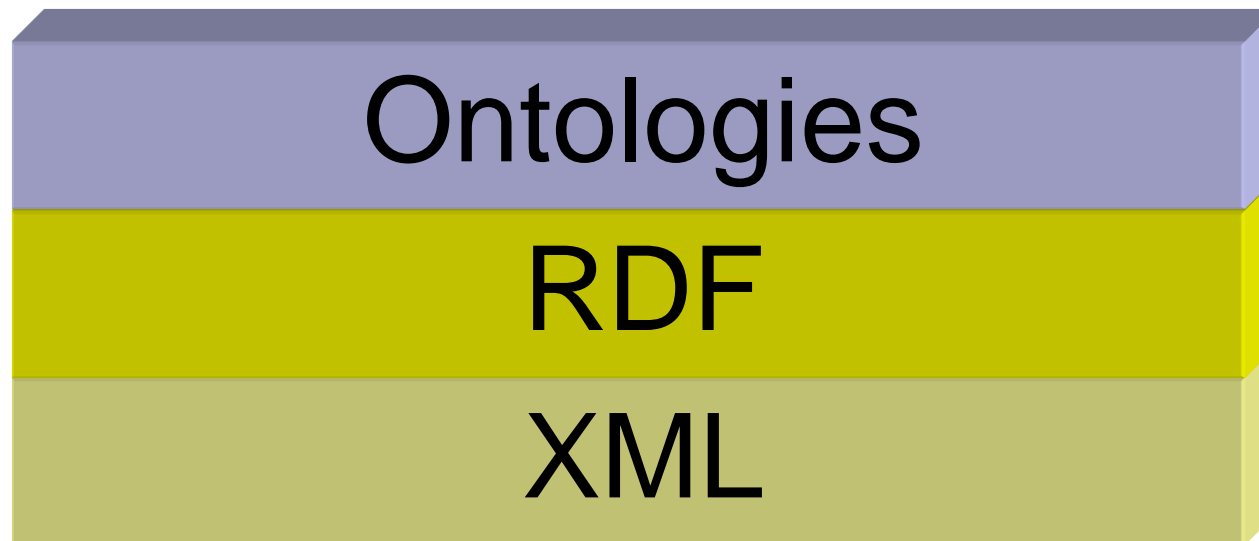
□ Tools

- ⇒ **Ontology builders and browsers**
- ⇒ **Ontology integration tools**
- ⇒ **Semantic annotators**
- ⇒ **Reasoners**

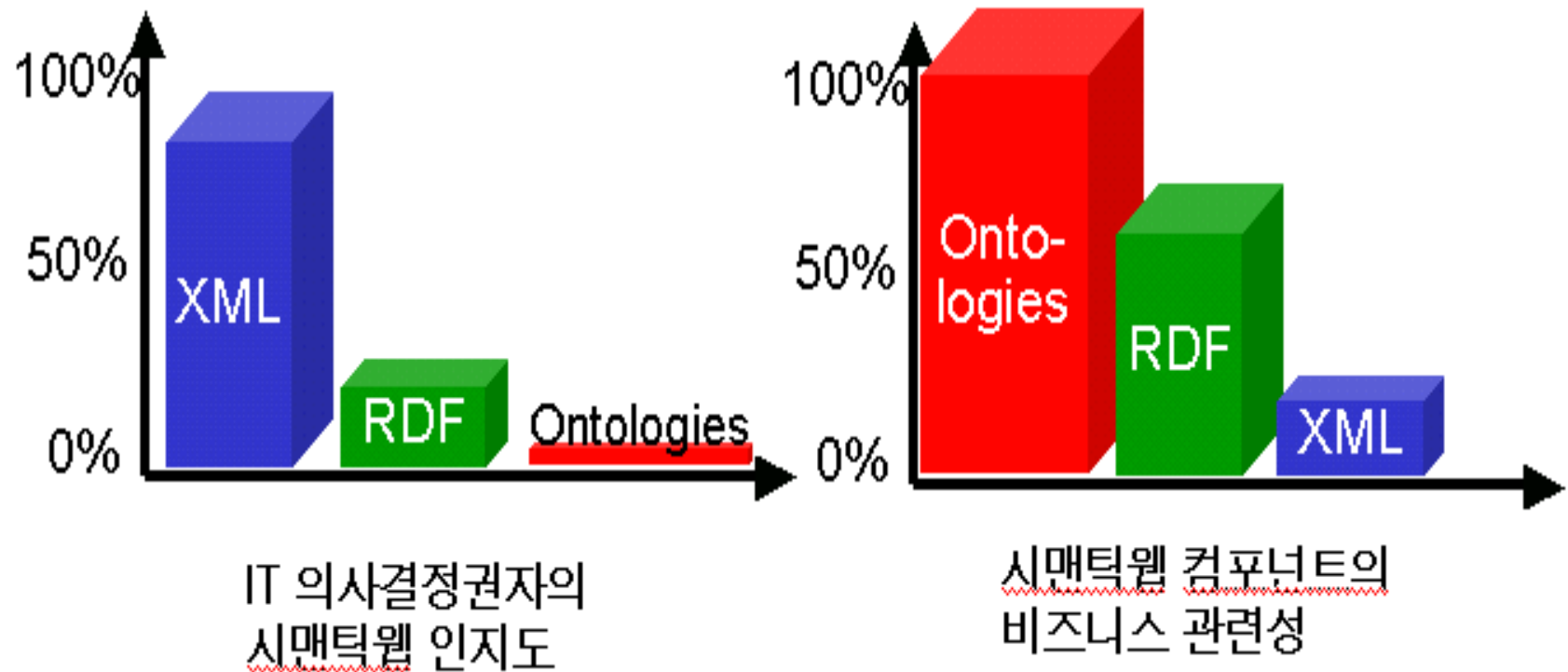
□ Applications

- ⇒ **Knowledge management systems**
- ⇒ **Natural language search engines**
- ⇒ **E-Commerce**

The 3 cornerstones of the Semantic Web

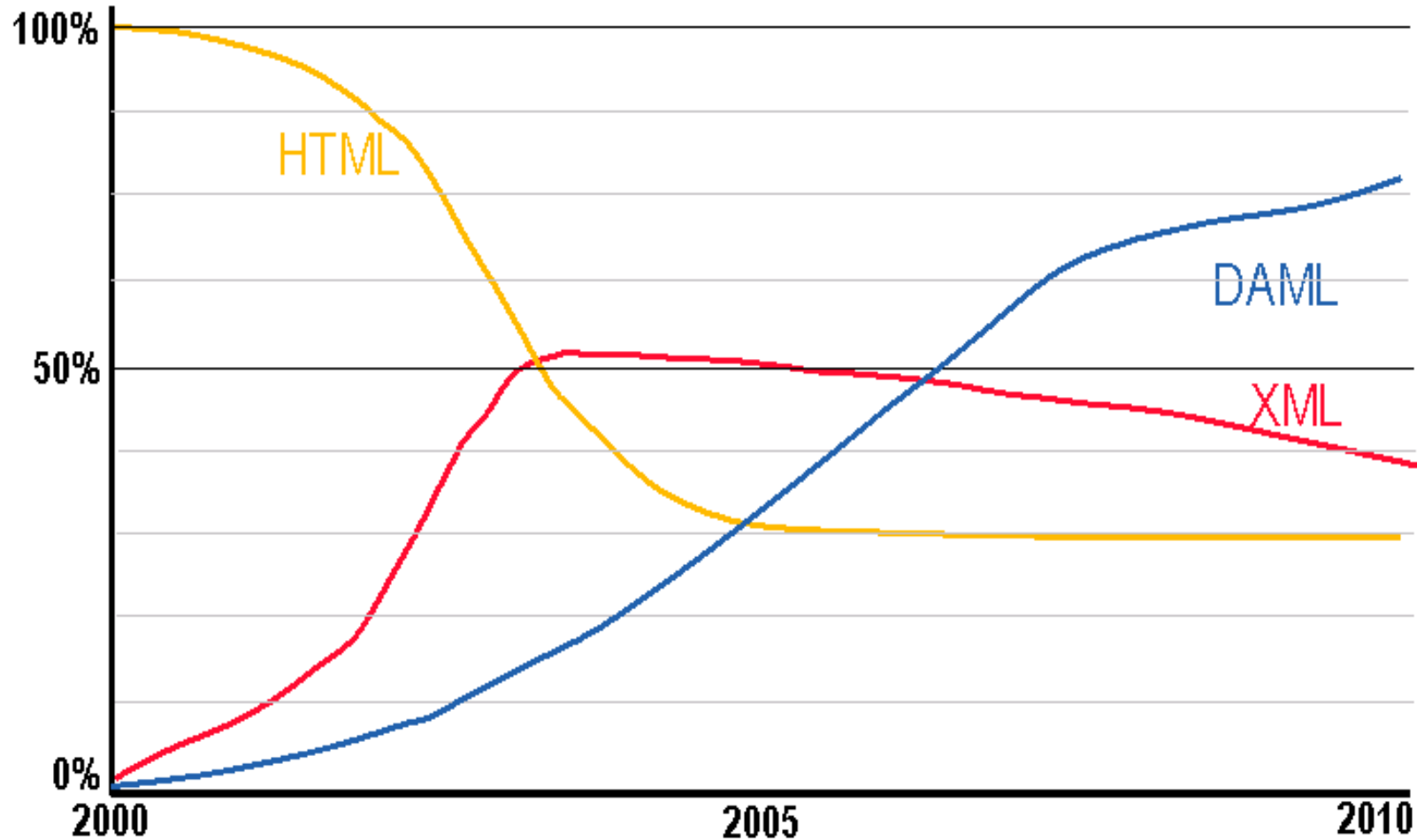


Semantic Web Awareness and Its Business Relevance



Chris Horak, 2002

Future Trends of Markup Languages



Source: DARPA 2001

Approaches for the Semantic Web

□ Centralized

- ⇒ **Easier to manage on a small scale (one company)**
- ⇒ **Hard to get agreements between corporations**
- ⇒ **Cannot scale to the size of a global knowledge base**

□ Decentralized

- ⇒ **Distributed, but fully connected**
- ⇒ **Common denominator approach**
- ⇒ **Some fuzziness allowed to achieve versatility**

} Semantic
Web

Agenda

- Introduction

- **Representations in the Semantic Web**

 - ⇒ **Meaning**

 - ⇒ **Knowledge**

 - ⇒ **Ontology**

- Languages

- Tools

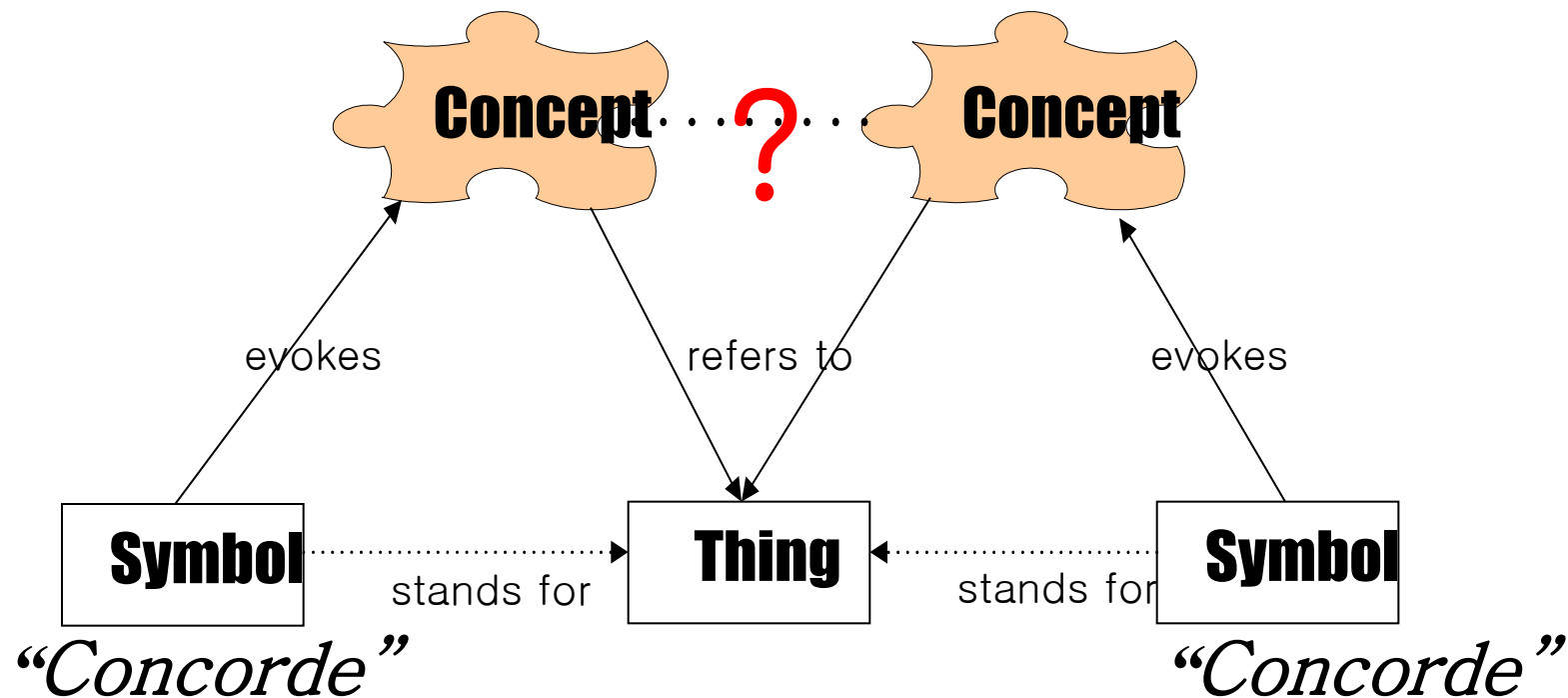
- Applications

- Conclusions

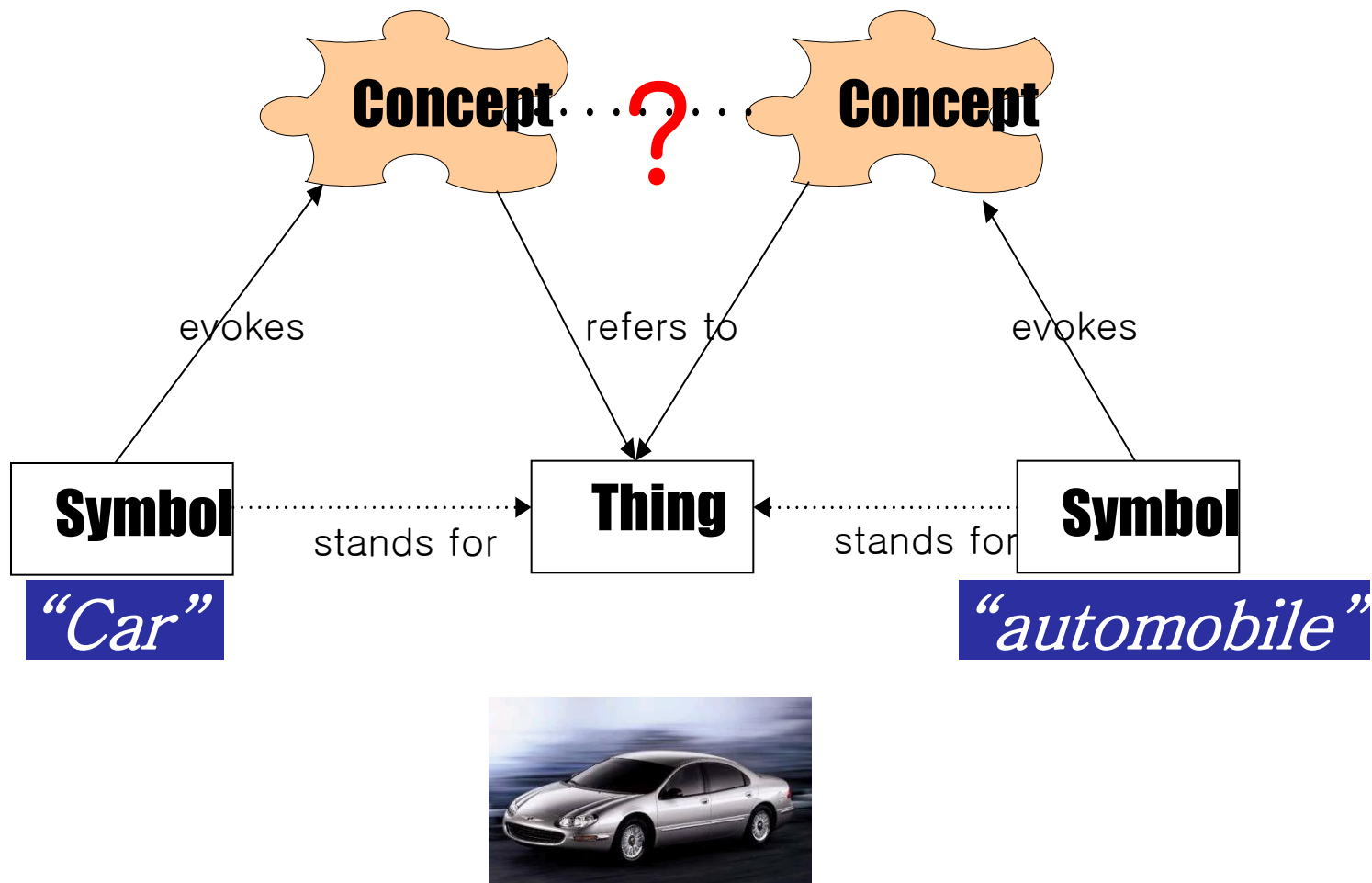
Definition of Re-presentation

- ❑ A relationship between two domains where the first is meant to “*stand for*” the second.
- ❑ The first domain, the representer, is more concrete, immediate, or accessible in some way than the second.
- ❑ The most popular type of representer is a formal symbol so that it can be machine-processable.

Meaning and Human Communication

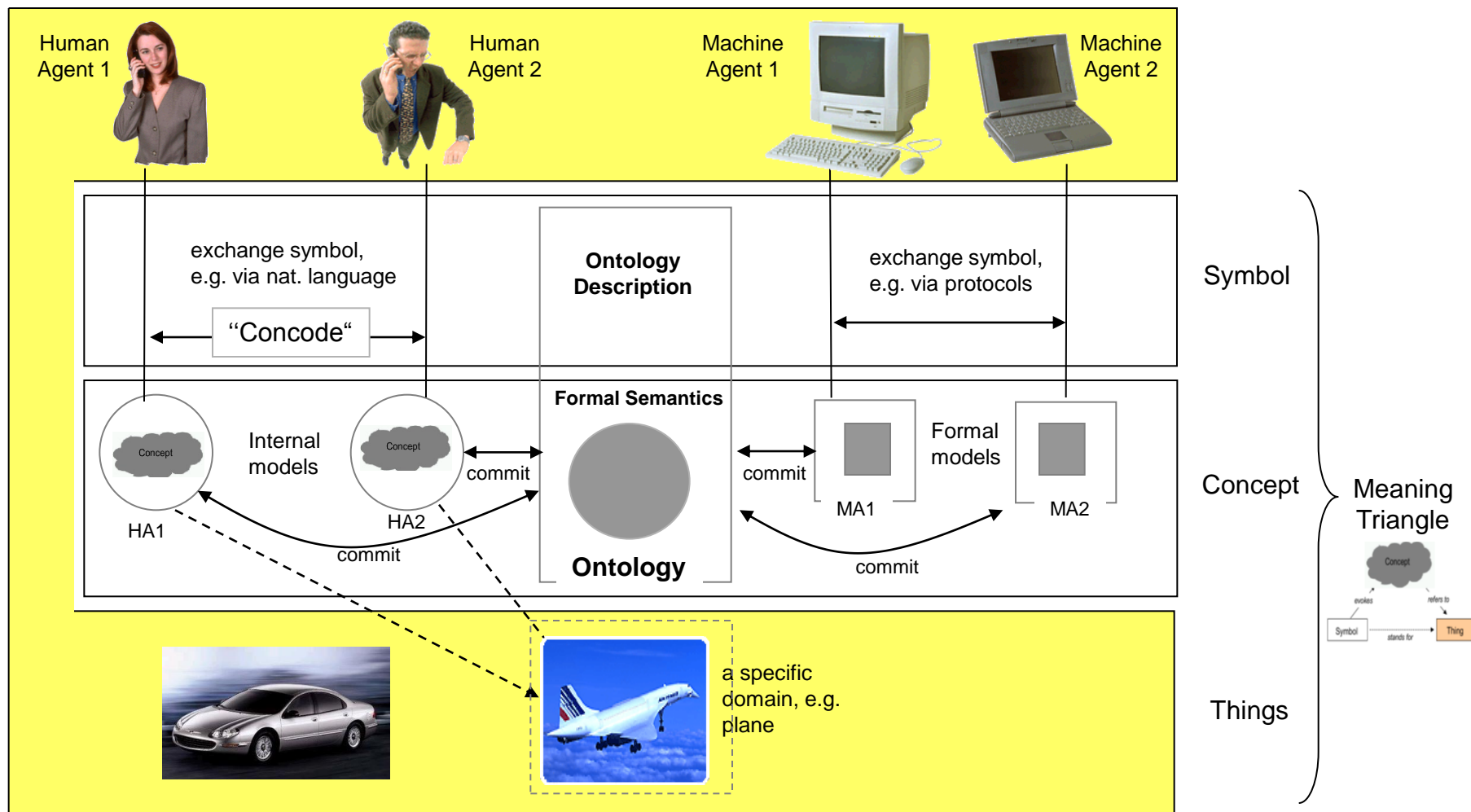


Meaning and Human Communication



Human and Machine communication

[Maedche et al., 2002]



Knowledge Representation

□ Proposition

- ⇒ “**Hong-Gee knows that**  **....**”
- ⇒ **An abstract entity that can be *true* or *false***

□ Belief (or Knowledge)

- ⇒ **A collection of propositions held by an agent to be true**
- ⇒ **Represents different possible ways the world could be**

□ Knowledge Representation

- ⇒ **The field of study within AI**
- ⇒ **Concerned with using formal symbols to represent a collection of propositions believed by some agents**

□ Reasoning

- ⇒ **The formal manipulation of the symbols representing a collection of believed propositions to produce representations of new ones**

Two Approaches to Knowledge Representation

□ Logic-based representations

- ⇒ **Motivated from First-Order Predicate Calculus**
- ⇒ **Reasoning amounts to verifying logical consequence**
- ⇒ **Rigorous mathematical notions that unambiguously capture facts about the world.**

□ Non-logic based representations

- ⇒ **Motivated from cognitive intuitions**
- ⇒ ***Ad hoc* data structure, and *ad hoc* reasoning procedures that manipulate the structures**
- ⇒ **Practical tools to describe the world – easy representation and efficient reasoning**
- ⇒ **But lack of precise semantic characterization**

Ontology Representation

- ❑ A general logical theory constituted by a vocabulary
- ❑ A set of statements about a domain of interest in some logic language
- ❑ The specification of conceptualizations, used to help programs and humans share knowledge
- ❑ An agreed-upon vocabulary for exchanging information
- ❑ The working model of entities and interactions in some particular domain of knowledge or practices, such as electronic commerce

Ontology versus Knowledge Base

□ Ontology

- ⇒ **A general logical theory**
 - ☑ The general conceptual structures of a domain of interest
- ⇒ **A set of *intensional* logical statements**
- ⇒ **Mostly developed during the setting up of an ontology-based systems**

□ Knowledge base

- ⇒ **A theory of particular circumstances**
 - ☑ The specification of a given state of affairs
- ⇒ **A set of *extensional* statements**
- ⇒ **The facts in a knowledge base may be constantly changing**

Ontology as Knowledge Lego

hand

extremity

body

chronic

acute

abnormal

normal



gene

protein

cell

expression

Lung

inflammation

infection

bacterial

ischaemic

deletion

polymorphism

Knowledge Lego: *Reusable Concepts*

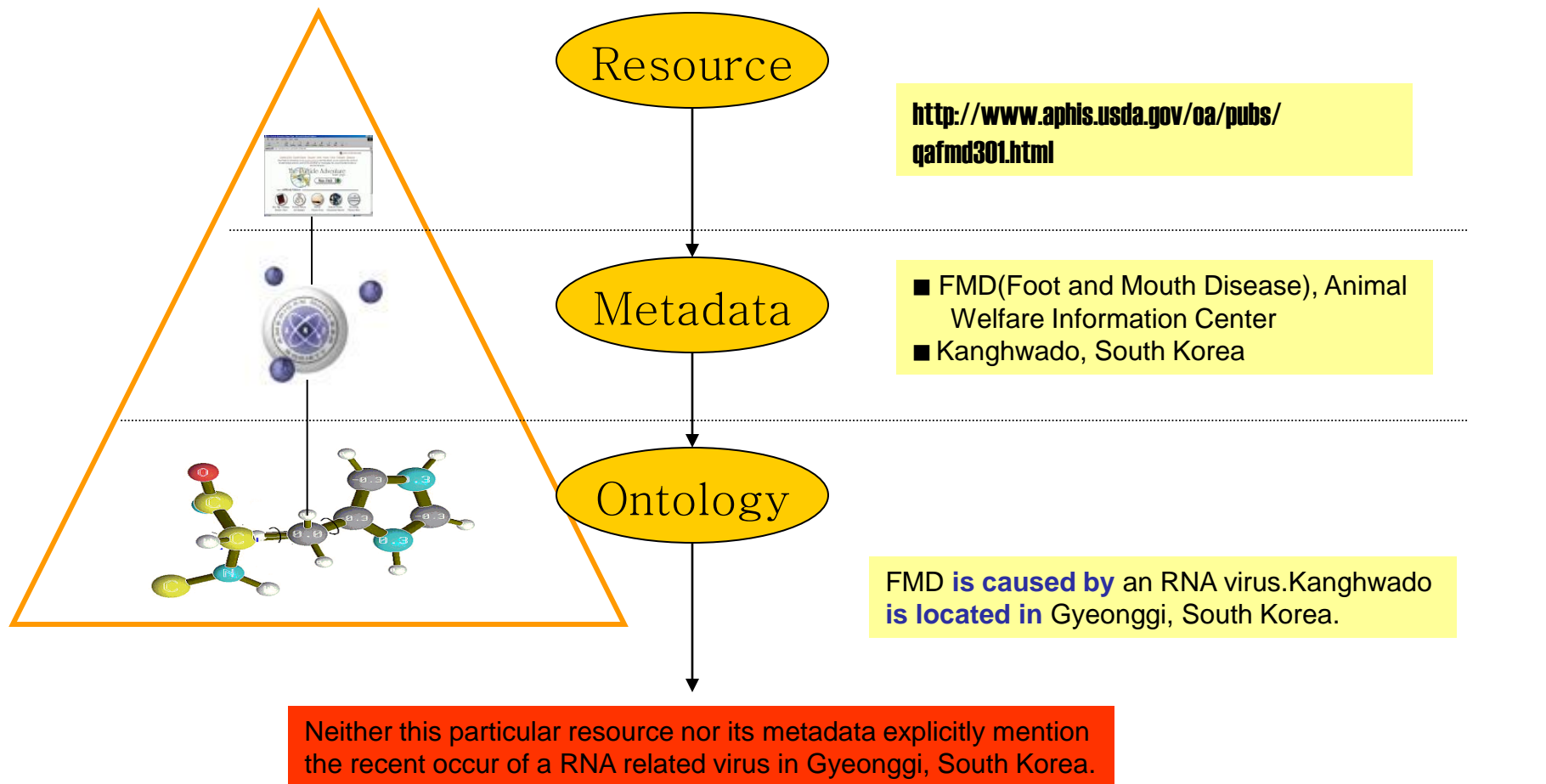
[Rector, 2002]

“SNPolymorphism of CFTRGene causing Defect in MembraneTransport of ChlorideIon causing Increase in Viscosity of Mucus in CysticFibrosis...”



“Hand which is anatomically normal”

Information Retrieval Using Ontologies



Only an assisted search that maps metadata to underlying ontologies could retrieve this resource in response to the query “**the recent occur of a RNA related virus in Gyeonggi, South Korea.**”

Ontological Representation

A matter of rigour and representational expressivity

□ Lightweight

The basic building blocks are

- ⇒ Concepts, atomic types
- ⇒ Relationships between concepts
- ⇒ Is-a hierarchy

□ Heavyweight

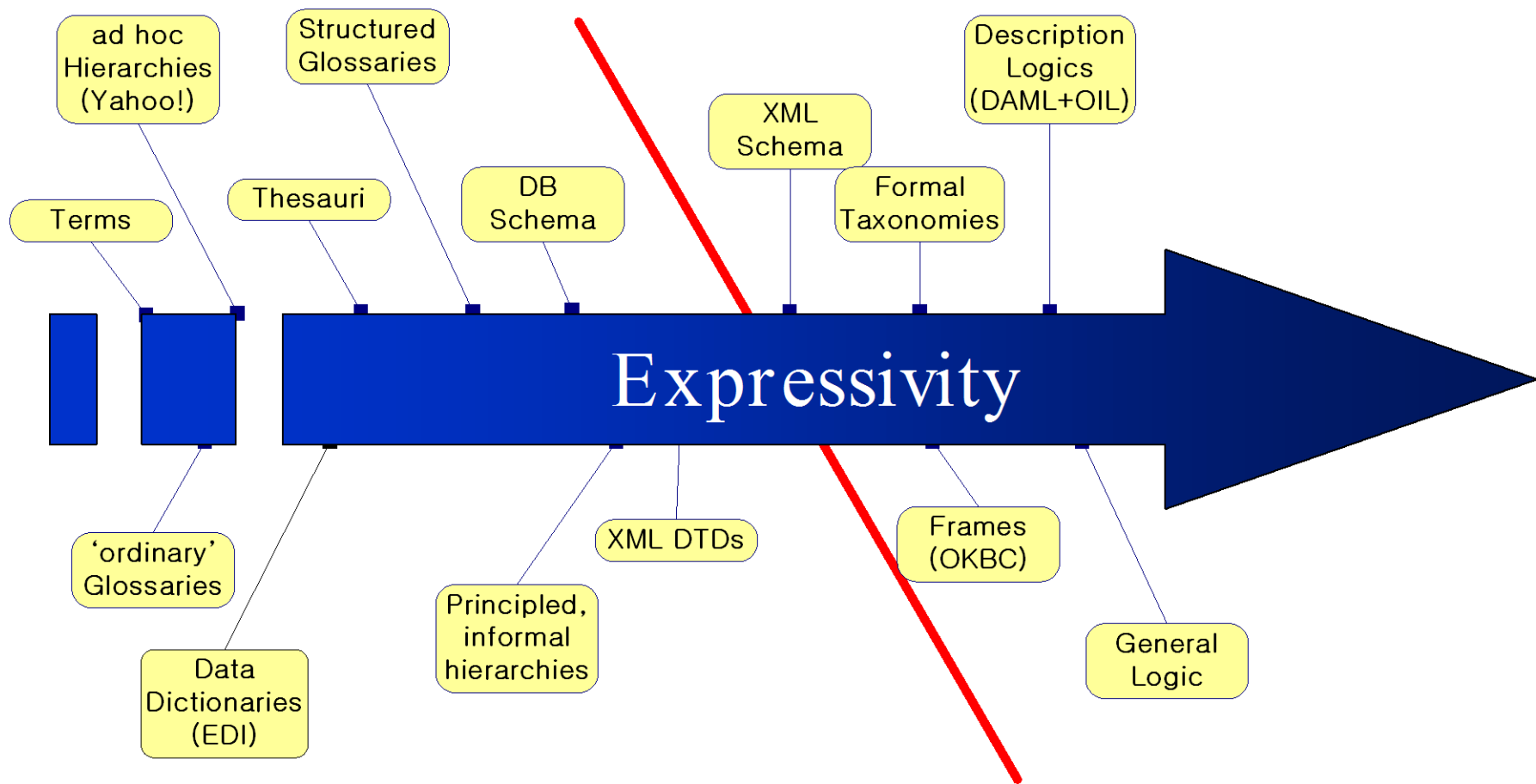
A small set of constructs for complex structures

- ⇒ Metaclasses
- ⇒ Type constraints on relations
- ⇒ Cardinality constraints
- ⇒ Taxonomy of relations
- ⇒ Reified statements

Implicit knowledge can be inferred automatically

- ⇒ Axioms
- ⇒ Semantic entailments
- ⇒ Inference systems

Kinds of Ontologies



Agenda

- Introduction
- Representations in the Semantic Web
- **Languages**
 - ⇒ **Language requirements**
 - ⇒ **Layer language model for WWW**
 - ⇒ **XML**
 - ⇒ **RDF & RDFS**
 - ⇒ **DAML+OIL**
 - ⇒ **OWL**
 - ⇒ **Topic MAPS**
- Tools
- Applications
- Conclusion

Languages for the Semantic Web

- ❑ Should provide formal syntax and formal semantics to enable automated processing of the contents
- ❑ Should provide standardized vocabulary referring to real-world semantics so that machine and human agents can share information and knowledge → Ontology

Ontology Language Requirement

□ Universal Expressivity

- ⇒ **The data format should have enough expressive power to express any form of data.**

□ Syntactic Interoperability

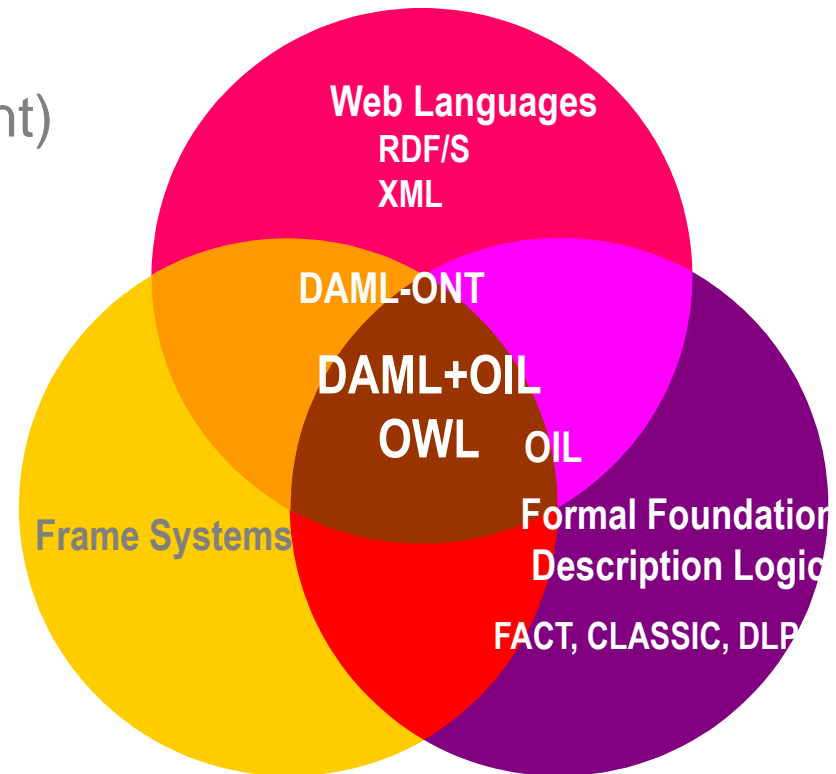
- ⇒ **The data should be easily readable by applications (or parsers).**
- ⇒ **The representation of the data should be easily exploited by (queries).**

□ Semantic Interoperability

- ⇒ **The machine can understand the meaning of the data.**
- ⇒ **Unknown terms can be defined by known terms.**

Ontology Languages

- 1st generation web language
: ***for Data Display*** (HTML)
- 2nd generation web language (current)
: ***for Data Description***
(XML, XML Schema)
- Next generation web language
: ***for Data Definition***
(RDFS, DAML+OIL, OWL)



Deborah McGuinness, NSF/NCAR October 30, 2002

History of Ontology Languages

□ RDF(S)

⇒ **Developed by W3C (1999, 02)**

□ Ontology Inference Layer : OIL

⇒ **Developed by group of European researchers (2000, 01)**

□ DAML Ontology Language : DAML-ONT

⇒ **Developed by US researchers working in DAML program (2000, 10)**

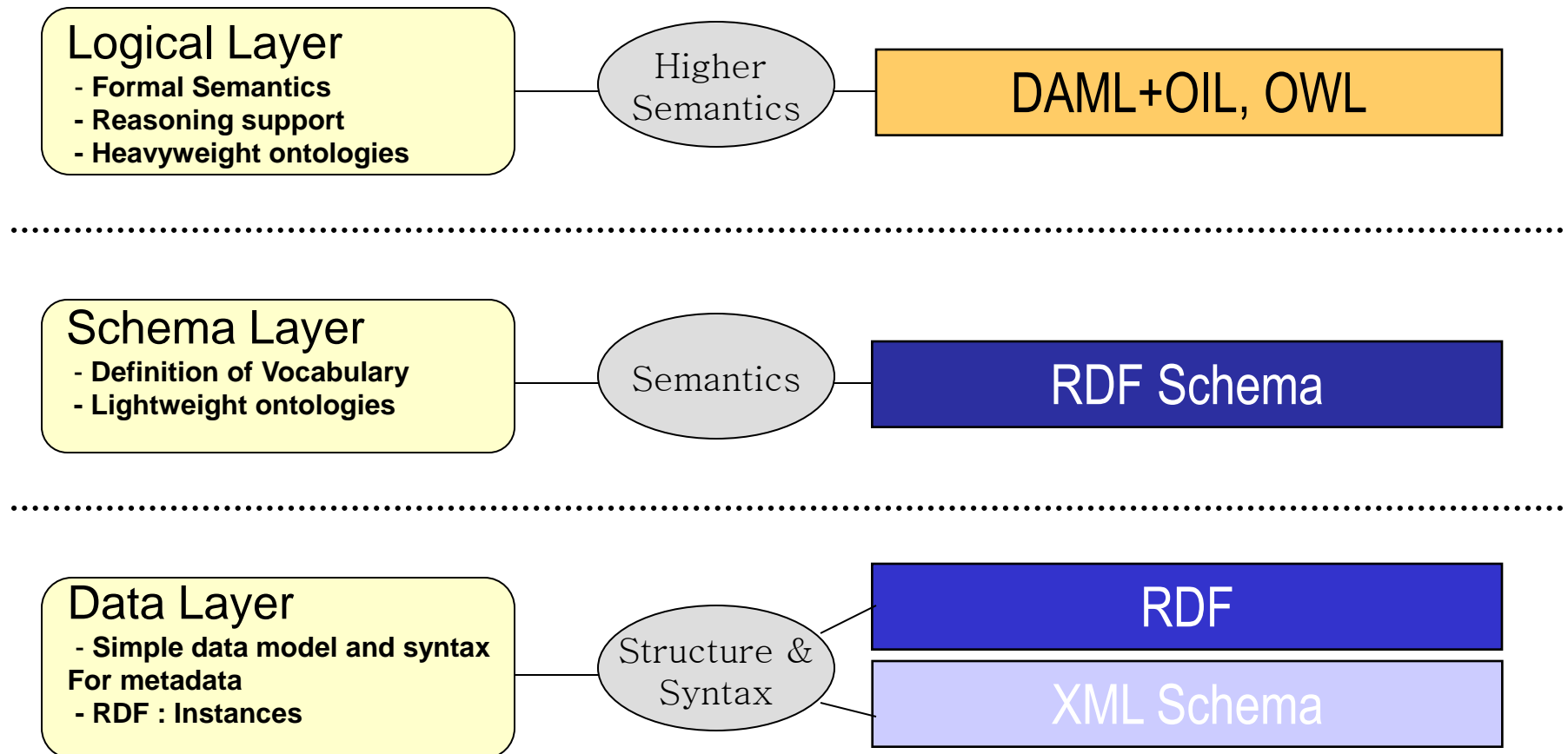
□ Effort merged in DAML+ OIL

⇒ **Developed by EU/US joint committee (2000, 12)**

□ W3C Web Ontology group : OWL

⇒ **W3C standard based on DAML+OIL (2002, 02)**

Three Layered Architecture of the Semantic Web



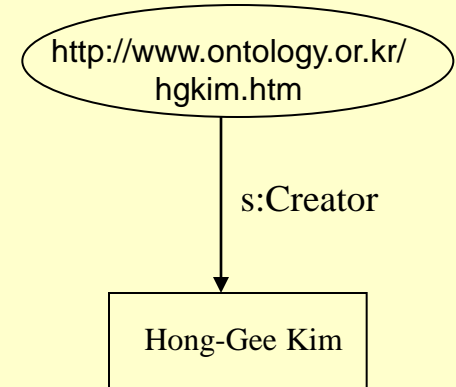
XML is not enough for SI

- ❑ Do not impose a common interpretation of the data
- ❑ Only feasible for closed collaboration
- ❑ The intended meaning of different elements is implicit

```
<class-def>  
  <name>carnivore</name>  
  <slot-constraint>  
    <name>is-kind-of</name>  
    <has-value>animal</has-value>  
  </slot-constraint>  
</class-def>
```

RDF is not enough

- ❑ **RDF** consists of two parts
 1. **RDF Model** (a set of triples)
 2. **RDF Syntax** (different **XML serialization syntaxes**)
 - ⇒ **RDF a small set of modelling primitives + syntax**
 - ⇒ **RDF *does not* commit to a domain vocabulary**
- ❑ **RDF Schema** for definition of Vocabularies (simple Ontologies) for RDF
- ❑ RDF/RDFS is only a very weak semantic interpretation
- ❑ RDF/RDFS is not an inference model
- ❑ Cannot express (limited semantics)
 - Class-def defined herbivore**
 - subclass-of animal, NOT carnivore**



Triples

Resource (subject)
http://www.ontology.or.kr/hgkim.htm

Property (predicate)
http://www.schema.org/#Creator

Value (object) "Hong-Gee Kim"

RDF(S) Example

RDF Document

```
<?xml version='1.0' encoding='ISO-8859-1'?>
<!DOCTYPE rdf:RDF [
  <!ENTITY rdf 'http://www.w3.org/1999/02/22-rdf-syntax-ns#'>
  <!ENTITY mv 'http://protege.stanford.edu/mv#'>
]>
<rdf:RDF xmlns:rdf="&rdf;"
  xmlns:mv="&mv;"
  <mv:Truck rdf:about="&mv;test3_03">
    <mv:registeredTo rdf:resource="&mv;test3_04"/>
  </mv:Truck>
  <mv:Person rdf:about="&mv;test3_04"
    mv:name="Ora Lassila"/>
</rdf:RDF>
```

RDFS

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

```
<rdf:Class rdf:about="&mv;Truck">
  <rdf:subClassOf rdf:resource="&mv;MotorVehicle"/>
</rdf:Class>
<rdf:Property rdf:about="&mv;registeredTo"
  rdfs:maxCardinality="1">
  <rdf:domain rdf:resource="&mv;MotorVehicle"/>
  <rdf:range rdf:resource="&mv;Person"/>
</rdf:Property>
```

Limit of RDFS

❑ Severely lacking in expressive power

⇒ **Domain and range constraints rather than Value-Type**

☑ E.g., can't define class of people all of whose children are male

⇒ **No cardinality constraints**

☑ Particularly important for “exactly 1” and “at most 1”

⇒ **No decompositions**

☑ Particularly important for “disjoint” and “exhaustive”

⇒ **No axioms, No negation**

❑ Not useful for checking consistency

E.g., can't prove an object is not an instance of a class

❑ More powerful ontology representation languages are needed.

DAML+ OIL as an extension of RDF(S)

DAML+OIL

RDF(S)

class-def
subclass-of
slot-def
subslot-of
domain
range

class-expressions

: AND, OR, NOT

slot-constraints

: has-value, value-type
cardinality etc

slot-properties

: transitive, symmetric etc

data-type

: string, integer

Example of DAML+OIL

Class-def

defined White-van-man
subclass-of Man
slot-constraint drives
has-values White-van

covered White-van-man by
Aggressive-driver

Class-def

primitive White-van-man

equivalent

White-van-man
(Man and
slot-constraint drives has-
values White-van))

covered White-van-man by
Aggressive-driver

```
<rdfs:Class rdf:ID="White-van-man">  
  <rdfs:subClassOf>  
    <rdfs:Class  
      rdf:about="Aggressive-driver"/>  
  </rdfs:subClassOf>  
</rdfs:Class>  
<rdfs:Class rdf:about="White-wan-man">  
  <daml:sameClassAs>  
    <rdfs:Class>  
      <daml:intersectionOf>  
        <rdfs:Class rdf:about="man"/>  
        <daml:Restriction>  
          <daml:onProperty  
            rdf:resource="drives"/>  
          <daml:hasClass  
            rdf:resources="White-van"/>  
        </daml:Restriction>  
      </daml:intersectionOf>  
    </rdfs:Class></...></rdfs:Class>
```


□ Syntax = Extension of RDFS

- ⇒ **constraints on properties: exists, forall, cardinality**
- ⇒ **equivalence, disjointness, covering**
- ⇒ **necessary and sufficient condition**

□ Semantics

- ⇒ **The model theory is based on Description Logic**

□ Instance Date

- ⇒ **RDF is used for class/property membership assertions.**

□ Data types

- ⇒ **Using the full range of XML Schema data types**

□ Machine Understanding and Automated Reasoning

DAML+ OIL language structure

- Header
- Class elements
- Property elements
- Instances
- zero or more headers followed by zero or more class elements, property elements, instances.

Header

- daml:Ontology contains versionInfo, comment, imports elements

```
<Ontology rdf:about="">  
  <versionInfo>  
    $Id: NOTE-daml+ oil-reference-20011218.html,v 1.6 2001/12/18  
    22:12:09 connolly Exp $  
  </versionInfo>  
  <rdfs:comment>An example ontology</rdfs:comment>  
  <imports rdf:resource="http://www.w3.org/2001/10/daml+ oil"/>  
</Ontology>
```

Class Elements (1)

❑ Disjointness

⇒ **daml:disjointWith**, **daml:disjointUnionOf**

❑ Equality

⇒ **daml:sameClassAs**, **daml:equivalentTo**

❑ Boolean combinations of class expressions

⇒ **daml:intersectionOf**, **daml:unionOf**, **daml:complementOf**

❑ Enumeration elements

⇒ **daml:oneOf**

Class Elements (2)

□ Property Restrictions

⇒ **Cardinality restrictions**

- ☑ **daml:cardinality, daml:maxCardinality, daml:minCardinality**
- ☑ **daml:cardinalityQ, daml:maxCardinalityQ, daml:minCardinalityQ**

⇒ **Value restrictions**

- ☑ **daml:toClasss, daml:hasValue, daml:hasClass**

Property Elements

- ❑ `daml:samePropertyAs`
- ❑ `daml:equivalentTo`
- ❑ `daml:inverseOf`
- ❑ `daml:transitiveProperty`
- ❑ `daml:uniqueProperty`
- ❑ `daml:unambiguousProperty`

Instances

■ Examples of Instances

```
<continent rdf:ID="Asia"/>
```

```
<rdf:Description rdf:ID="Asia">  
  <rdf:type>  
    <rdfs:Class rdf:about="#continent"/>  
  </rdf:type>  
</rdf:Description>
```

```
<rdf:Description rdf:ID="India">  
  <is_part_of rdf:resource="#Asia"/>  
</rdf:Description>
```

DAML+ OIL is not enough

- semantics is too weak
- malformed restrictions
- unsuitable named element

□ OWL (Web Ontology Language)

- ⇒ **Web ontology language developed by W3C**
- ⇒ **Reversion of DAML+OIL**
- ⇒ **Based on Description Logic**
- ⇒ **Perform reasoning task**
- ⇒ **Adds more vocabularies for describing properties and classes**

Three sublanguages of OWL

□ OWL Lite

- ⇒ **Subset of DL**
- ⇒ **light-weight** 한 ontology 구축에 적합

□ OWL DL

- ⇒ **Support Description Logic segment**
- ⇒ **Has properties for reasoning systems**
- ⇒ **heavy-weight**한 ontology 구축에 적합

□ OWL Full

- ⇒ **Union of OWL and RDFS**
- ⇒ **Allow free mixing of OWL with RDF Schema**
- ⇒ **Not enforce a strict separation of classes, properties, individuals**

□ OWL Lite \subset DL \subset Full

OWL Lite Constructions (1)

□ RDF Schema Features:

- ⇒ **Class**
- ⇒ ***rdf:Property***
- ⇒ ***rdfs:subClassOf***
- ⇒ ***rdfs:subPropertyOf***
- ⇒ ***rdfs:domain***
- ⇒ ***rdfs:range***
- ⇒ **Individual**

□ Header Information

- ⇒ *imports*
- ⇒ *versionInfo*
- ⇒ *priorVersion*
- ⇒ *backwardCompatibleWith*
- ⇒ *incompatibleWith*

■ (In)Equality

- *equivalentClass*
- *equivalentProperty*
- *sameIndividualAs*
- *differentFrom*
- *allDifferent*

Examples of (In)Equality

■ owl:equivalentClass

```
<owl:Class rdf:about="#US_President">  
  <owl:equivalentClass  
    rdf:resource="#PrincipalResidentOfWhiteHouse"/>  
</owl:Class>
```

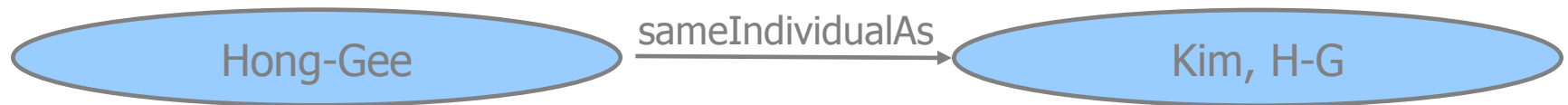
■ owl:equivalentProperty

: hasLeader, hasHead equivalentProperty,



■ owl:sameIndividualAs

- used to create different names that refer to the same individual



■ owl:differentFrom

- Frank is **differentFrom** Deborah

: 명시하지 않으면 reasoner는 반드시 다른 individual이라고 추론하지 않음

OWL Lite Constructions (2)

■ Property Type

Restrictions:

- *allValuesFrom*
- *someValuesFrom*

■ Class Intersection

- *intersectionOf*

■ Datatypes

■ Restricted Cardinality

- *minCardinality* (only 0 or 1)
- *maxCardinality* (only 0 or 1)
- *cardinality* (only 0 or 1)

■ Property Characteristics:

- *inverseOf*
- *transitiveProperty*
- *symmetricProperty*
- *functionalProperty*
- *InverseFunctionalProperty*

Examples of Property type restrictions (1)

■ owl:allValuesFrom

```
<owl:Restriction>  
  <owl:onProperty rdf:resource="#hasParent" />  
  <owl:allValuesFrom rdf:resource="#Human" />  
</owl:Restriction>
```



■ owl:someValuesFrom : 부모 중 적어도 한 명은 Physician 이다.

```
<owl:Restriction>  
  <owl:onProperty rdf:resource="#hasParent" />  
  <owl:someValuesFrom rdf:resource="#Physician"/>  
</owl:Restriction>
```

Examples of Property type restrictions (2)

- **owl:inverseOf**

if hasChild *inverseOf* hasParent **and** Deborah *hasParent* Louise
then Louise *hasChild* Deborah

- **owl:transitiveProperty**

if pair(X,Y), pair(Y,Z) are instances of transitive property P,
then pair(X,Z) is also instance of P

- **owl:symmetricProperty**

if pair(X,Y) is an instance of P, **then** pair(Y,X) is also instance of P

OWL DL, Full constructions

□ Class Axioms:

- ⇒ ***oneOf***
- ⇒ ***disjointWith***
- ⇒ ***equivalentClass***
(applied to class expressions)
- ⇒ ***rdfs:subClassOf***
(applied to class expressions)

□ Arbitrary Cardinality:

- ⇒ ***minCardinality***
- ⇒ ***maxCardinality***
- ⇒ ***cardinality***

■ Boolean Combinations of Class Expressions:

- *unionOf*
- *intersectionOf*
- *complementOf*

■ Filler Information:

- *hasValue*

Changes from DAML+ OIL to OWL (1)

- ⇒ **cyclic subclasses are allowed**
- ⇒ **multiple `rdfs:domain` and `rdfs:range` properties are handled as intersection**
- ⇒ **`rdf:parseType="daml:collection" → rdf:parseType="Collection"`**
- ⇒ **not support using datatypes as types**

- `<size>`
 `<xsd:integer rdf:value="10"/>`
 `</size>`

Instead use

- `<size rdf:datatype="&xsd;integer">10</size>`

- ⇒ **qualified restrictions removed**

- ☑ **`daml:cardinalityQ`**

- ☑ **`daml:hasClassQ`**

- `daml:maxCardinalityQ`

- `daml:minCardinalityQ`

- ⇒ **`owl:symmetricProperty` added**

Changes from DAML+ OIL to OWL (2)

□ properties and classes renamed

DAML+ OIL	OWL
daml:differentIndividualFrom	owl:differentFrom
daml:equivalentTo	owl:sameAs
daml:sameClassAs	owl:equivalentClass
daml:samePropertyAs	owl:equivalentProperty
daml:hasClass	owl:someValuesFrom
daml:sameClassAs	owl:equivalentClass
daml:samePropertyAs	owl:equivalentProperty
daml:toClass	owl:allValuesFrom
daml:UnambiguousProperty	owl:InverseFunctionalProperty
daml:UniqueProperty	owl:FunctionProperty

OWL Header

```
<rdf:RDF
  xmlns      = "http://www.example.org/wine#"
  xmlns:owl  = "http://www.w3.org/2002/07/owl#"
  xmlns:rdf  = "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs = "http://www.w3.org/2000/01/rdf-schema#"
  xmlns:xsd  = "http://www.w3.org/2000/10/XMLSchema#">

  <owl:Ontology rdf:about="http://www.example.org/wine.owl">
    <rdfs:comment>An example OWL ontology</rdfs:comment>
    <owl:imports rdf:resource="http://www.example.org/food.owl"/>
  </owl:Ontology>
```

□ **rdf:about =""** : current document

□ **owl:import**

- ⇒ **if A imports B, and B imports C → A imports B, C**
- ⇒ **if A imports B, and B imports A → equivalent**
- ⇒ **Lite imports DL or Full → becomes an DL or Full**

Classes Description (1)

□ *owl:intersectionOf*

- ⇒ links a class to a list of class description
- ⇒ represent the “AND”

```
<owl:Class rdf:ID="Adult">
  <owl:intersectionOf rdf:parseType="Collection">
    <owl:Class rdf:about="#Person"/>
    <owl:restriction>
      <owl:onProperty rdf:resource="#age"/>
      <owl:someValuesFrom
        df:resource="http://www.w3.org/TR/@@/owl-ex-dt#over17"/>
    </owl:Restriction>
  </owl:intersectionOf>
</owl:Class>
```

Adult = Person □ ∃ age.over17

Classes Description (2)

□ *owl:unionOf*

⇒ **links a class to a list of class description**

⇒ **represent the “OR”**

```
<owl:Class>
  <owl:unionOf rdf:parseType="Collection">
    <owl:Class>
      <owl:oneOf rdf:parseType="Collection">
        <owl:Thing rdf:about="#Tosca" />
        <owl:Thing rdf:about="#Salome" />
      </owl:oneOf>
    </owl:Class>
    <owl:Class>
      <owl:oneOf rdf:parseType="Collection">
        <owl:Thing rdf:about="#Turandot" />
        <owl:Thing rdf:about="#Tosca" />
      </owl:oneOf>
    </owl:Class>
  </owl:unionOf>
</owl:Class>
```

Classes Description (3)

□ *owl:complementOf*

⇒ represent the “NOT”

⇒ example express “neither meat nor fish”

```
<owl:Class>
  <owl:complementOf>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#Meat"/>
        <owl:Class rdf:about="#Fish"/>
      </owl:unionOf>
    </owl:Class>
  </owl:complementOf>
</owl:Class>
```

$\neg (\text{Meat} \cup \text{Fish})$

Classes Description (4)

□ $\text{WhiteWine} \sqcap \text{hasSugar} . (\text{Dry} \sqcup \text{OffDry})$

```
<owl:Class rdf:ID="WhiteNonSweetWine">
  <owl:intersectionOf rdf:parseType="Collection">
    <owl:Class rdf:about="#WhiteWine" />
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasSugar" />
      <owl:allValuesFrom>
        <owl:Class>
          <owl:oneOf rdf:parseType="Collection">
            <owl:Item rdf:resource="#Dry" />
            <owl:Item rdf:resource="#OffDry" />
          </owl:oneOf>
        </owl:Class>
      </owl:allValuesFrom>
    </owl:Restriction> ...
```

Properties (1)

□ *owl:inverseOf*

```
<owl:ObjectProperty rdf:ID="hasChild">  
  <owl:inverseOf rdf:resource="#hasParent"/>  
</owl:ObjectProperty>
```

□ *owl:FunctionalProperty*

```
<owl:FunctionalProperty rdf:ID="husband">  
  <rdfs:domain rdf:resource="#Woman" />  
  <rdfs:range rdf:resource="#Man" />  
</owl:FunctionalProperty>  
  
<owl:ObjectProperty rdf:ID="husband">  
  <rdf:type rdf:resource="&owl;FunctionalProperty" />  
  <rdfs:domain rdf:resource="#Woman" />  
  <rdfs:range rdf:resource="#Man" />  
</owl:ObjectProperty>
```


Properties (2)

□ *owl:TransitiveProperty*

```
<owl:TransitiveProperty rdf:ID="subRegionOf">  
  <rdfs:domain rdf:resource="#Region"/>  
  <rdfs:range rdf:resource="#Region"/>  
</owl:TransitiveProperty>
```

□ *owl:SymmetricProperty*

```
<owl:SymmetricProperty rdf:ID="friendOf">  
  <rdfs:domain rdf:resource="#Human"/>  
  <rdfs:range rdf:resource="#Human"/>  
</owl:SymmetricProperty>
```

Requirements for OWL

✓ = already covered by DAML+ OIL

- ✓ Ontologies as distinct objects
- ✓ Unambiguous term referencing with URIs
- ✓ Explicit ontology extension
- ✓ Ontology metadata
- ✓ Versioning information
- ✓ Class definition primitives
- ✓ Property definition primitives
- ✓ Data types
- ✓ Class, property individual equivalence
- ❑ Local unique names assumptions
- ❑ Attaching information to statements
- ❑ Classes as instances
- ✓ Cardinality constraints
- ❑ User-displayable labels
- ❑ Supporting a character model
- ❑ Supporting a uniqueness of Unicode strings

Need for Automated Reasoning

□ **Ontology Development**

- ⇒ **Consistency check for classes and relations**
- ⇒ **Consistency check is very important for multi-authored ontologies**

□ **Ontology Integration**

- ⇒ **Find and assert inter-ontology relationships (for mapping and merging)**
- ⇒ **Automatically computes integrated class structure**

□ **Ontology Deployment**

- ⇒ **Consistency check for facts and individuals with respect to ontology**

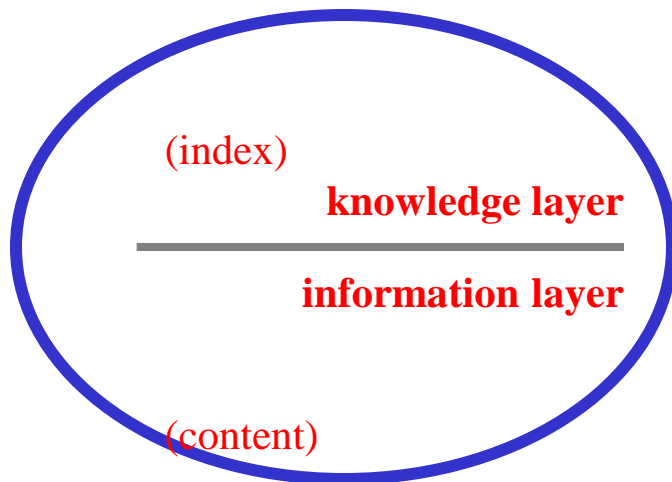
□ **The Semantic Web needs the logical layer on top**

Topic Maps

- ❑ In the early 1990's, DEC and O'Reilly were collaborating in the Davenport Group
 - ⇒ **DEC were bundling O'Reilly's UNIX documentation with their systems and wanted to create a Master Index of all documentation**
 - ⇒ **They tried merging indexes and failed – miserably!**
- ❑ This led to the insight that indexes really are knowledge structures
 - ⇒ **If their semantics could be captured formally, automated processing would be possible**
- ❑ This was the starting point for Topic Maps
- ❑ Topic Maps became an ISO project in 1996 and was approved in 2000
- ❑ XML Topic Maps (XTM) became part of ISO 13250 in 2001
- ❑ Dubbed the GPS of the World Wide Web by Charles Goldfarb, inventor of XML
 - ⇒ **One of its potentials is to enable giant indexes for Gisle Hannemyr's giant printing press**
 - ⇒ **But its immediate potential in smaller scale applications, e.g. within an enterprise**

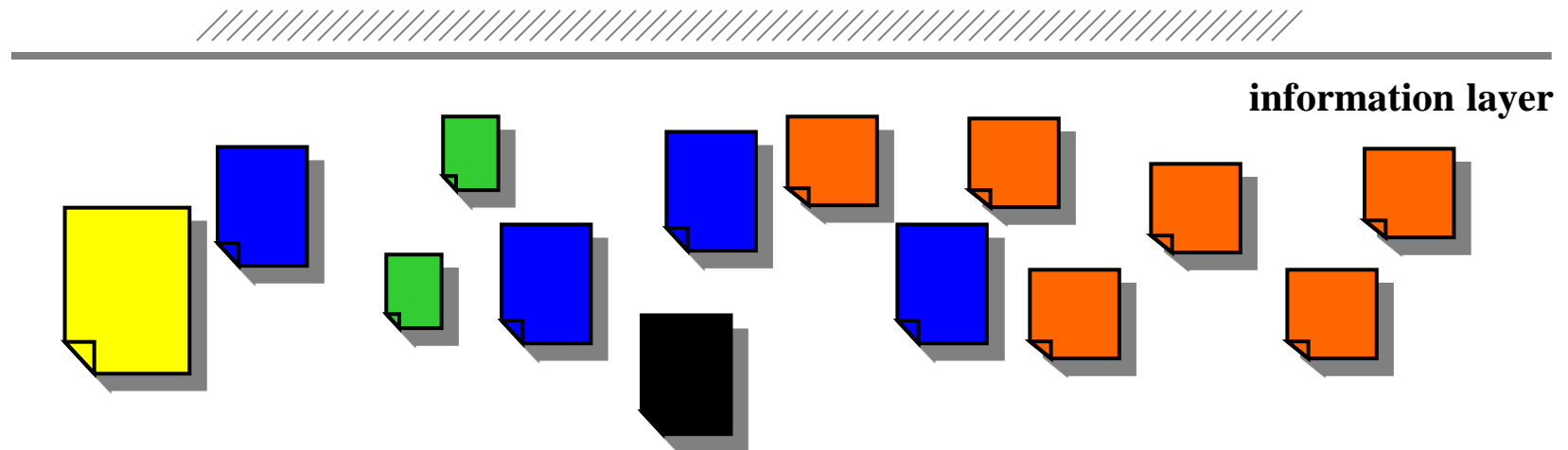
The 2-Layer Topic Map Model

- ❑ The core concepts of Topic Maps are based on those of the back-of-book index
- ❑ The same basic concepts have been extended and generalized for use with digital information
- ❑ Envisage a 2-layer data model consisting of
 - ⇒ **a set of information resources (below), and**
 - ⇒ **a “knowledge map” (above)**
- ❑ This is like the division of a book into content and index



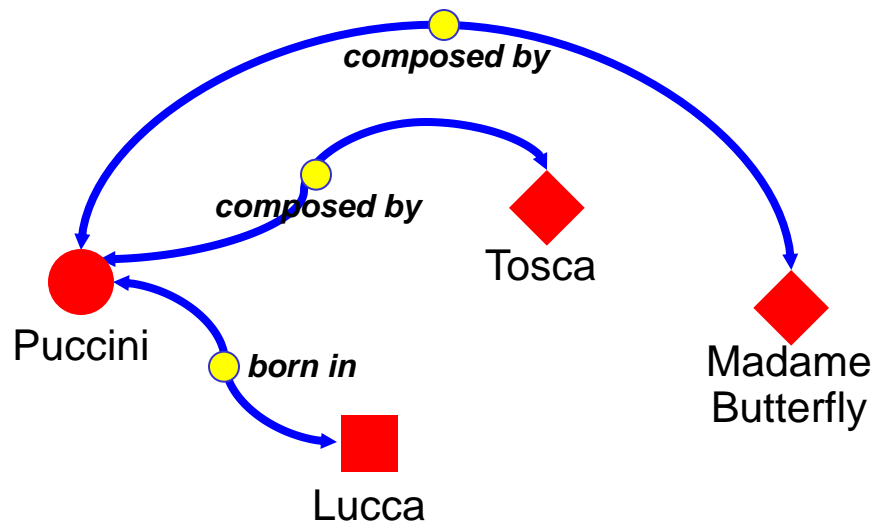
The Information Layer in Topic Map

- The lower layer contains the content
 - ⇒ usually digital, but need not be
 - ⇒ can be in any format or notation
 - ⇒ can be text, graphics, video, audio, etc.
- This is like the content of the book to which the back-of-book index belongs



The Knowledge Layer of Topic Map

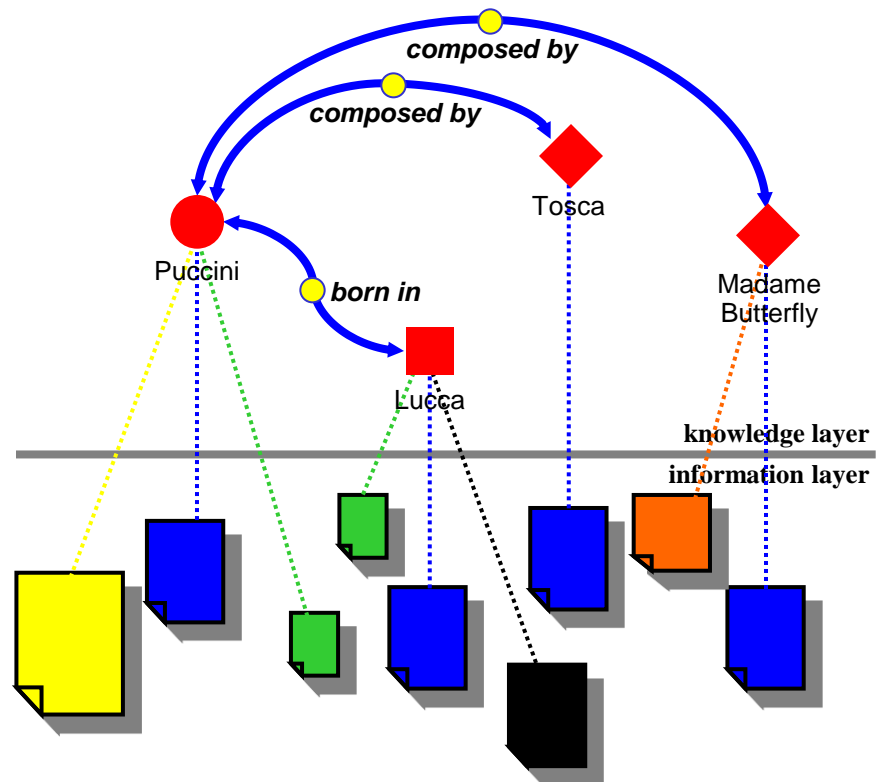
- The upper layer consists of topics and associations
 - ⇒ **Topics** represent the subjects that the information is about
 - ☑ Like the list of topics that forms a back-of-book index
 - ⇒ **Associations** represent relationships between those subjects
 - ☑ Like “see also” relationships in a back-of-book index



knowledge layer

Core Topic Maps Concepts

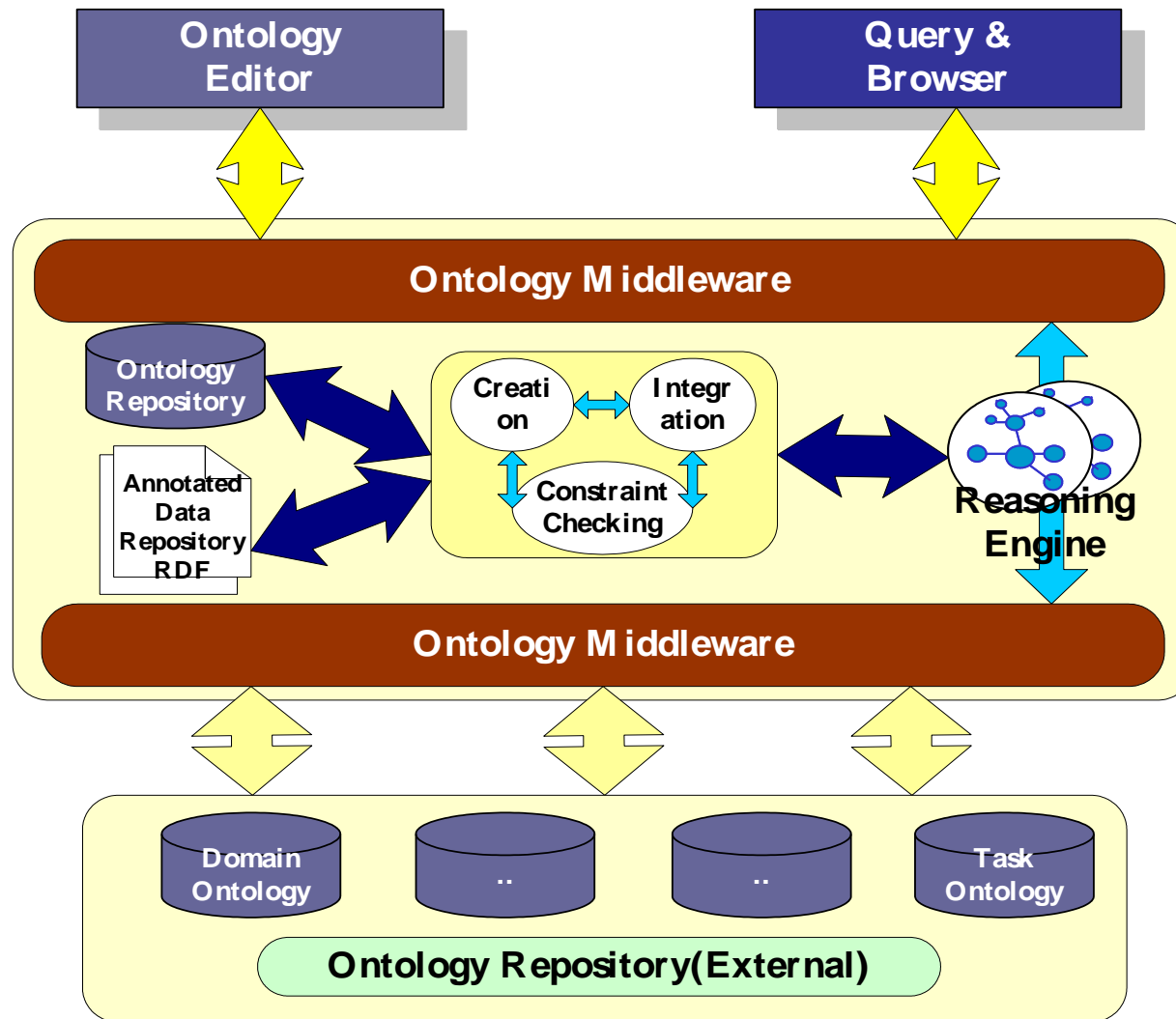
- **A pool of information**
 - any type or format
- **A knowledge layer**
- **Topics**
 - a set of knowledge topics for the domain in question
- **Associations**
 - expressing relationships between knowledge topics
- **Occurrences**
 - information that is relevant in some way to a given knowledge topic
- **= The TAO of Topic Maps**



Agenda

- Introduction
- Representations in the Semantic Web
- Languages
- **Tools**
 - ⇒ **Tool architecture**
 - ⇒ **Ontology builder & browser**
 - ⇒ **Ontology modeling tool**
 - ⇒ **Ontology integration tool**
 - ⇒ **Annotator**
 - ⇒ **Reasoner**
- Applications
- Conclusion

Ontology Tool Architecture



Query and Brower : RDF*ferret*

The screenshot shows the RDFferret web interface in a browser window. The interface has a light blue background and a search form with the following elements:

- Search Form:** Includes a dropdown menu for "[9] Sculpture", a text input field for "text contains:", and buttons for "Go!" and "Clear".
- Filters:** Checkboxes for "Match case", "Match words exactly", "Search page titles only", and "Compact display".
- Display Options:** "Show me" set to "10 results" and "Sorted by" set to "relevance".
- Super-classes:** A link to "super-classes: Artifact, Resource Help".

Results: restricted to Sculpture (1-9 of 9)

- 1. David Sculpture**
material: Marble
has_style: <http://www.ibiblio.org/vm/paint/Al/it-ren/>
exhibited: <http://www.thais.it/scultura/geda.htm>
name: David
MICHELANGELO di Lodovico Buonarroti Simoni (b. 1475, Caprese, d. 1564, Roma) - David - 1504 - Marble, height 434 cm - Galleria dell'Accademia, Florence - In 1501 Michelangelo was
- 2. Sculpteur/Porte de l'enfer Sculpture**
exhibited: <http://www.musee-rodin.fr/>
name: The Gates of Hell
material: bronze
has_style: http://www.britannica.com/bcom/eb/article/3/0_5716_117493+1_00.html
THE GATES OF HELL - Antoinette Le Normand-Romain Not far from The Thinker stands Rodin's monumental masterpiece, The Gates of Hell , installed in the garden of the Museum in 1937. By a decree of 16
- 3. Pietà Sculpture**
material: Marble
has_style: <http://www.ibiblio.org/vm/paint/Al/it-ren/>
exhibited: <http://www.roma2000.it/sc/piet.html>
name: Pietà
MICHELANGELO di Lodovico Buonarroti Simoni (b. 1475, Caprese, d. 1564, Roma) - Pietà - 1499 - Marble, height 174 cm, width at the base 195 cm - Basilica di San Pietro, Vatican - In the

Query and Browser : Ontoshare

The screenshot displays the OntoShare web application interface, titled "OntoShare - alistair duke". The interface is divided into several sections:

- Concepts:** A hierarchical tree structure on the left side, showing various concepts and their counts. The tree is expanded to show "Instant Messaging(12)" and "XML(6)".
- Documents in Instant Messaging:** A table listing documents related to instant messaging. The table has columns for Document title, Submitted By, Date, and Relevance.
- Document information:** A section showing details for a selected document, including the full document URL, annotation, comments, and summary.
- My concepts:** A list of concepts that the user has added or is interested in.
- Footer:** A row of buttons for user actions: "Share new document", "My profile", "Documents for me", "Search", "Interesting", "Not interesting", and "Add comment".

Documents in Instant Messaging Table:

Document title	Submitted By	Date	Relevance
The Register	Martin Crossley	2002-05-27	<input type="text" value=""/>
CW360° - Article Page	Nick Kings	2002-05-22	<input type="text" value=""/>
Text Article: Microsoft pledges Passport openness Microsoft: "...	Martin Crossley	2002-05-21	<input type="text" value=""/>
Developing Microsoft .NET Web Service Clients for EJB Web Servi...	Tim Stevens	2002-05-21	<input type="text" value=""/>
Instant Messaging Planet: Wireless IM: Converse to Acquire Odigo	Martin Crossley	2002-05-20	<input type="text" value=""/>
Text Article: Microsoft hits out at Passport privacy slur Users hav...	Martin Crossley	2002-05-17	<input type="text" value=""/>
BBC News DOT LIFE It's a hamster on your mobile. Or possibly...	Alistair Duke	2002-05-15	<input type="text" value=""/>
Text Article: EDS bans instant messaging But won't say why... E...	Martin Crossley	2002-05-09	<input type="text" value=""/>
The Register	Martin Crossley	2002-05-07	<input type="text" value=""/>
Content Wire Preview - Home - Fresh Picks	Martin Crossley	2002-05-03	<input type="text" value=""/>

Document information:

Full document: <http://www.theregister.co.uk/content/4/25433.html>

Annotation: Microsoft Passport may be incompatible with European data protection law

Comments:
No comments have been added yet.

Summary:
EU looks at MS Passport for privacy infringement By John Lettice Posted: 24/05/2002 at 18:39 GMT Microsoft's problems with Brussels have been compounded by news that the European Commission is investigating whether Passport is compatible with European data protection law. European law is substantially tougher than the US equivalent in this area, and Microsoft's record on database control is somewhat patchy, so it wouldn't be a big surprise if the Commission decided there was a problem here. The news of the investigation came in a letter to Netherlands European Parliament member Erik Meijer, who had raised numerous objections to Passport, and accused Microsoft of "surreptitiously" passing on registration information to "unknown parties." The Commission's response is that it is looking into it "as a matter of

Editor : Protégé-2000

The screenshot displays the Protégé-2000 editor interface. On the left, a project browser shows a hierarchy of classes: **Person** (Class), **Employee** (Class), **Columnist** (Class), **Editor** (Class), **Reporter** (Class), and **Salesperson** (Class). A red arrow points from the **Employee** class in the hierarchy to a list of slots. The slots are: **current_job_title**, **date_hired**, **salary**, **name**, and **other_information**. Another red arrow points from the **Employee** class to the **salary** slot. The main window shows the **Documentation** tab with the text "Editors are responsible for the content of sections." and the **Constraints** tab with the constraint "editor-employees-salary-constraint". The **Other Facets** tab shows the facets for the **Employee** class: **Section** and **Employee**.

Project: newspaper Protégé-2000 (C:\Program Files\Protégé-2000\examples\newspaper\newspaper.pprj)

Class Hierarchy:

- Person (Class)
- Employee (Class)
 - Columnist (Class)
 - Editor (Class)
 - Reporter (Class)
 - Salesperson (Class)

Slots for Employee:

- current_job_title
- date_hired
- salary
- name
- other_information

Documentation: Editors are responsible for the content of sections.

Constraints: editor-employees-salary-constraint

Other Facets:

- Section
- Employee

Modeling Tool: SemTalk

Visio 2000

File Edit View Insert Format Extras Shape SemTalk Fenster Hilfe

Standard Arial 12pt F K U A B I L

SemTalk: zeichn...

Zeichnung1:Seite1

Model

- Diagrams
 - Generic
 - Seite1 (Class)
 - Train
 - High Speed Train
 - Local Train
 - Station
 - Central Station
- Objects
 - Instance
 - Station
 - Central Station
 - Train
 - High Speed Train
 - Local Train
- Relations

CLASS

Class

Property

sub Class...

Comment

definition Of

Dokument

Seite1

Zeichenblatt 1/1 Status: Bereit

c:\eigene dateien\rd\se\trains.vsd - Microsoft Internet Explorer von T-Online

File Edit View Favorites Extras ?

Adresse c:\eigene dateien\rd\se\trains.htm Wechseln zu

Diagrams

- Vehicle Model
 - Generic
 - Classes
 - Instances
 - Classes
 - Relations

Train

Name	Value
Number	110
SuperClass	Vehicle
Sub Class	Long Distance Train
Instance	Airport Express
stops at	Station
Diagram	Classes

Merge & Integration Tools : PROMPT

The screenshot displays the Protégé-2000 PROMPT interface. The main window is titled "<new> Protégé-2000" and has a menu bar with "Project", "Edit", "Window", "Help", and "Prompt". Below the menu bar is a toolbar with icons for file operations and a tabbed interface with "Prompt", "Classes", "Slots", "Forms", "Instances", and "Queries".

The "Prompt" tab is active, showing a "Suggestions" pane on the left and a "Result classes" pane on the right. The "Suggestions" pane has a "To Do list" table with columns "Name", "Arg1", "Arg2", and "Params". The "Result classes" pane shows a "merged" list of classes: ":THING A", ":SYSTEM-CLASS A", "Reservation_record--air A", and "Reservation". A red arrow points from the "Reservation" class in the result pane to the "Reservation" class in the "Template Slots" pane.

The "To Do list" table contains the following suggestions:

Name	Arg1	Arg2	Params
copy	Record	air	
copy	Award_travel	air	
copy	Customer	air	
copy	Itinerary	air	
copy	Individual	air	
copy	Location	car	
copy	Vehicle	car	
copy	Driver	car	
merge	Reservation_record--air	Reservation	
merge	Check	air	
merge	Credit_card	air	
copy	Aircraft	air	
copy	Flight	air	
copy	Payment_record	air	

The "Reason for selected suggestion" pane shows the message: "Record air is a superclass of Reservation_record--air". A "Do It" button is at the bottom of the suggestions pane.

The "Template Slots" pane shows a table of slots for the "Reservation" class:

Name	Type	Cardinality	
customer_type--air	Class	single	parent
customer_type--car	Symbol	single	allowe
drop_off_date--car	String	single	
drop_off_location--car	Instance	single	classe
itinerary--air	Instance	single	classe
pick_up_date--car	String	single	
pick_up_location--car	Instance	single	classe
record_locator--air	String	single	
reservation_number--car	Integer	single	
traveler--air	Instance	multiple	classe
traveler--car	Instance	multiple	classe
vehicle--car	Instance	single	classe

Reasoner : FaCT

CORBA-FaCT request form - Microsoft Internet Explorer

파일(F) 편집(E) 보기(V) 즐겨찾기(A) 도구(T) 도움말(H)

뒤로 앞으로 중지 검색 즐겨찾기 미디어

주소 http://www.cs.man.ac.uk/~horrocks/FaCT/CORBA-FaCT.html 이동 연결

Google 웹 검색 사이트 검색 PageRank 페이지 정보 위로 하이라이트

CORBA-FaCT Request Form

Please take a few moments to complete this form before downloading the **alpha-test** version of **CORBA-FaCT** system. This information is only used to keep track of who is using FaCT and will not be accessible to anyone outside the FaCT research group. We may (very) occasionally contact you to enquire if you are using FaCT in any interesting applications.

Please enter the following information

Name

Email

Institution

Address

Send Clean Form

완료 인터넷

Reasoner : OilEd

The screenshot displays the OilEd 3.4 application window. The main interface is divided into several panes:

- Classes List:** A vertical list of classes on the left, including "A Clamp #1", "Absolute Pressure #1", "ACI #1", "ACR #1", "AD #1", "Adjustable Bouyancy Life J", "AI #1", "Air embolism #1", "Alternate Air Source #1", "Alveoli #1", "Ambient pressure #1", "Anoxia #1", "Apnea #1", "Argon #1", "Ascent Check Depth #1", "Ascent Rate #1", "Atmospheric Pressure #1", "Auto Air/Air II #1", "AV #1", "Backup #1", "Bailout #1", "Bar #1", "Barotrauma #1", "Boat #1", "Body Part #1", "Bottom Mix #1", "Bottom Time #1", "Bradycardia #1", and "Buddy #1".
- Properties:** A pane on the right showing "SubclassOf" and "SameClassAs" properties.
- Server Window:** A small window titled "Server" with tabs for "Window", "Panic", "Logging", and "KB". The "Logging" tab is active, showing a status of "Logging".
- FaCT Client:** A window titled "FaCT Client" with tabs for "Connection", "Transaction", "Tell", "Utilities", and "Ask". The "Connection" tab is active, showing a status of "Status:" and a result of "Result:". The "Log" tab is also active, showing a log of commands and responses.

The FaCT Client console shows the following log entries:

```
Allegro Common Lisp Console - [shiq-app.dxl]
Allegro CL Enterprise Edition 5.0.1 [Windows/x86] (8/16/1 14:02)
Copyright (C) 1985-1999, Franz Inc., Berkeley, CA, USA. All Rights Reserved
Assumed hostname: KORODA
ORblink server uses port 1040.
HTTP server uses port 8000.
Binding lispserver FaCTLisp.
USER(1): Got command "GET /CosNaming HTTP/1.1" on #<TEXT stream socket
connected from
koroda/8000
to
koroda/2146
@
#x20d59112>
GET: Firing handler #<Function HTTP-NAMESERVER> with arguments (("CosNam
Got command "GET /CosNaming HTTP/1.1" on #<TEXT stream socket
connected from
koroda/8000
to
koroda/2149
@
#x20d6ada2>
GET: Firing handler #<Function HTTP-NAMESERVER> with arguments (("CosNam
```

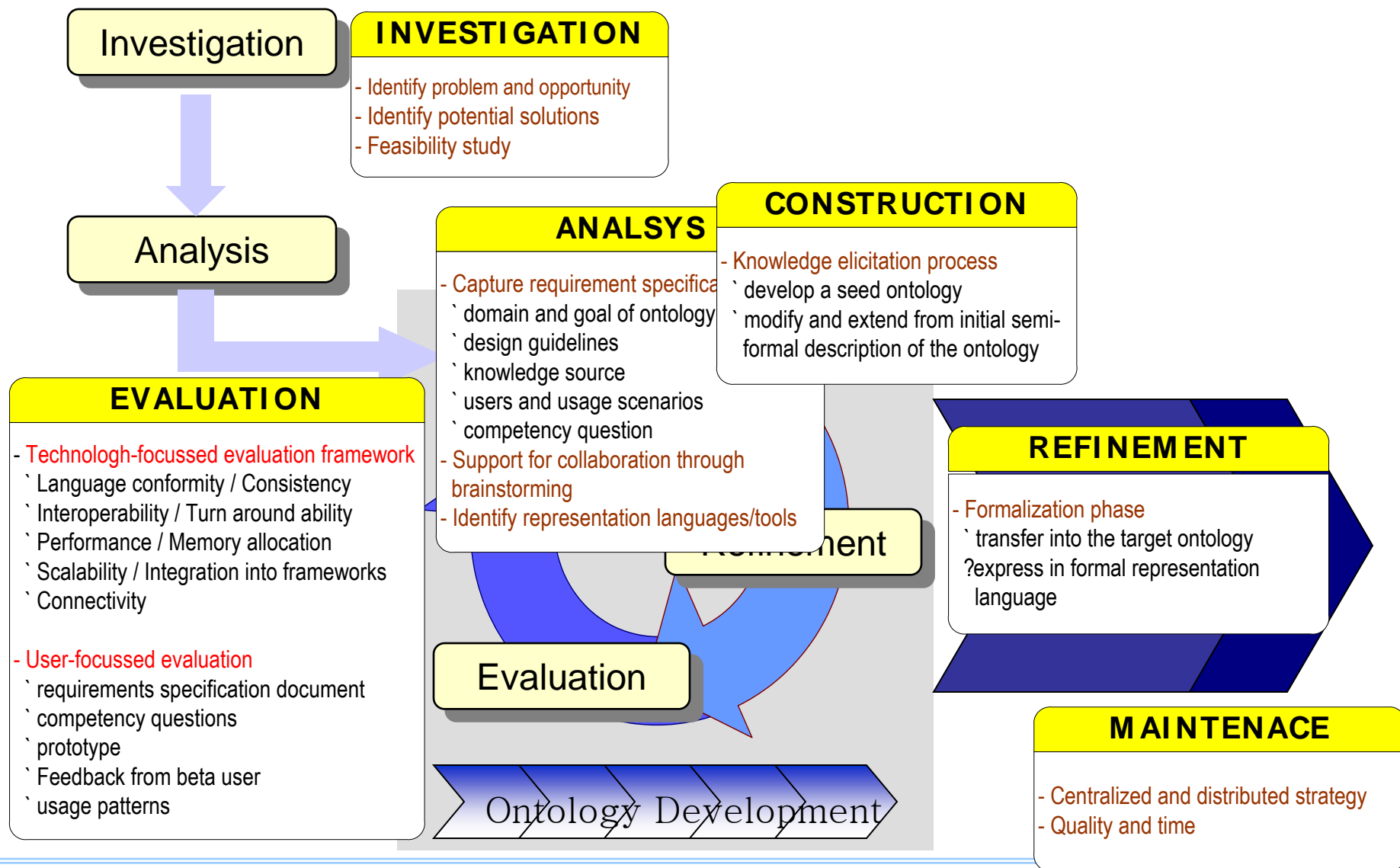

Tools

Feature	OILEd	OntoEdit	Ontolingua	OpenKnoME	Protégé-2000	WebODE
Developers	Uni. of Manchester	Ontoprise	KSL(Stanford Uni.)	Uni. of Manchester	SMI(Stanford)	Ontology Group(UPM)
Availiability	Open source	Freeware	Free Web Access	Freeware	Open source	Free Web Access
Architecture	standalone	Standalone	Client/Server	Client/Server	Standalone	3-tier
Extensibility	no	Plug-ins	None	None	Plug-ins	Plug-ins
Import for Language	XML RDFS(S) Flogic DAML+ OIL	XML RDFS(S) Flogic DAML+ OIL	Ontolingua IDL KIF	GRAIL GALEN IR	XML RDF(S) XML Schema	XML RDF(S) CARIN
Export to Language	OIL RDF(S) DAML+ OIL SHIQ	XML RDF(S) Flogic DAML+ OIL	KIF3.0 CLIPS LOOM OKBC PROLOG	GRAIL CLIPS GALEN IR HTML	XML RDF(S) XML(S) Flogic JAVA	XML RDF(S) OIL DAML Prolog
KR paradigm	DL(DAML+ OIL)	Frames+ FO L	Frames+ FO L	DL(GRAIL)	Frames+ FO L	Frames+ FO L
Graphical taxonomy	No	No	Yes	No	Yes	Yes
Collaborative working	No	No	Yes	Yes	No	Yes

Agenda

- Introduction
- Representations in the Semantic Web
- Languages
- Tools
- **Applications**
 - ⇒ **Methodologies**
 - ⇒ **Application areas and use case**
 - ⇒ **Future of the Semantic Web technologies**
- Conclusion

An Ontology Building Life-cycle



Methodologies

□ Building ontologies

- ⇒ **Cyc Methodology / Uschold and King / Gruninger and Fox**
- ⇒ **KACTUS Methodology / METHONTOLOGY / SENSUS Methodology**

□ Cooperative Ontologies

- ⇒ **CO4 methodology / (KA)² methodology**

□ Learning Ontologies

- ⇒ **Aussenac-Gille's and colleagues methodology**

□ Merge Ontologies

- ⇒ **FCA-merge / PROMPT**

□ Evaluation Ontologies

- ⇒ **Guarino's group methodology**
- ⇒ **Gomez Perez's evaluation methodology**

General Use Case

- ❑ Web Portals
- ❑ Multimedia collections
- ❑ Corporate web site management
- ❑ Design documentation
- ❑ Agents and services
- ❑ Ubiquitous computing

Research and Commercial Issues

□ Ontology Representation Languages

- ⇒ Further implementation in upper layers on the top of DAML+OIL/OWL
- ⇒ DAML-S, DAML-RULES, OWL-Med (some are domain-specific)
- ⇒ Full-fledged Description Logic (*Expressivity & Tractability*)
- ⇒ Nonmonotonic reasoning support

□ Management Tools

- ⇒ Corporate memory management through agents
- ⇒ GUI-based knowledge transformation tool (syntactic, semantic, and semiotic)
- ⇒ Integrated ontology management system

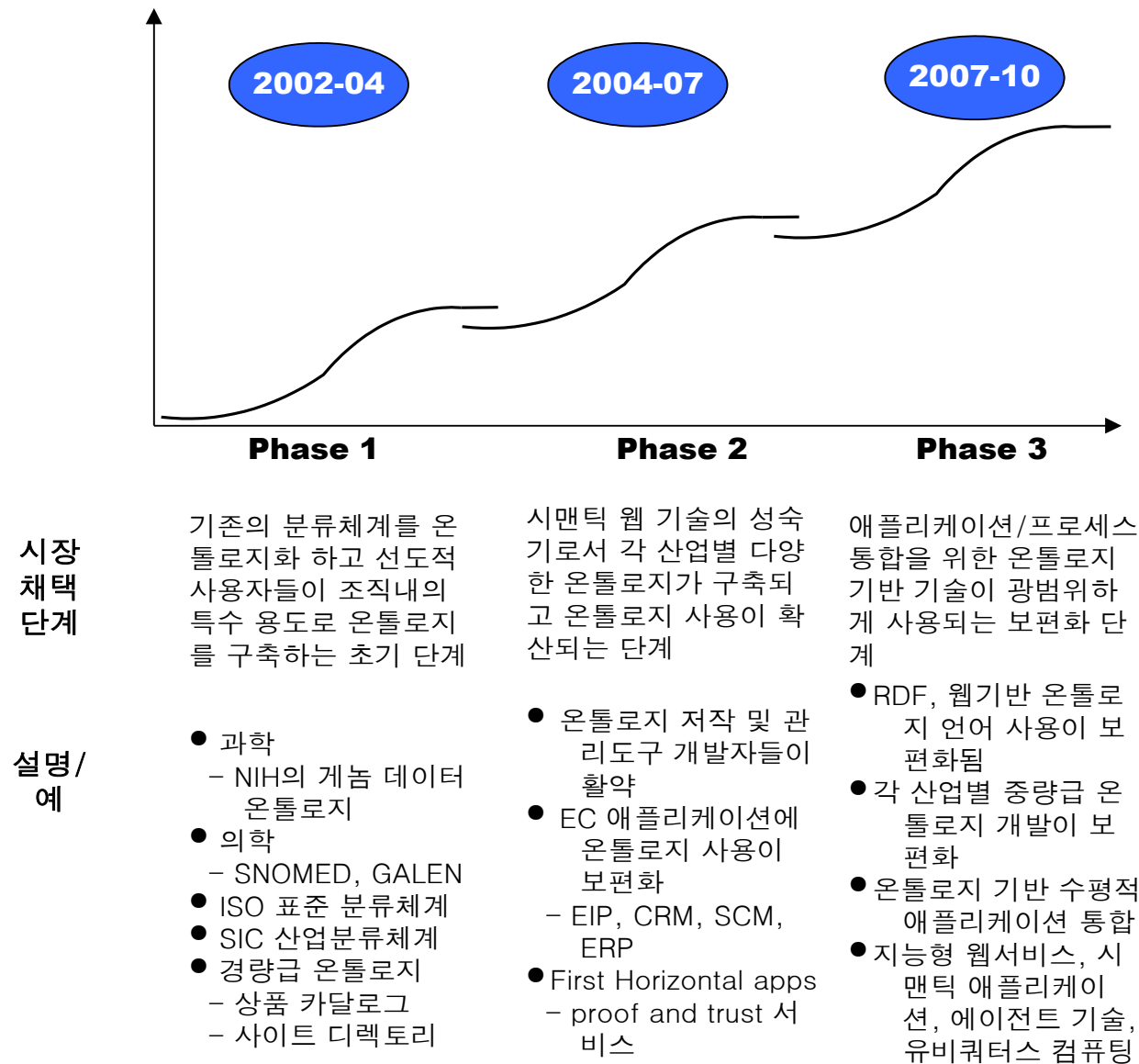
□ Methodologies

- ⇒ Collaborative ontology engineering
- ⇒ Domain-specific & tool-specific methodologies

□ Business Applications

- ⇒ Information Search
- ⇒ Skills management
- ⇒ Exchanging knowledge in a virtual organization

Future of the Semantic Web Technology



Source: Christian Ohms, 2002

Agenda

- Introduction
- Representations in the Semantic Web
- Languages
- Tools
- Applications
- Conclusion

The Semantic Web – A Roadmap

□ What the Semantic Web is NOT ...

- ⇒ **The Semantic Web is not Artificial Intelligence**
- ⇒ **The Semantic Web does not allow arbitrary complexity**
- ⇒ **The Semantic Web is not something that will ever be complete**

□ What the Semantic Web IS ...

- ⇒ **A great vision**
- ⇒ **Something that will be built over time**
- ⇒ **An emergent property of the global effort towards standardization around XML**

*“Ask not what the Semantic Web
can do for you,
ask what you can do for the
Semantic Web”*

Hans-Georg Stork

Thank You !!!