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 Reliabilty 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

Goooooooooogle ▶

결과 페이지: 1 2 3 4 5 6 7 8 9 10 다음

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 movin g 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

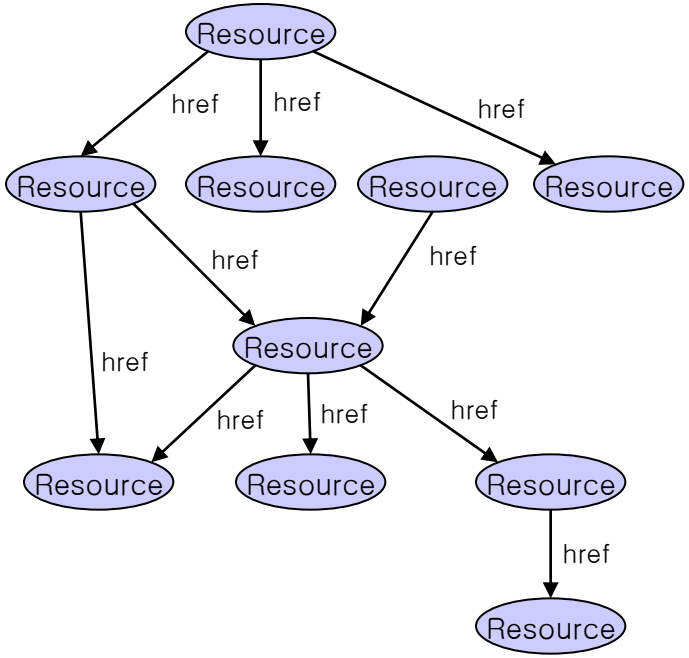
The Semantic Web

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

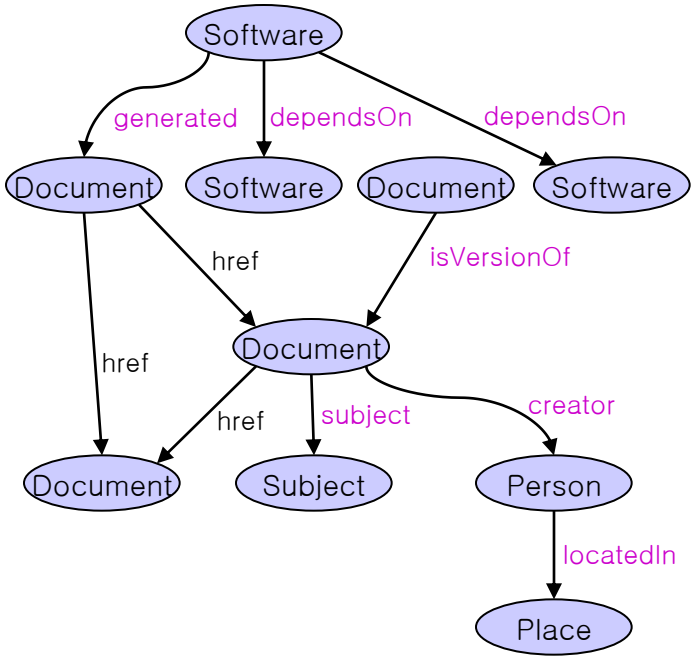
Source: Scientific American: The Semantic Web
TIM BERNERS-LEE, JAMES HENDLER and ORA LASSILA
http://www.sciam.com/2001/0501issue/0501berners-lee.html

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



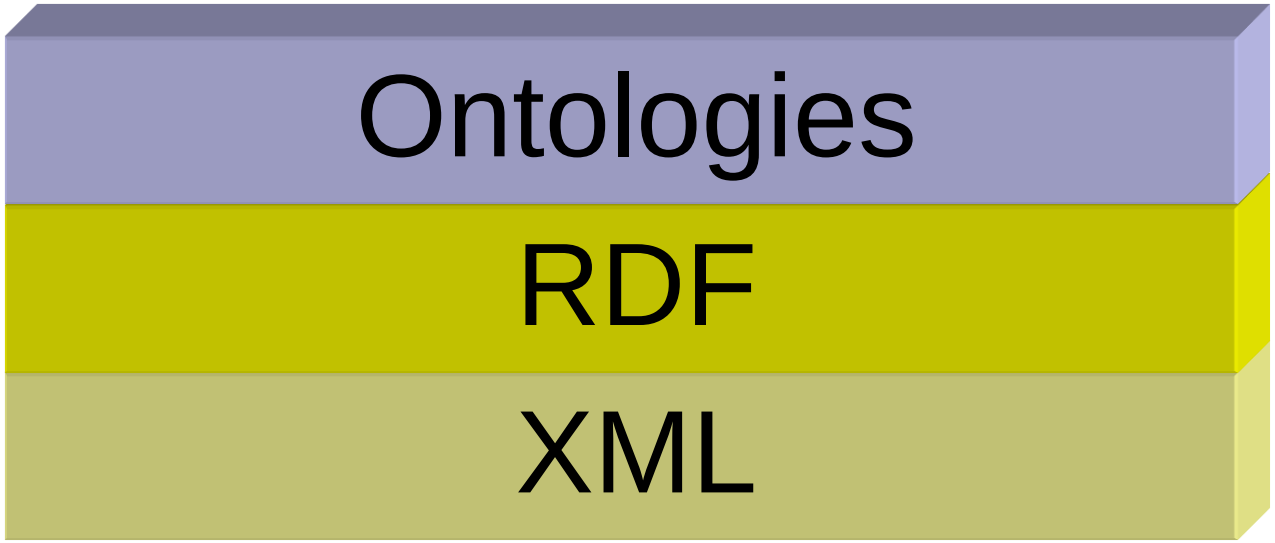
• More information available

More Intelligent Knowledge Exchange

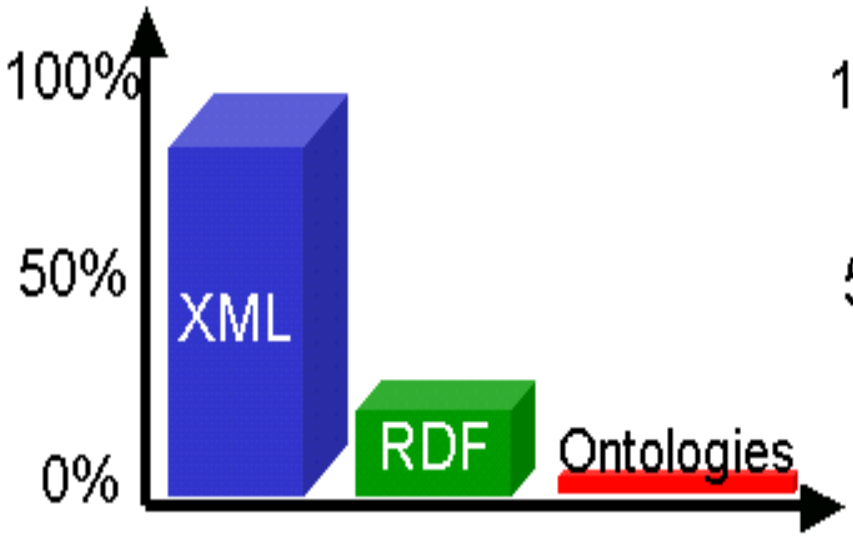
How the Semantic Web Will Be Possible

- 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**

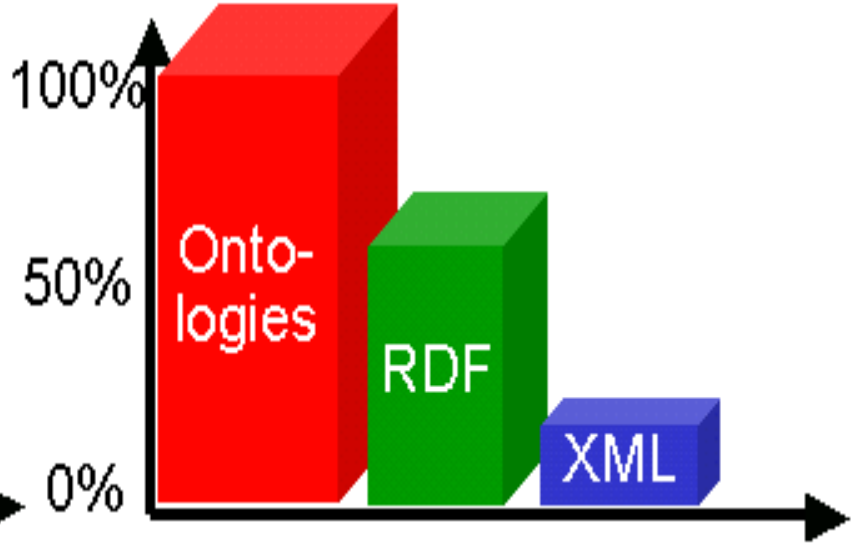
The 3 cornerstones of the Semantic Web



Semantic Web Awareness and Its Business Relevance



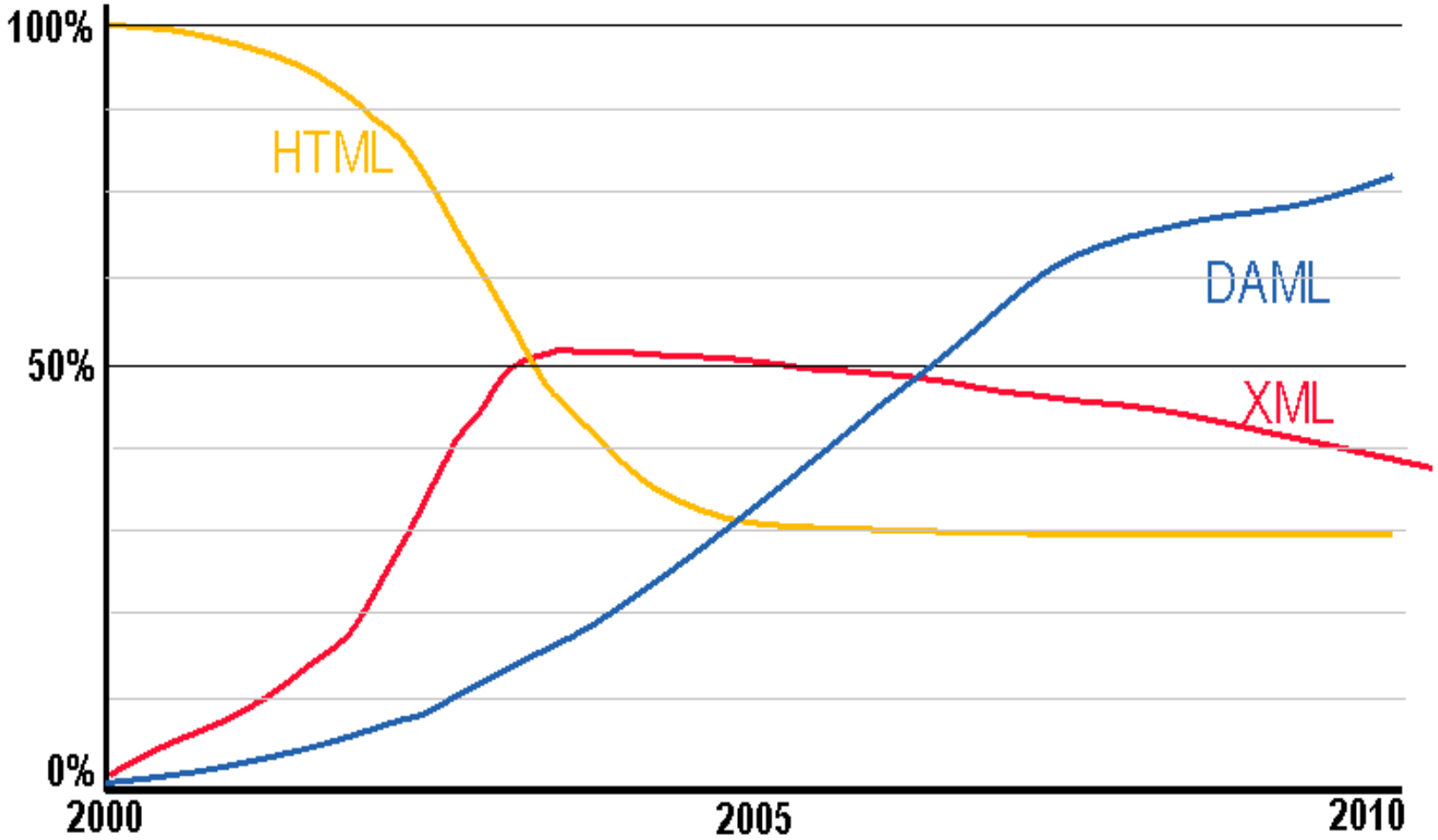
IT 의사결정권자의
시맨틱웹 인지도



시맨틱웹 컴포넌트의
비즈니스 관련성

Chris Horak, 2002

Future Trends of Markup Languages

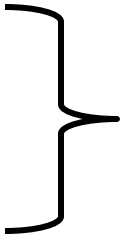


Source: DARPA 2001

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

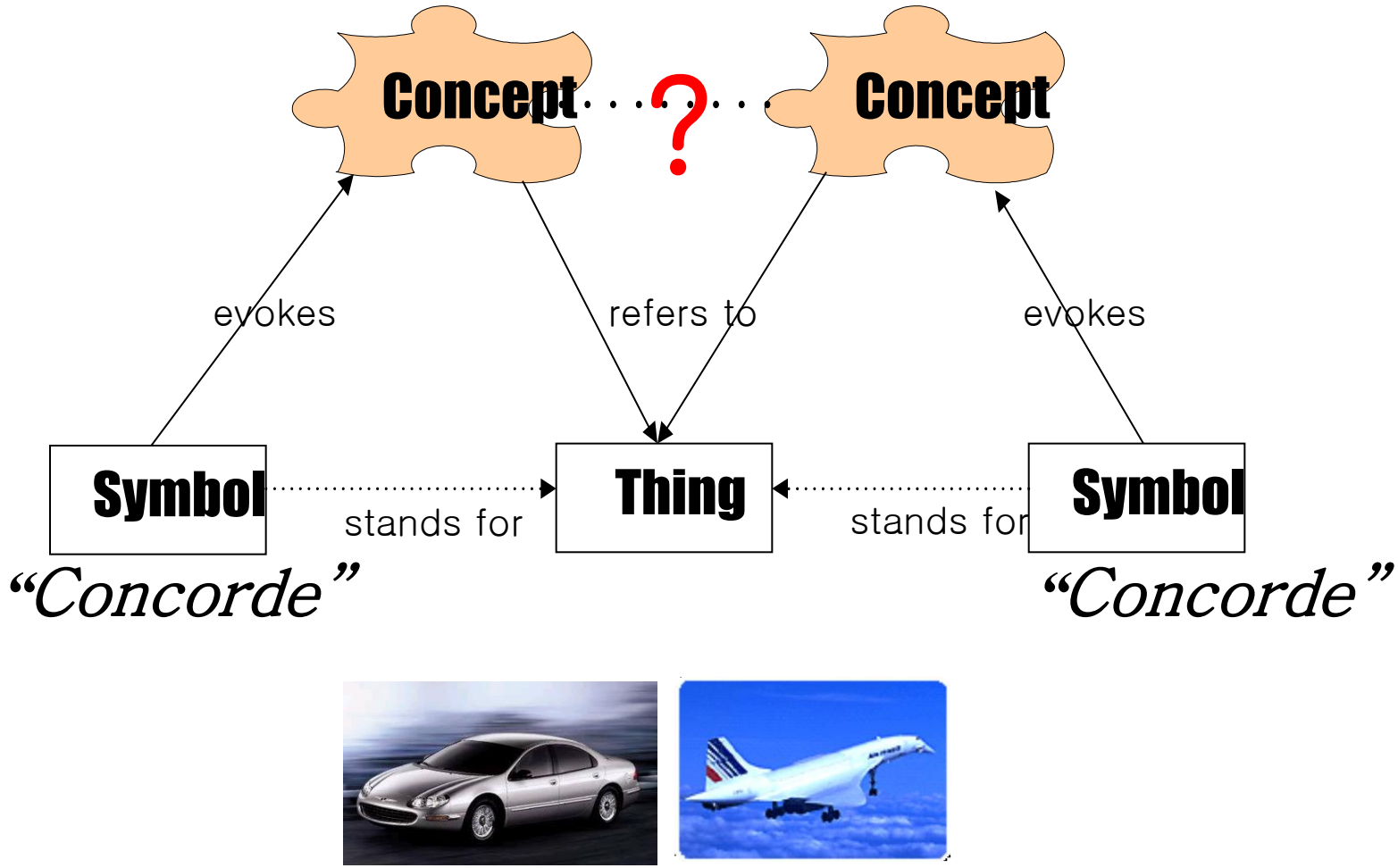
Agenda

- Introduction
- Representations in the Semantic Web
  - ⇨ **Meaning**
  - ⇨ **Knowledge**
  - ⇨ **Ontology**
- Languages
- Tools
- Applications
- Conclusions

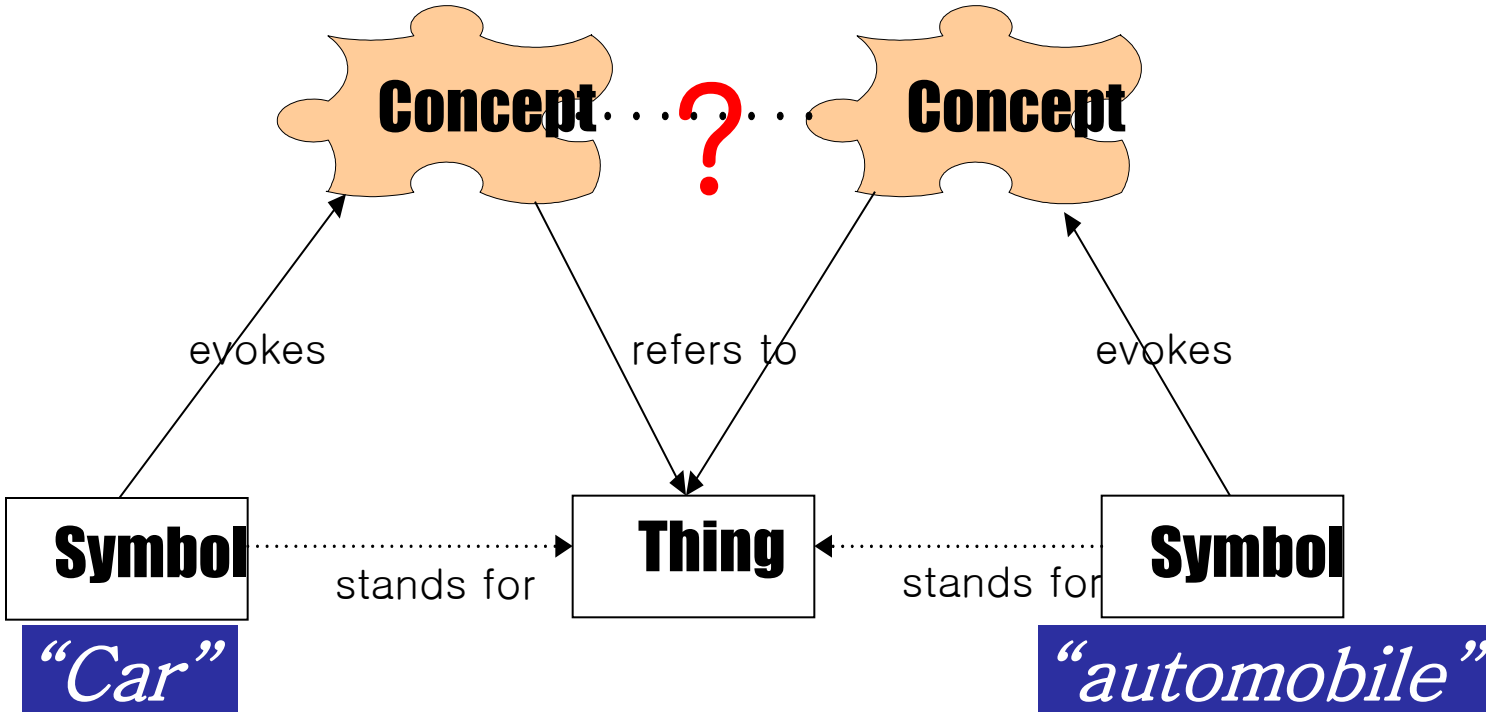
Definition of Re-presentation

- A relationship between two domains where the first is meant to "*stand for*" the second.
- The first domain, the represneter, 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

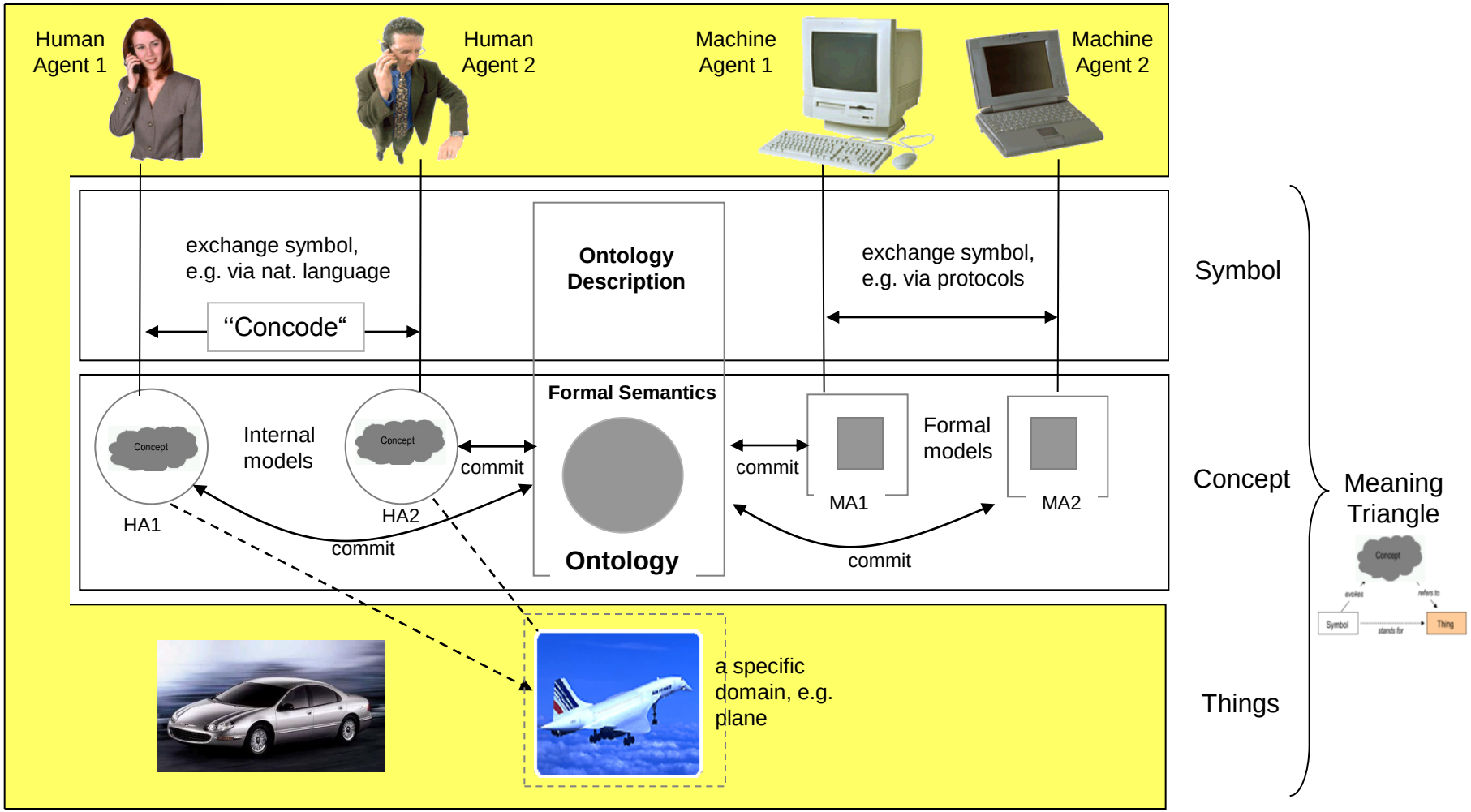


Meaning and Human Communication



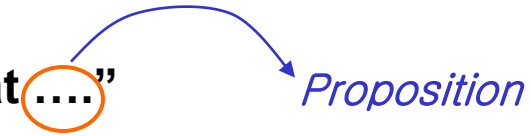
Human and Machine communication

[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

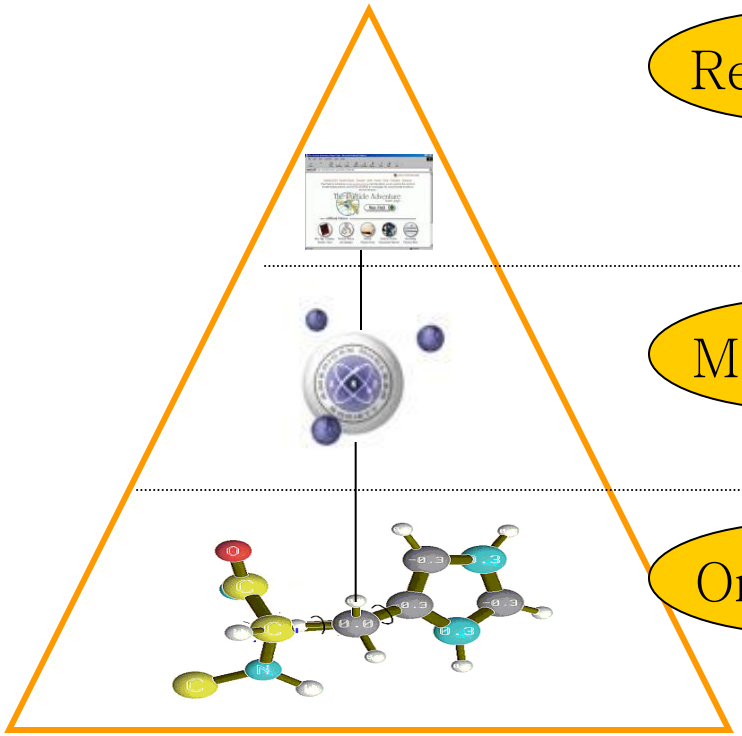
Knowledge Lego: *Reusable Concepts*

[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



Resource

http://www.aphis.usda.gov/oa/pubs/qafmd301.html

Metadata

- FMD(Foot and Mouth Disease), Animal Welfare Information Center
- Kanghwado, South Korea

Ontology

FMD **is caused by** an RNA virus.Kanghwado **is located in** Gyeonggi, South Korea.

Neither this particular resource nor its metadata explicitly mention the recent occur of a RNA related virus in Gyeonggi, South Korea.

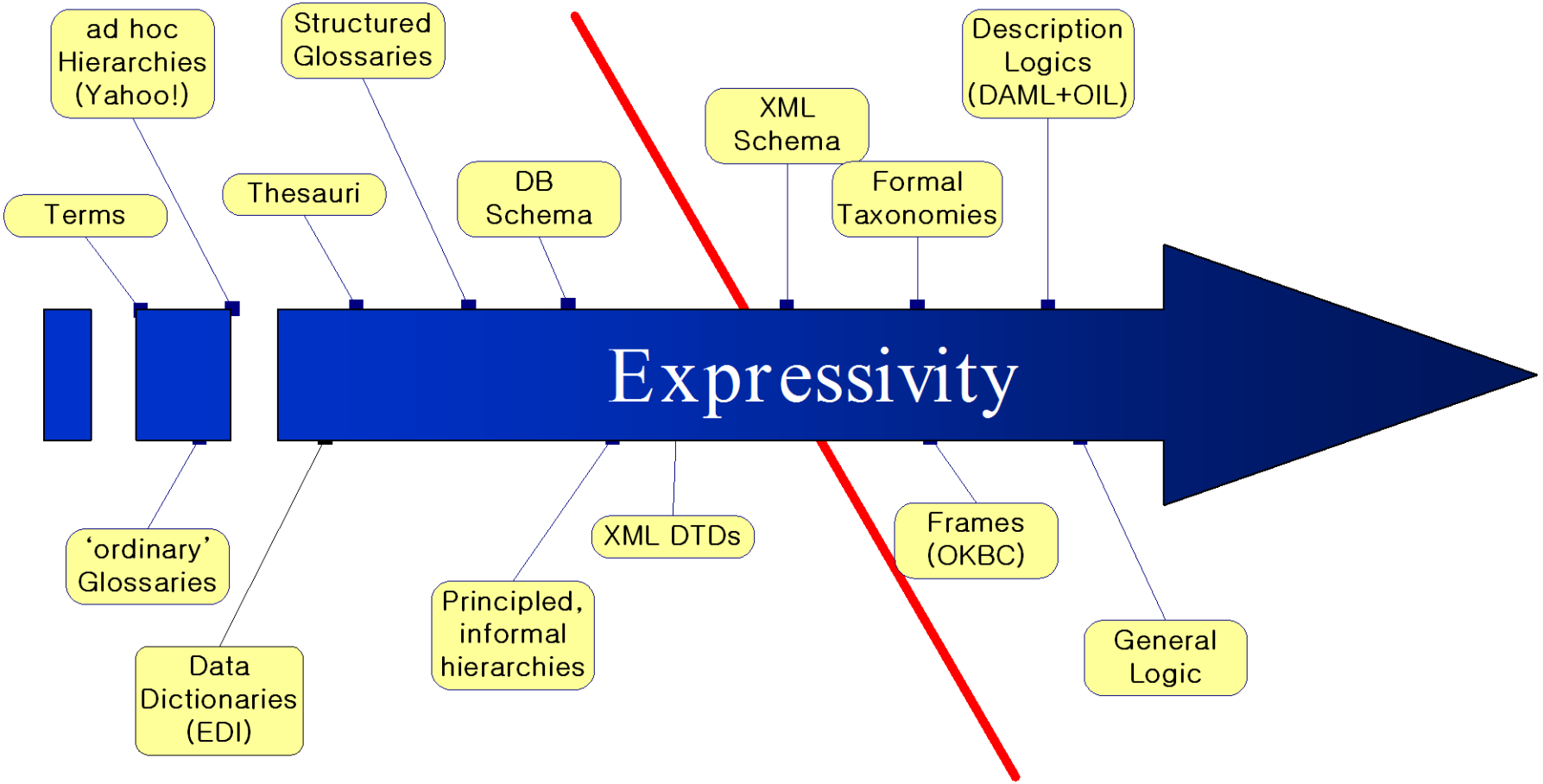
Only an assisted search that maps metadata to underlying ontologies could retrieve this resou in response to the query "***the recent occur of a RNA related virus in Gyeonggi, South Korea.***"

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

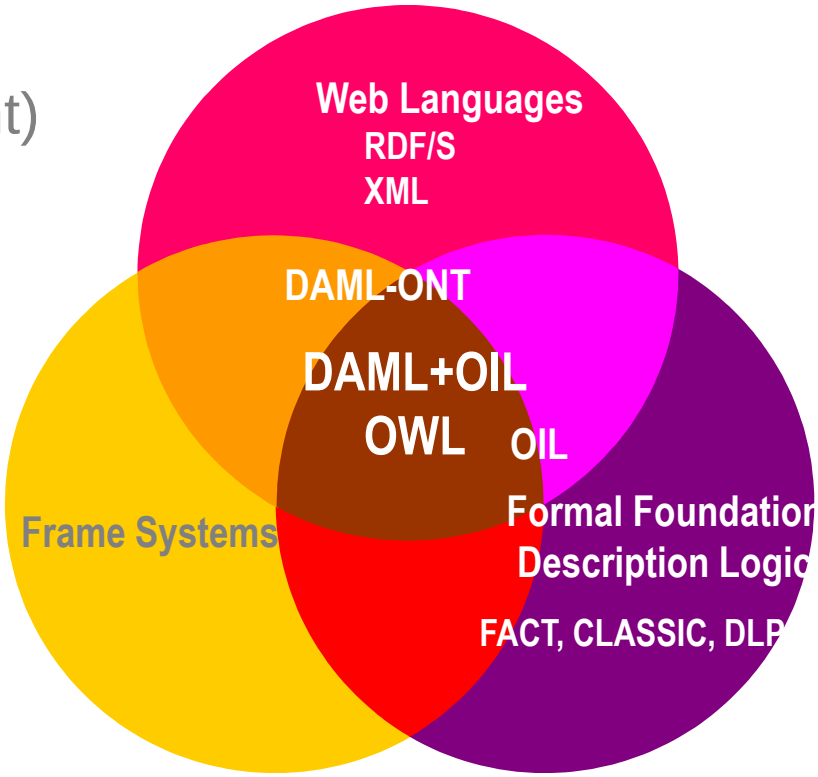
- Should provide *formal syntax* and *formal semantics* to enable automated processing of the contents
- Should provide standardized vocabulary referring to real-world semantics so that machine and human agents can share information and knowledge → *Ontology*

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)

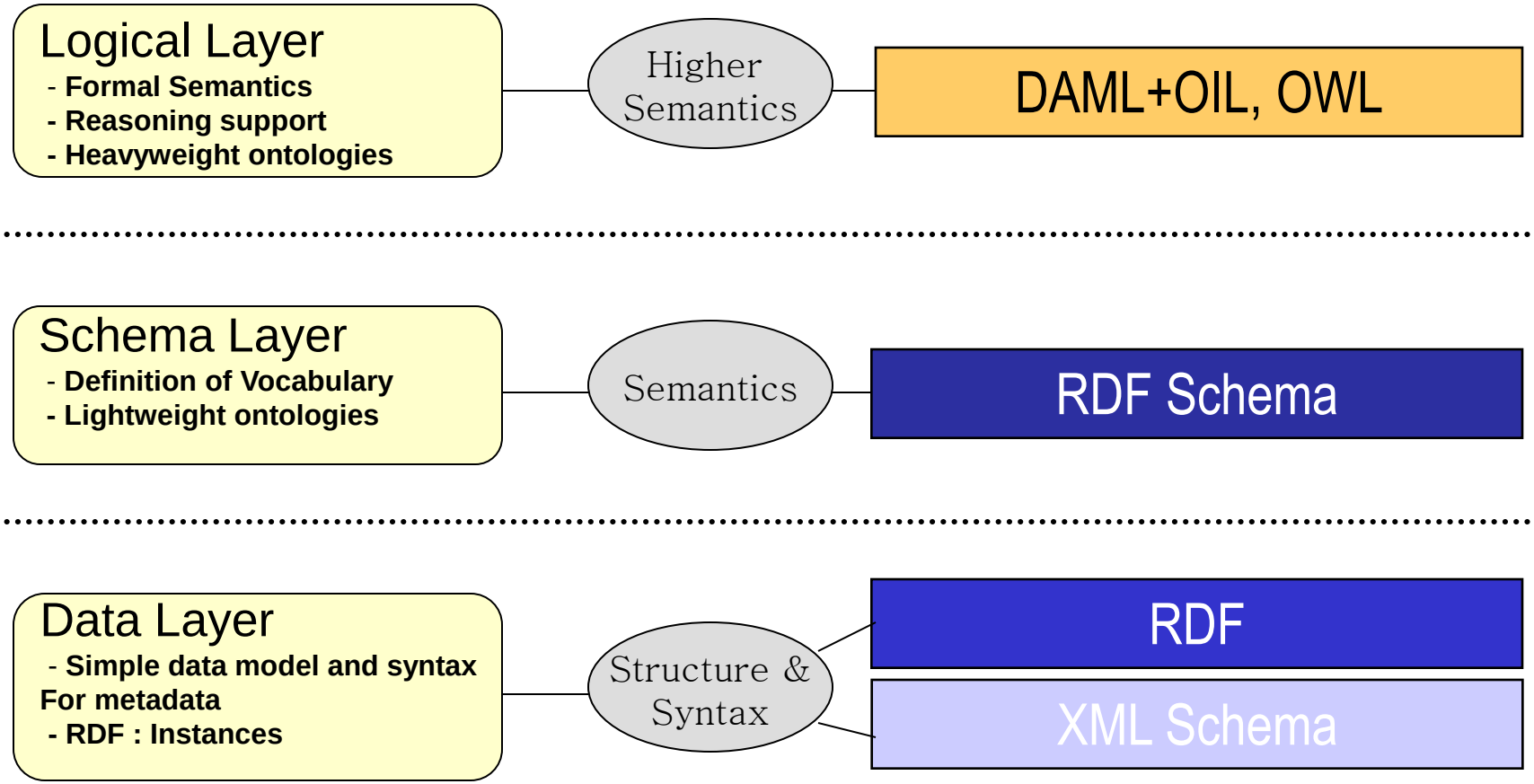


Deborah McGuinness, NSF/NCAR October 30, 2002

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

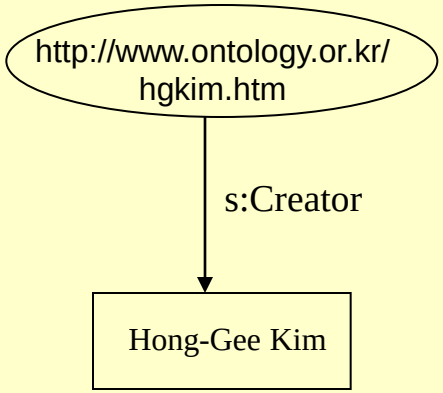
```
<class-def>
        <name>carnivore</name>
        <slot-constraint>
                <name>is-kind-of</name>
                <has-value>animal</has-value>
         </slot-constraint>
</class-def>
```

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

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

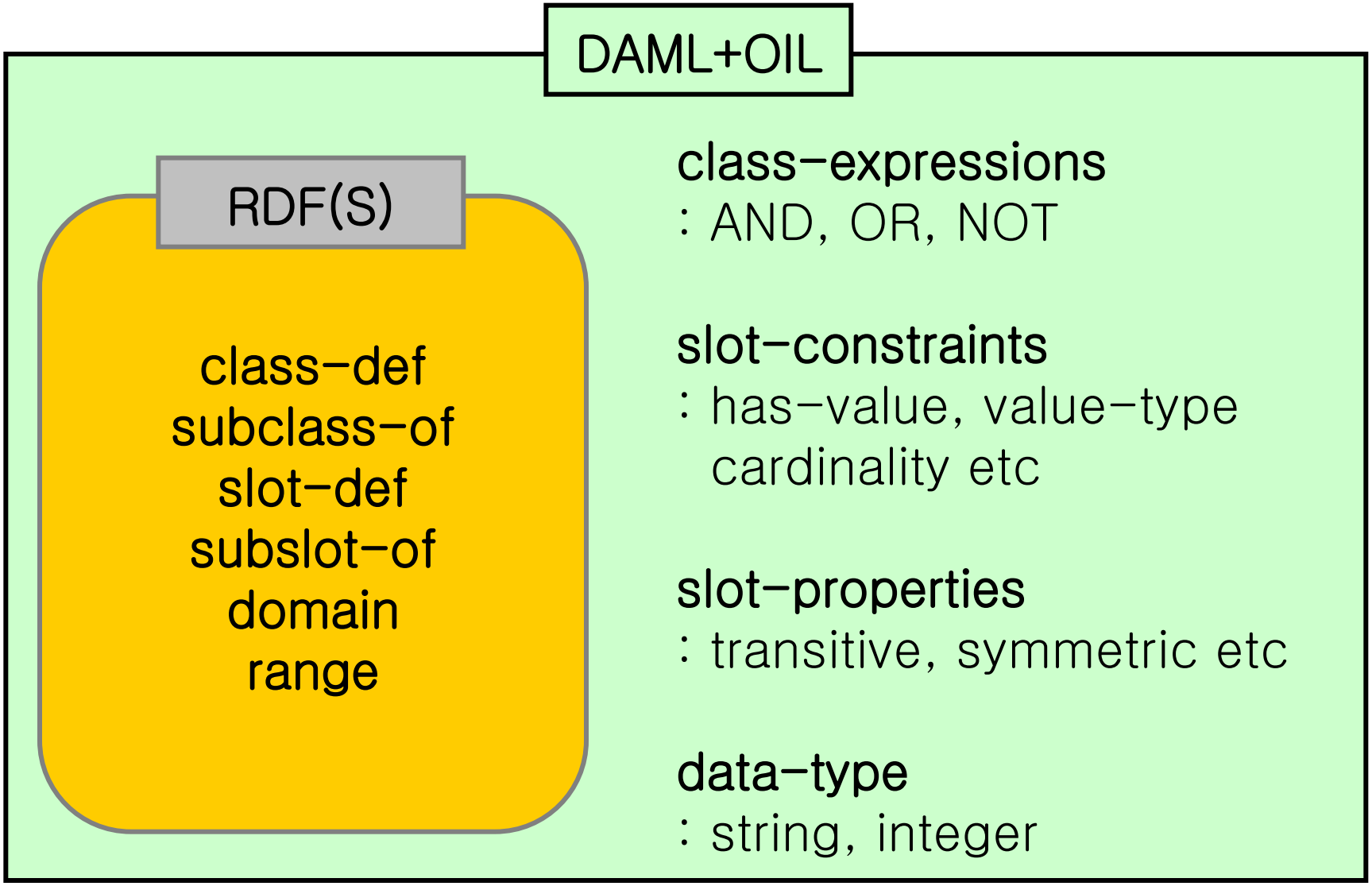
```
<rdfs:Class rdf:about="&mv;Truck">
 <rdfs:subClassOf rdf:resource="&mv;MotorVehicle"/>
</rdfs:Class>
<rdf:Property rdf:about="&mv;registeredTo"
 a:maxCardinality="1">
 <rdfs:domain rdf:resource="&mv;MotorVehicle"/>
 <rdfs:range rdf:resource="&mv;Person"/>
</rdf:Property>
```

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)



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>
```

DAML+ OIL/OWL

- 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:unambigousProperty

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

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-weight한 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 ⊂ DL ⊂ 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 ***differentFom*** Deborah

    : 명시하지않으면 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*
  - *functionalPropety*
  - *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 clasess 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, an d 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>
```

¬ (Meat ∪ Fish)

Classes Description (4)

- WhiteWine ∩ hasSugar.(Dry ∪ 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

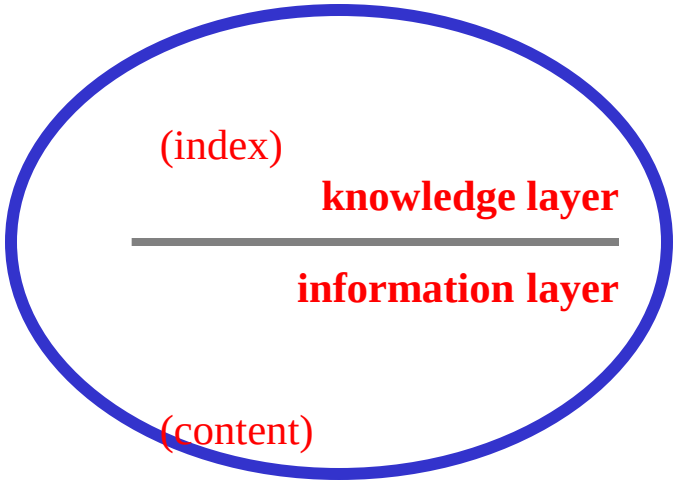
- Ontology Development
  - ⇨ **Consistency check for classes and relations**
  - ⇨ **Consistency check is very important for multi-authored ontologies**
- Ontology Integration
  - ⇨ **Find and assert inter-ontology relationships (for mapping and merging)**
  - ⇨ **Automatically computes integrated class 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**

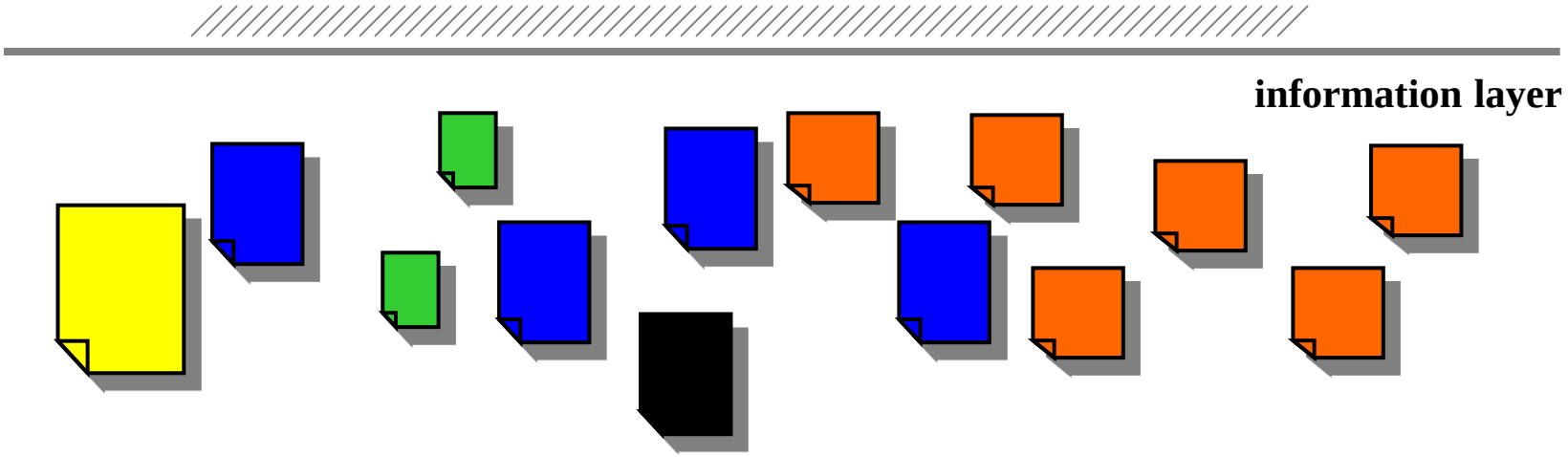
The 2-Layer Topic Map Model

- 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



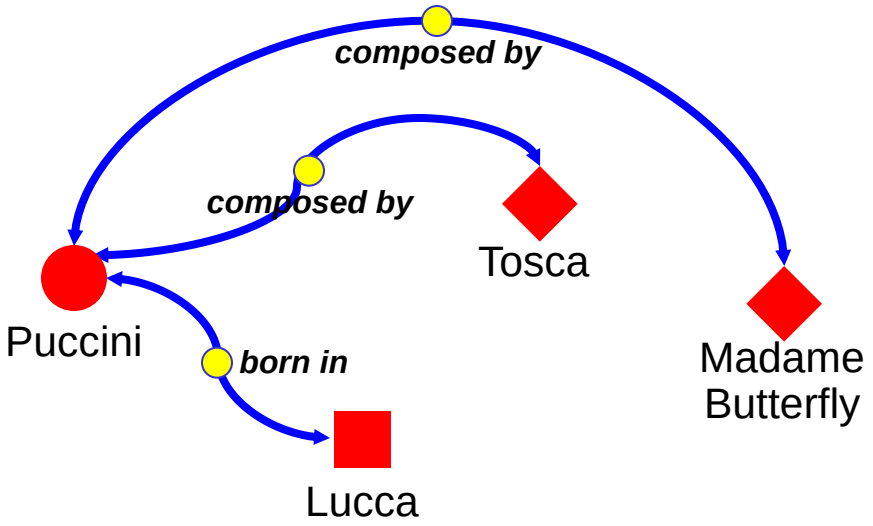
The Information Layer in Topic Map

- 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**

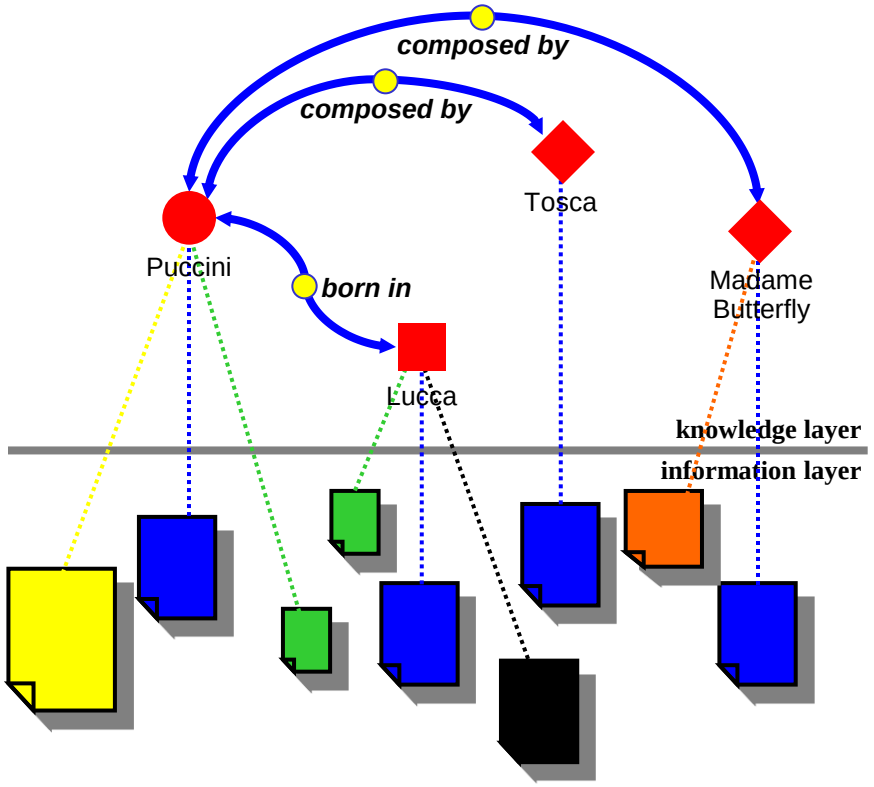


knowledge layer

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

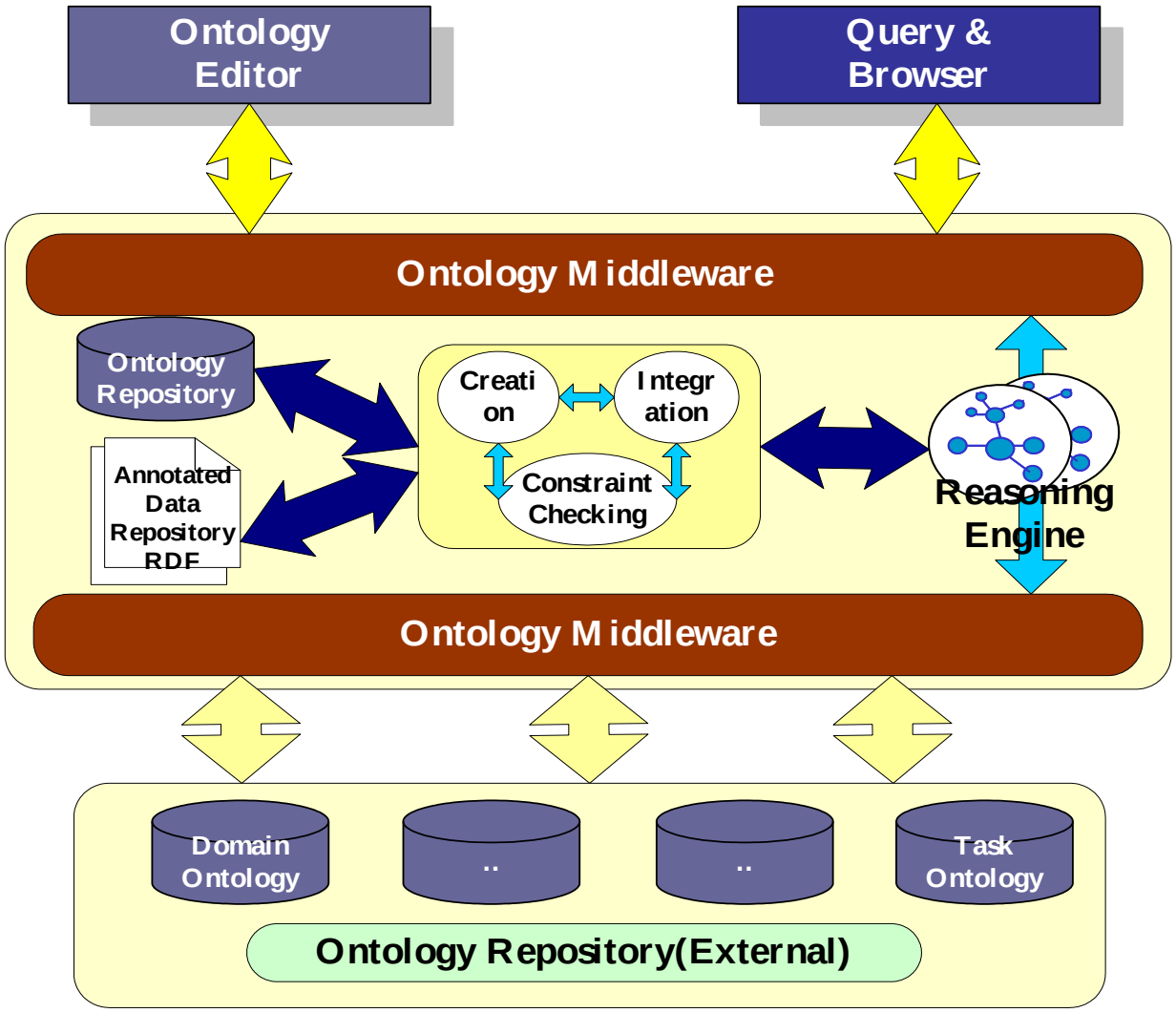
- **= The TAO of Topic Maps**



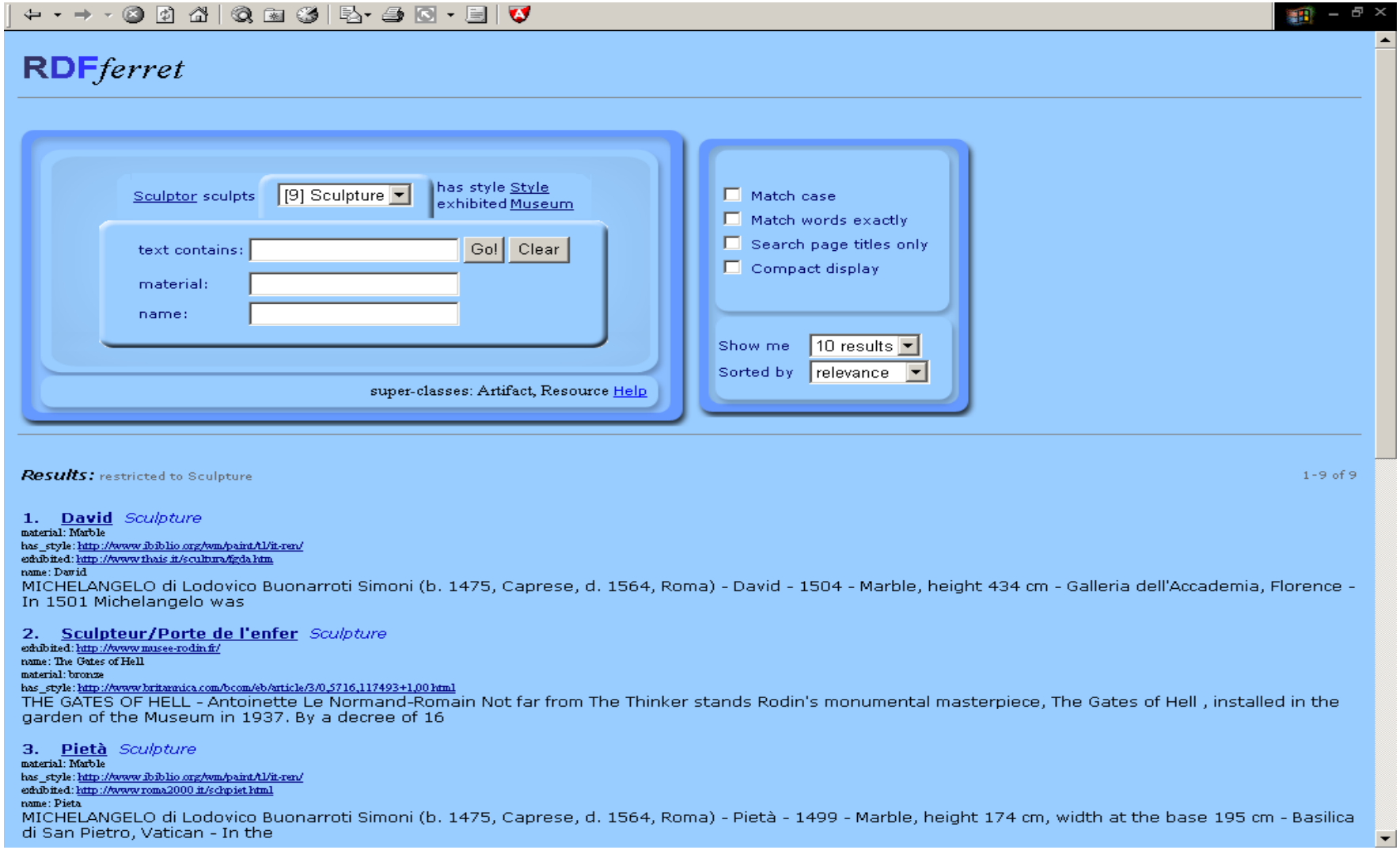
Agenda

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

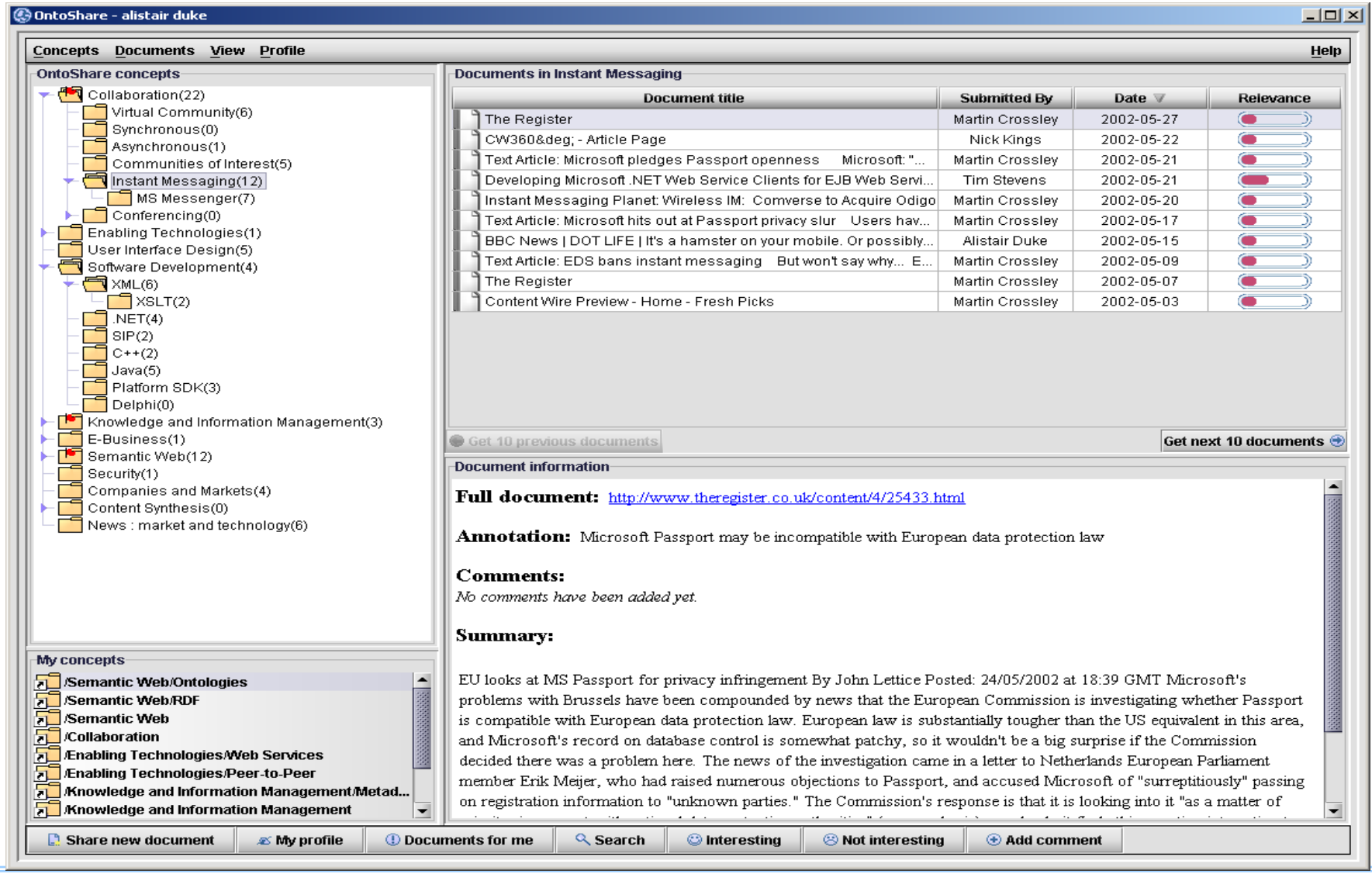
Ontology Tool Architecture



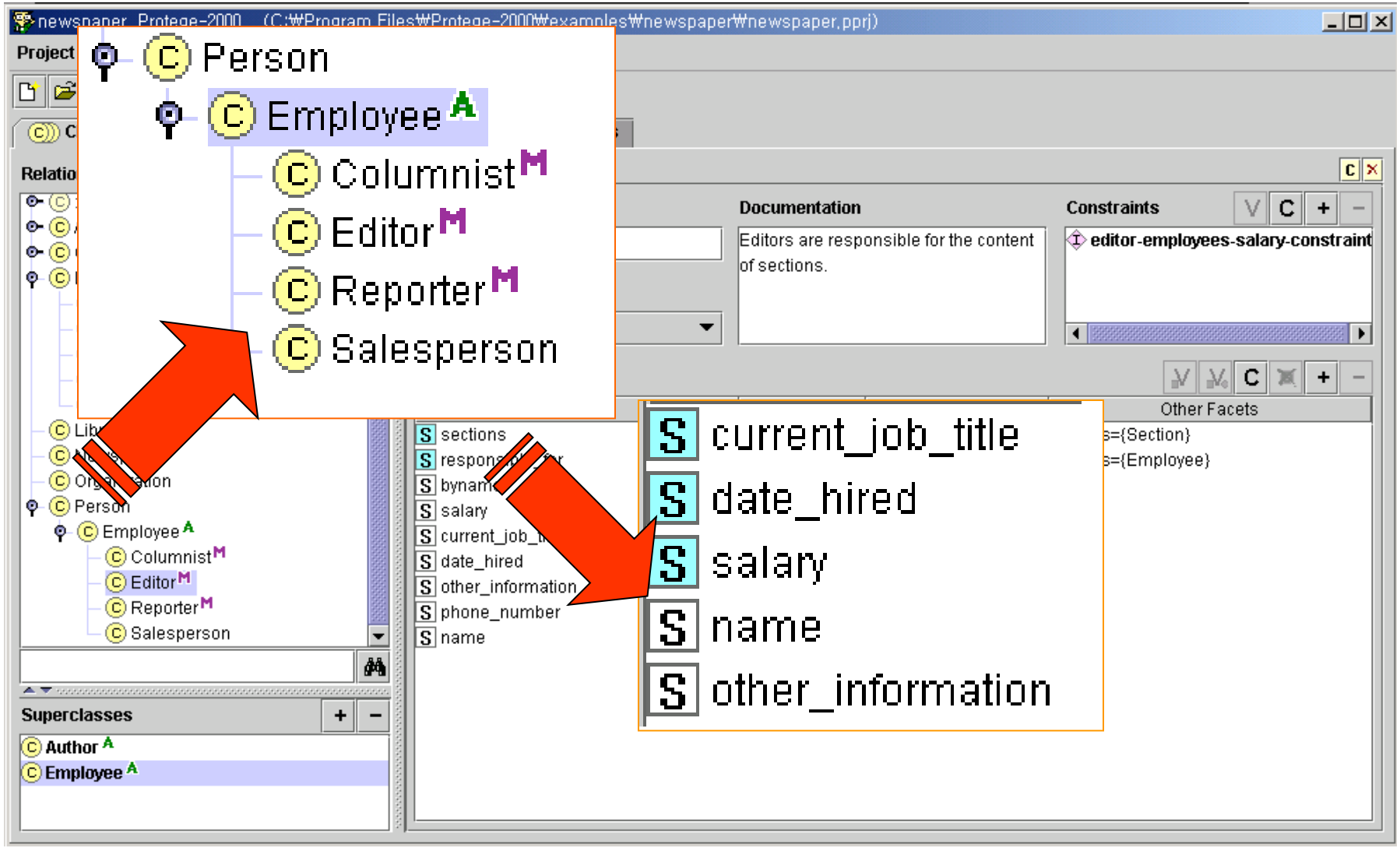
Query and Brower : RDF*ferret*



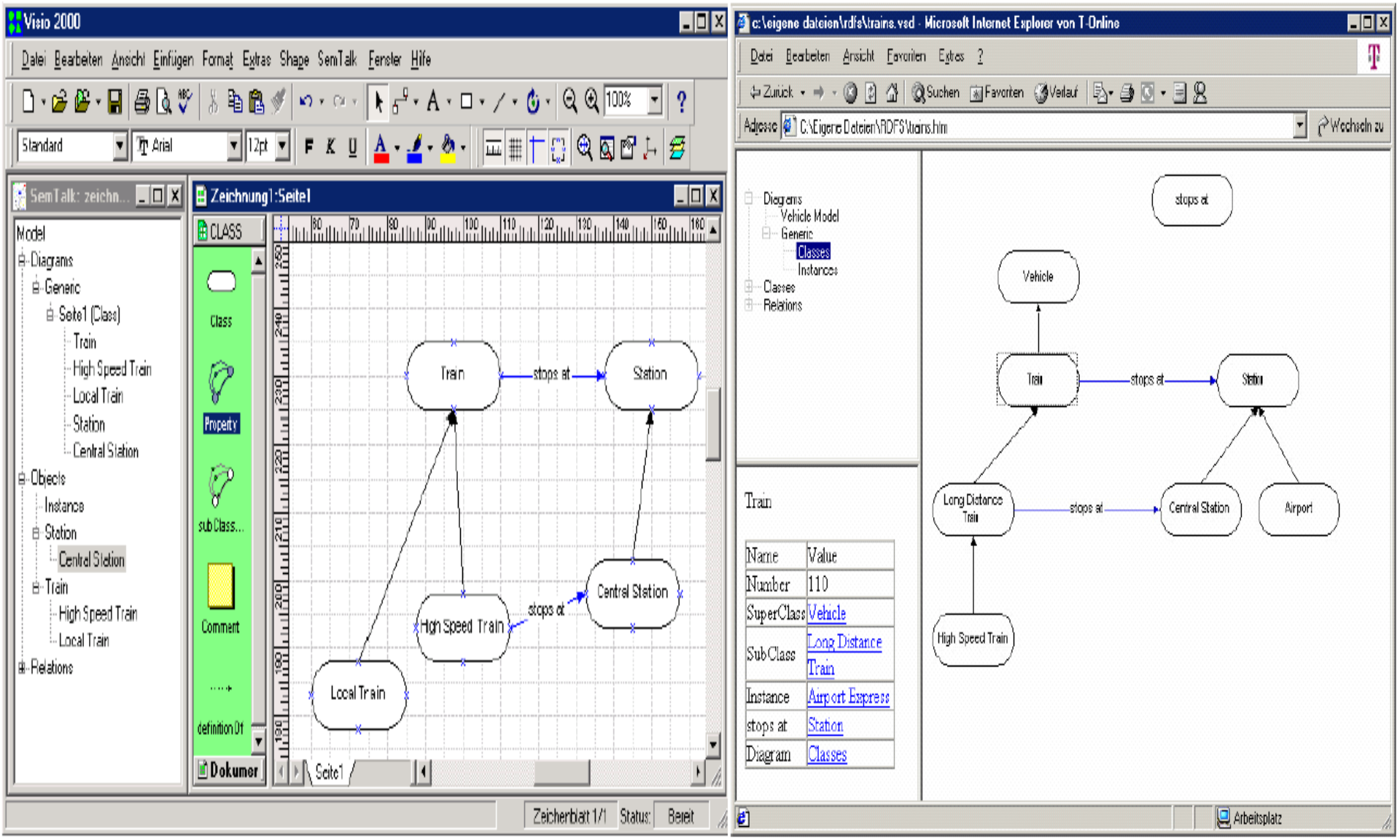
Query and Browser : Ontoshare



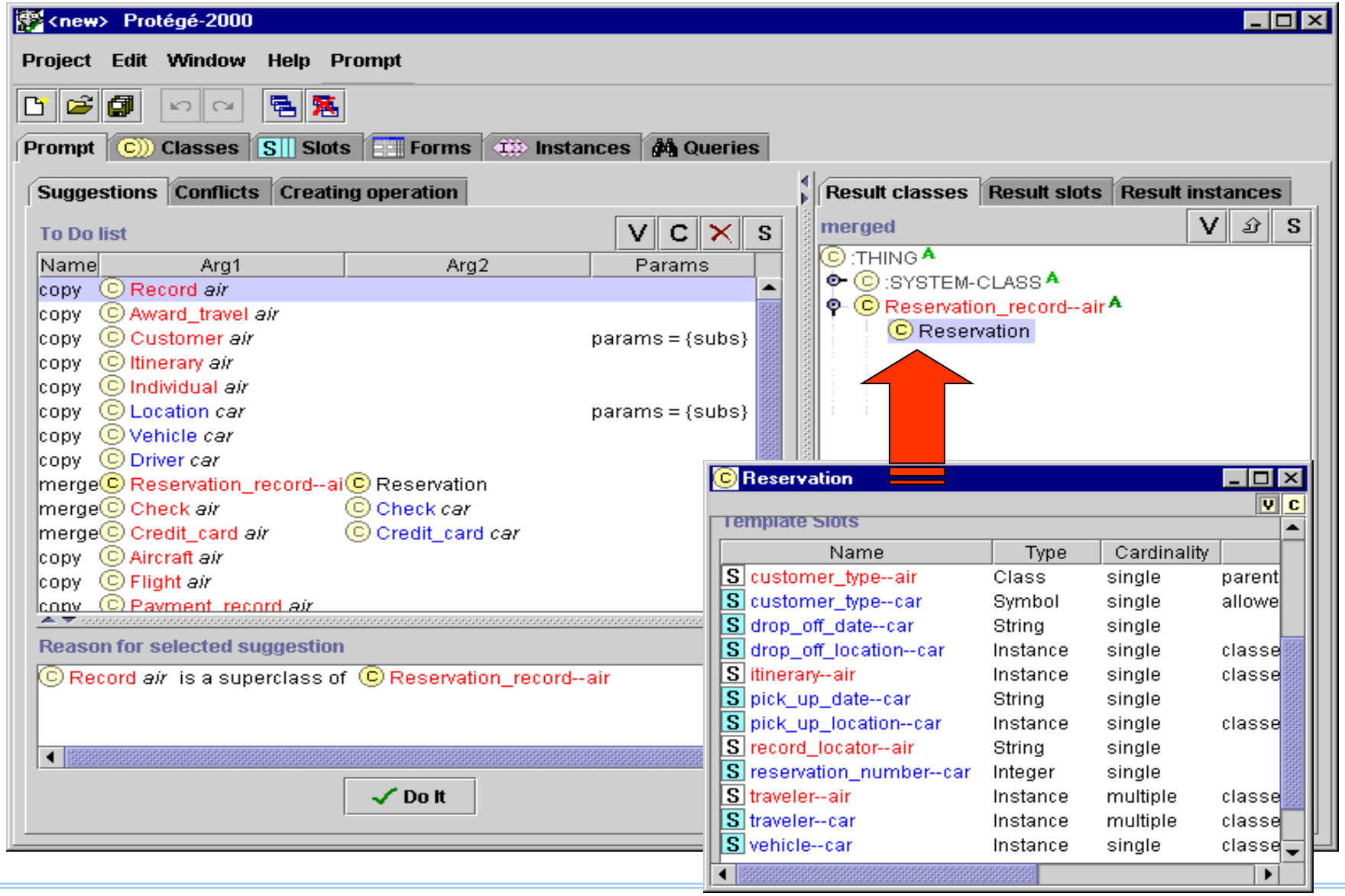
Editor : Protégé-2000



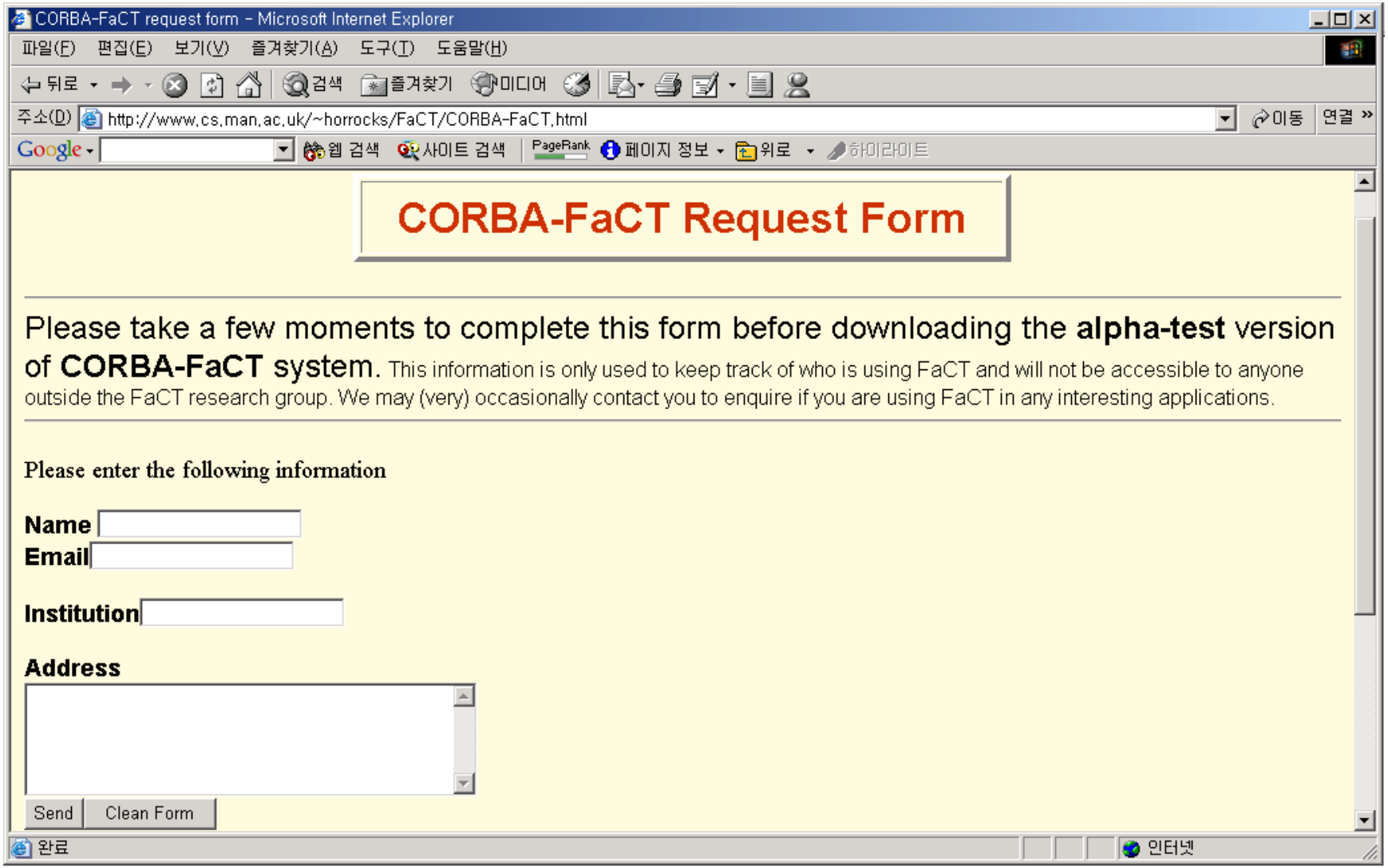
Modeling Tool: SemTalk



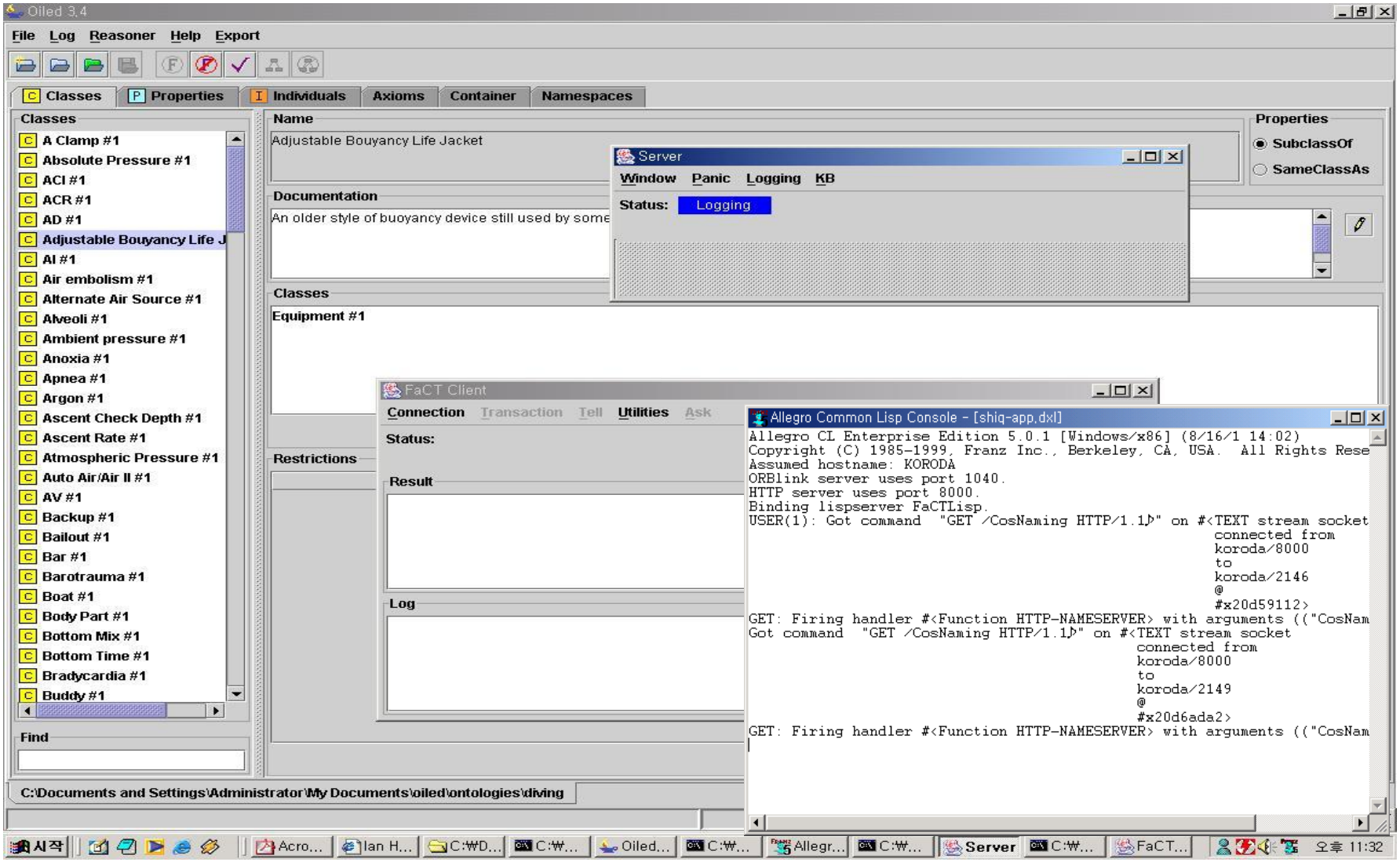
Merge & Integration Tools : PROMPT



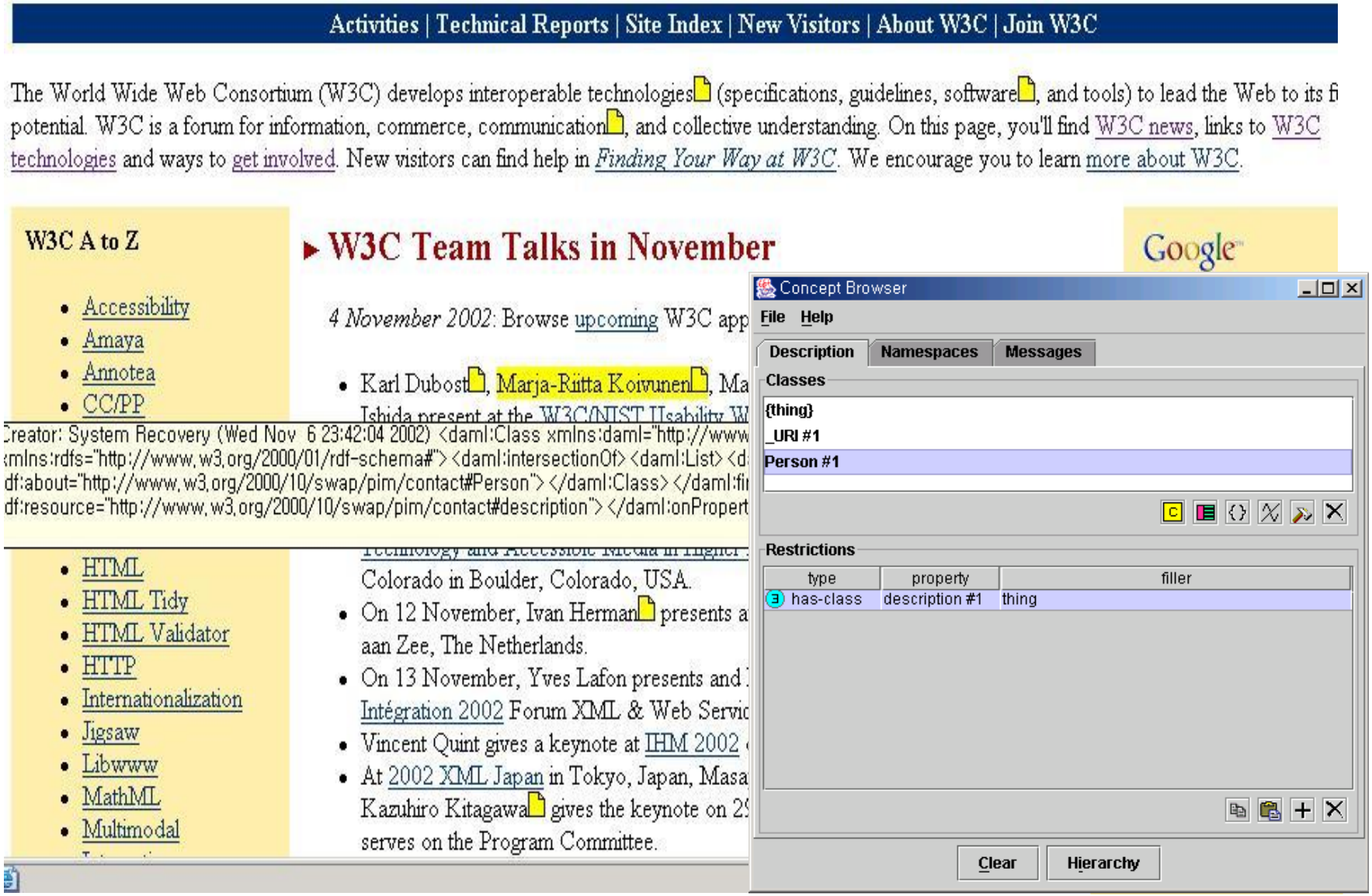
Reasoner : FaCT



Reasoner : OilEd



Annotator：COHSE



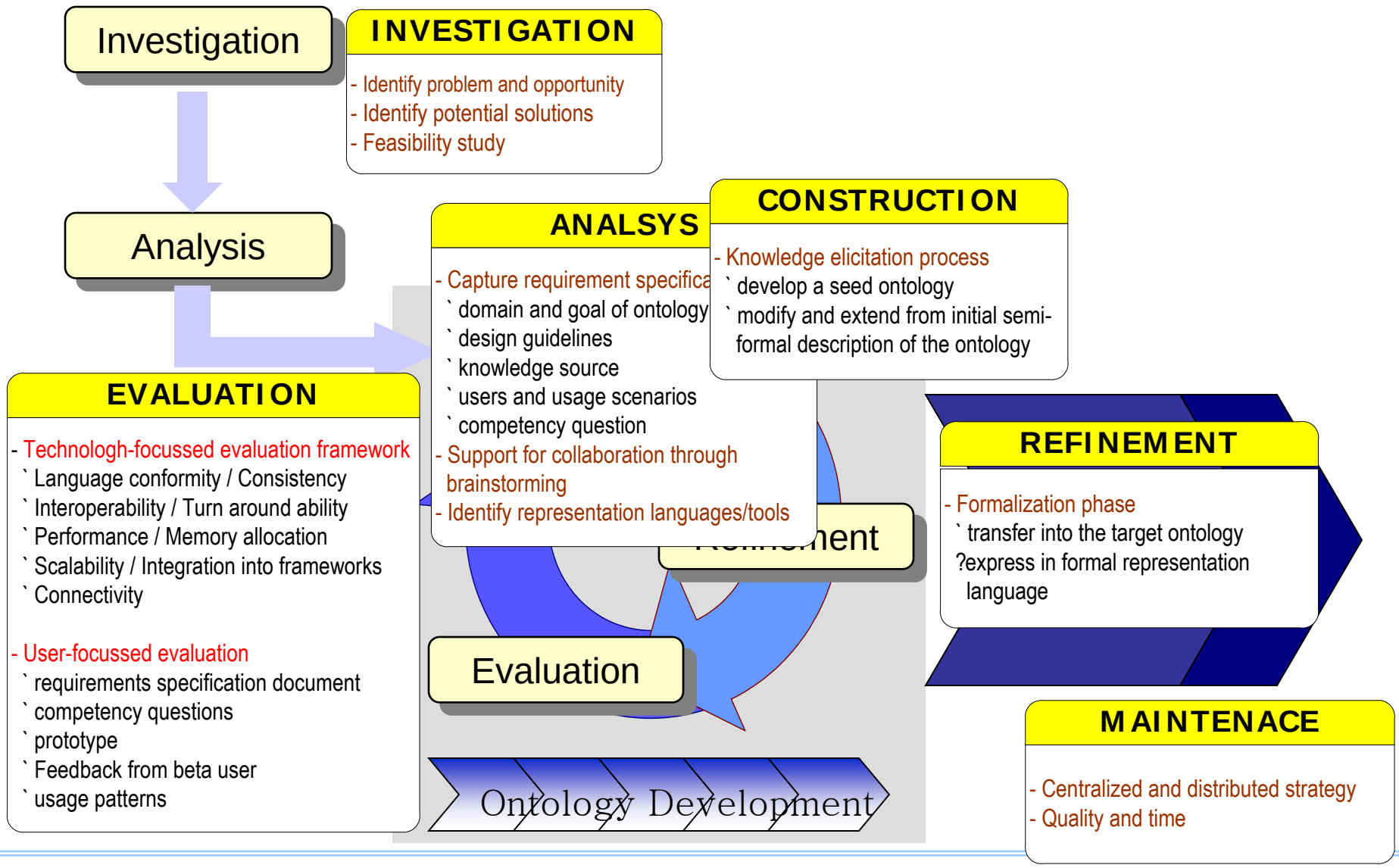
Tools

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+FOL	Frames+FOL	DL(GRAIL)	Frames+FOL	Frames+FOL
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**

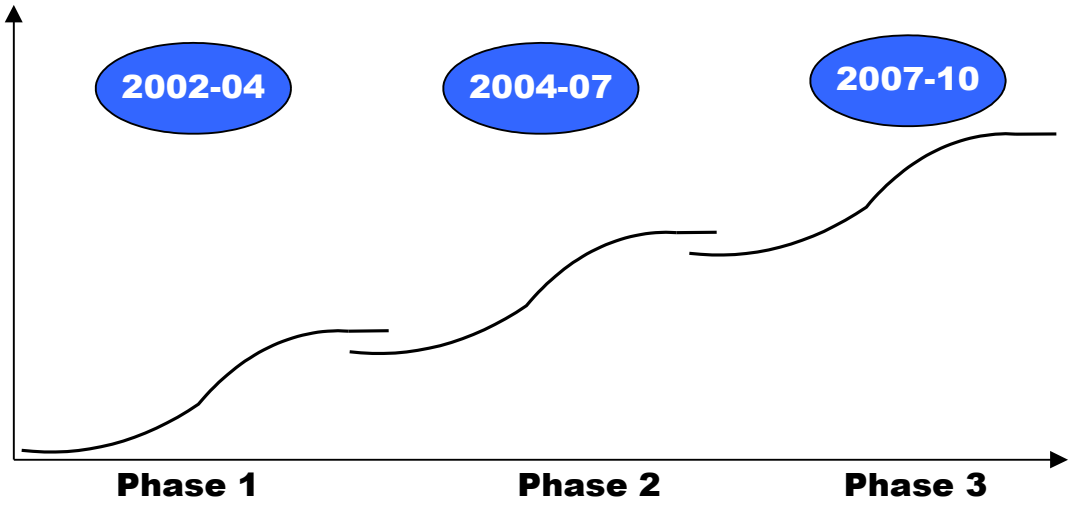
General Use Case

- 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



	Phase 1	Phase 2	Phase 3
시장 채택 단계	기존의 분류체계를 온톨로지화 하고 선도적 사용자들이 조직내의 특수 용도로 온톨로지를 구축하는 초기 단계	시맨틱 웹 기술의 성숙기로서 각 산업별 다양한 온톨로지가 구축되고 온톨로지 사용이 확산되는 단계	애플리케이션/프로세스 통합을 위한 온톨로지 기반 기술이 광범위하게 사용되는 보편화 단계
설명/예	● 과학 – NIH의 게놈 데이터 온톨로지 ● 의학 – SNOMED, GALEN ● ISO 표준 분류체계 ● SIC 산업분류체계 ● 경량급 온톨로지 – 상품 카달로그 – 사이트 디렉토리	● 온톨로지 저작 및 관리도구 개발자들이 활약 ● EC 애플리케이션에 온톨로지 사용이 보편화 – EIP, CRM, SCM, ERP ● First Horizontal apps – proof and trust 서비스	● RDF, 웹기반 온톨로지 언어 사용이 보편화됨 ● 각 산업별 중량급 온톨로지 개발이 보편화 ● 온톨로지 기반 수평적 애플리케이션 통합 ● 지능형 웹서비스, 시맨틱 애플리케이션, 에이전트 기술, 유비쿼터스 컴퓨팅

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**

Final Words

"Ask not what the Semantic Web can do for you,

ask what you can do for the Semantic Web"

Hans-Georg Stork

Thank You !!!