Towards Semantic Service-Oriented Systems on the Web

~ An Overview ~

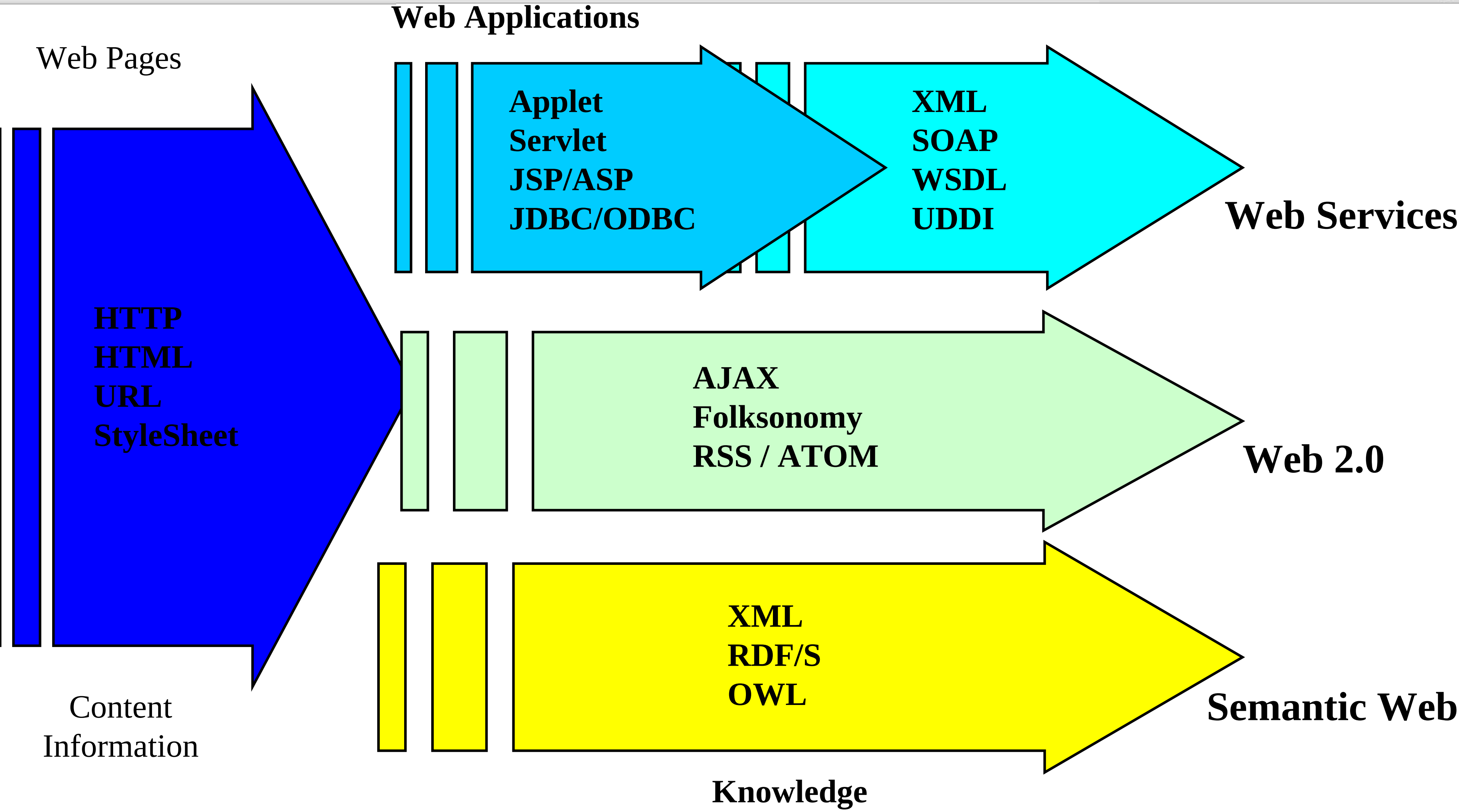
Dumitru Roman

Digital Enterprise Research Institue (DERI)

Dumitru Roman
dumitru.roman@deri.at

SEMAPRO 2007
04.11.2007

Recent trends around the Web



Outline

- Semantic Web
- Web Services
- Semantic Web Services (SWS)
  - Tasks to be Automated
- Existing Approaches to SWS
  - OWL-S, SWSF, IRS-III, WSDL-S
  - The WSMO Approach: WSMO, WSML, WSMX
- Conclusions
- Proposed Challenge for Measuring Success of SWS

Outline

- **Semantic Web**
- Web Services
- Semantic Web Services (SWS)
  - Tasks to be Automated
- Existing Approaches to SWS
  - OWL-S, SWSF, IRS-III, WSDL-S
  - The WSMO Approach: WSMO, WSML, WSMX
- Conclusions
- Proposed Challenge for Measuring Success of SWS

First some definitions from W3C …

- The Semantic Web provides a common framework that allows **data** to be shared and reused across application, enterprise, and community boundaries
  - The Semantic Web is a web of data
- The Semantic Web is about two things
  - It is about common formats for integration and combination of data drawn from diverse sources, where on the original Web mainly concentrated on the interchange of documents
  - It is also about language for recording how the data relates to real world objects

The Web…and it's problems

URI
(how to address data)

http://www.deri.at

What is the Web?

GET /index.html

HTTP
(how to transfer data)

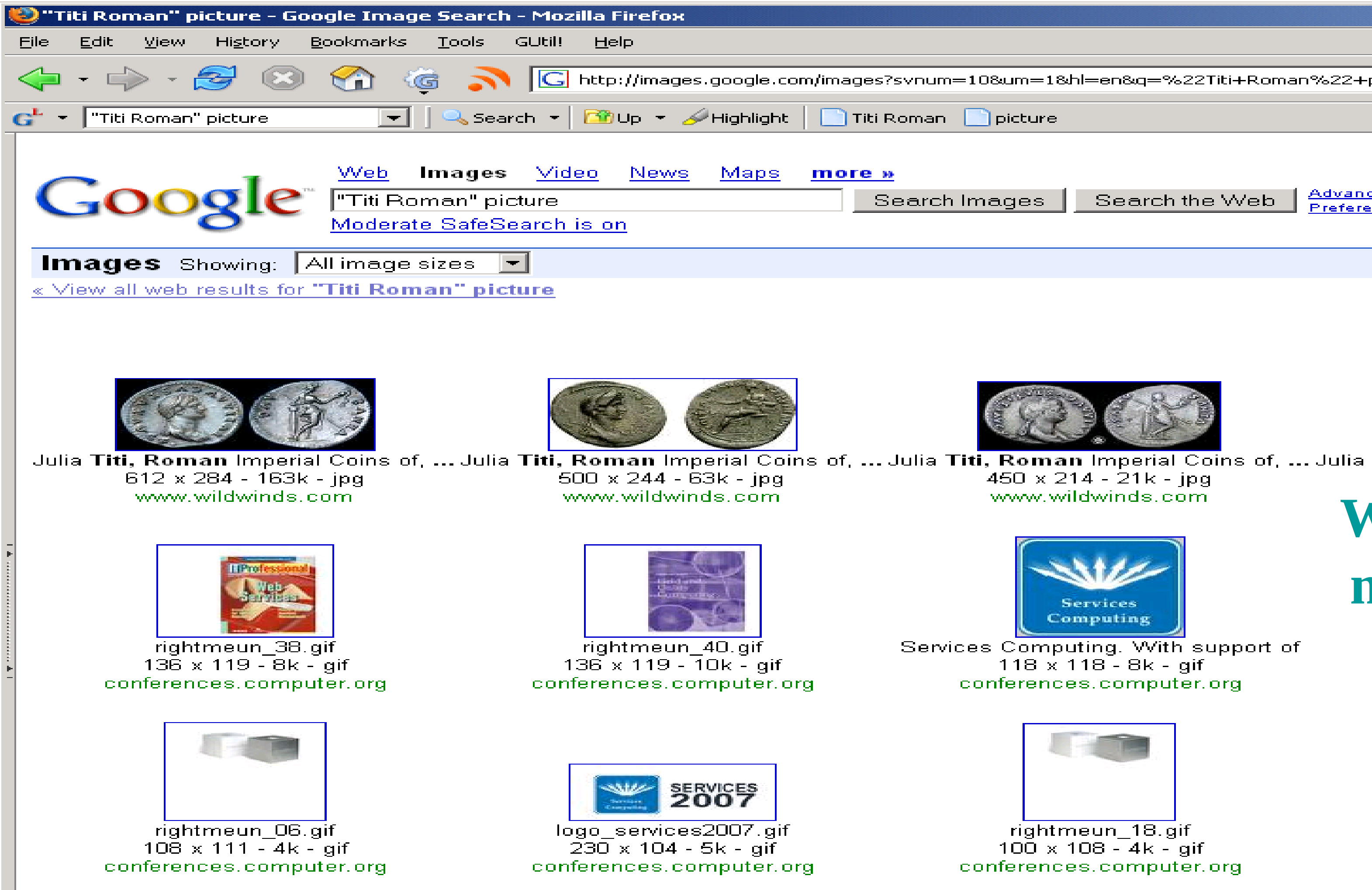
HTML
(how to mark up data for human reader)

<html><head><title>.....

Billions of diverse documents online; problems in:

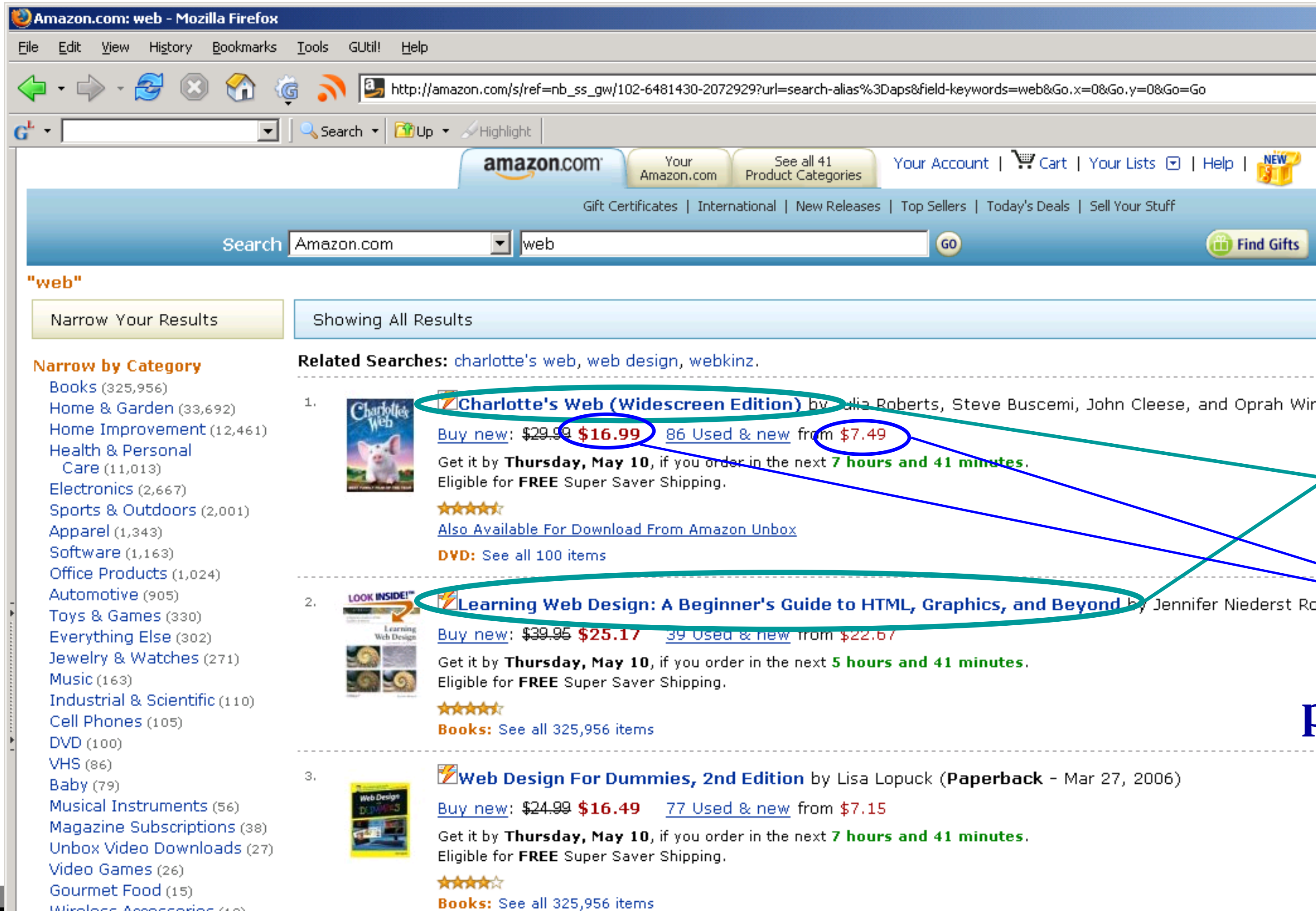
- Retrieving documents
- Extracting relevant data from retrieved documents
- Combining information from different sources to achieve a particular goal

Retrieving documents



Which one is my picture?

Extracting information

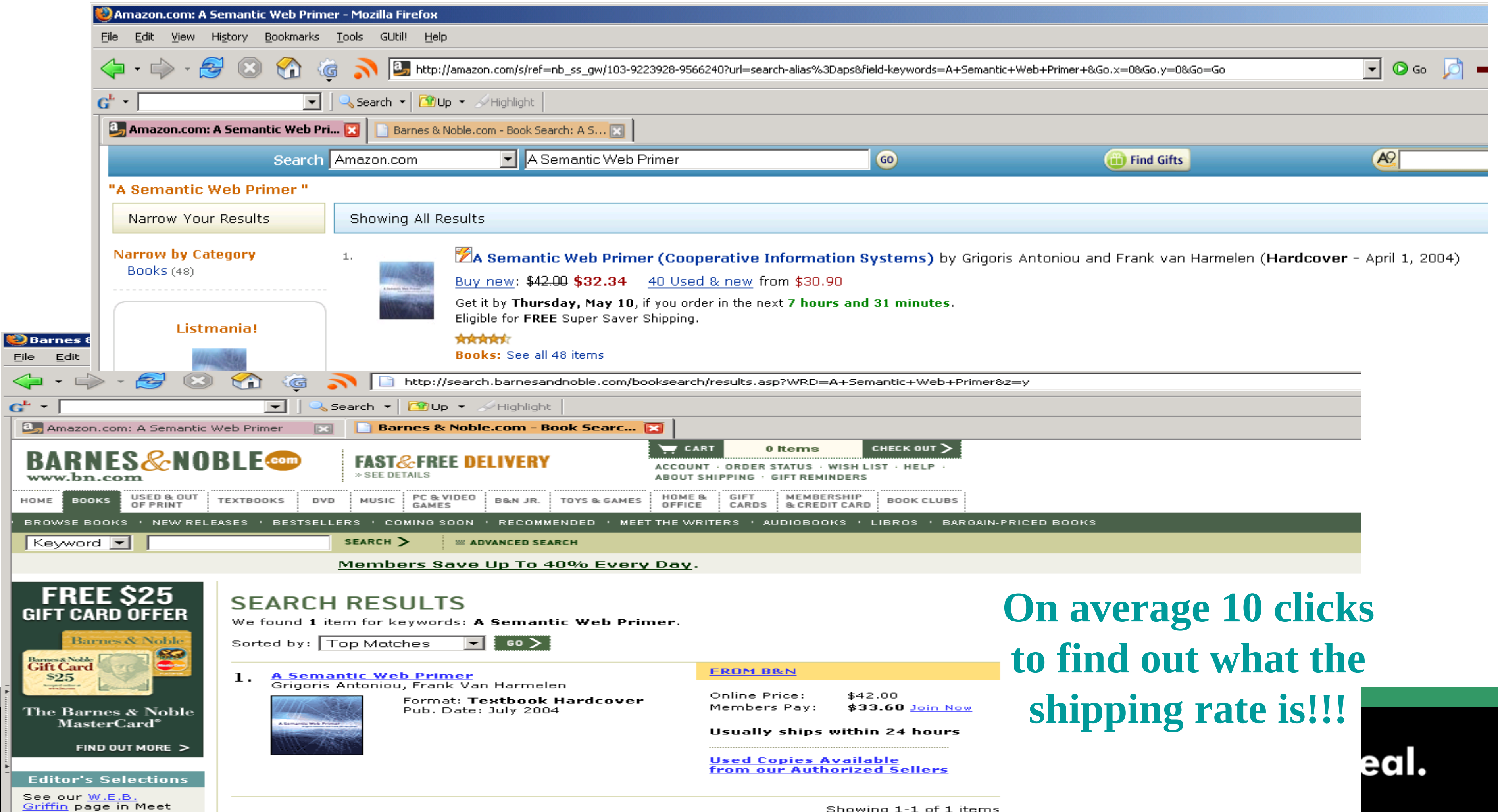


Which book is about the Web?

What is the price of the book?

Combining information

I want the cheapest copy of the book “A Semantic Web Primer”, taking into account the price for shipping the book!



On average 10 clicks to find out what the shipping rate is!!!

The solution!

- Instead of publishing natural language, publish machine-processable data
- Publish information in terms understandable for a machine
- Ask questions in terms understandable for a machine
- And: make sure all machines understand your terms!

=> The Semantic Web!

Publishing and querying machine processable data (cont'd)

- Publishing (*related-to* is transitive):

  B *related-to* A
  C *related-to* A
  D *related-to* C

  Semantic Web Data

  Semantic Web Ontology

  ?x *related-to* ?y and ?y *related-to* ?z => ?x *related-to* ?z

- Querying (give me all things related to A):

  ?x *related-to* A

  Answer:

  ?x = B
  ?x = C
  ?x = D

What is an ontology?

- Formal,
- explicit specification of
- a shared conceptualization of a domain.

- Meaning of ontology is unambiguous
- Avoids misunderstanding
- Specification using formal language
- Enables reasoning: making implicit information explicit
- Hampers consensus

- Make domain assumptions explicit
  - For reasoning
  - For clarifying understanding of domain
- Minimal ontological commitment
  - Too much explicit => no consensus
  - Too little explicit => ontology unusable
  - Minimal ontological commitment = "make as little as explicit as possible, while keeping ontology useful"

- Domain: specific part of the world
- Conceptualization
  - Forming idea of domain in the minds of people
- Shared among its users
  - Facilitates accepting the ontology

Elements of Ontologies

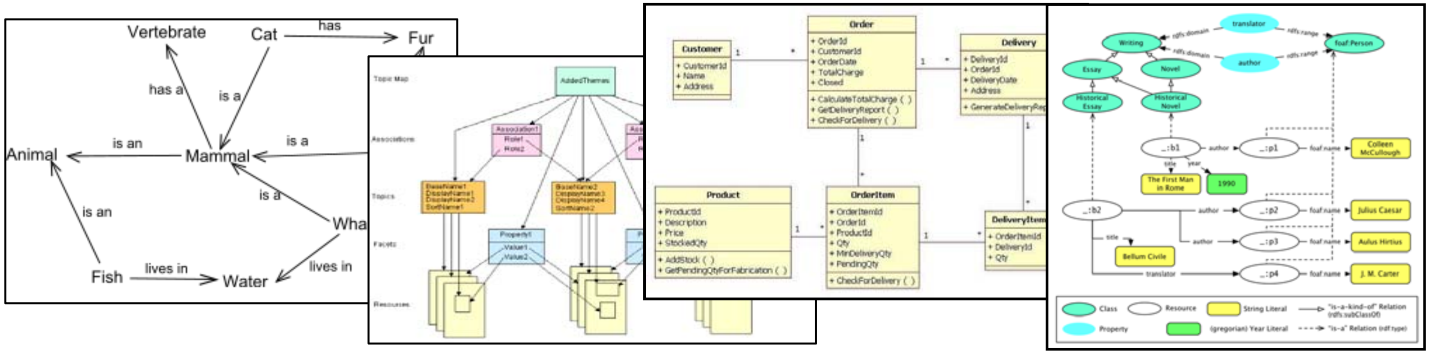
- Classes
  - Grouping of individuals with common properties
  - e.g. Persons, Cars, Universities, …
- Relations
  - Connections between individuals
  - May be attached to classes
  - e.g. hasName, hasAge, owns, …
- Individuals
  - Objects in the domain
  - May be instances of classes
- Axioms
  - Additional statements about the domain
  - Specified in logical language
  - e.g. “hasName has one value”

Ontologies and the Semantic Web

- Form the backbone of the Semantic Web
- Define the basic vocabulary for the annotations
- Enable reasoning with background knowledge, based on formal languages
- Interweave meaning for humans and machines
- Are shared

A wide variety of languages for Ontologies

- **Graphical**: Semantic Networks, Topic Maps, UML, RDF



- **Logical**: Description Logics, First Order Logic, Rules, Conceptual Graphs

DL Syntax	Example
$C_1 \sqcap \ldots \sqcap C_n$	Human ⊓ Male
$C_1 \sqcup \ldots \sqcup C_n$	Doctor ⊔ Lawyer
$\neg C$	¬Male
$\{x_1\} \sqcup \ldots \sqcup \{x_n\}$	{john} ⊔ {mary}
$\forall P.C$	∀hasChild.Doctor
$\exists P.C$	∃hasChild.Lawyer
$\leqslant nP$	⩽1hasChild
$\geqslant nP$	⩾2hasChild

Brothers are siblings

$\forall x, y \;\; Brother(x, y) \;\Rightarrow\; Sibling(x, y).$

"Sibling" is symmetric

$\forall x, y \;\; Sibling(x, y) \;\Leftrightarrow\; Sibling(y, x).$

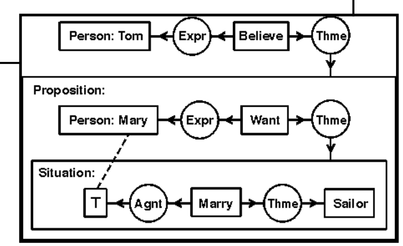
One's mother is one's female parent

$\forall x, y \;\; Mother(x, y) \;\Leftrightarrow\; (Female(x) \wedge Parent(x, y)).$

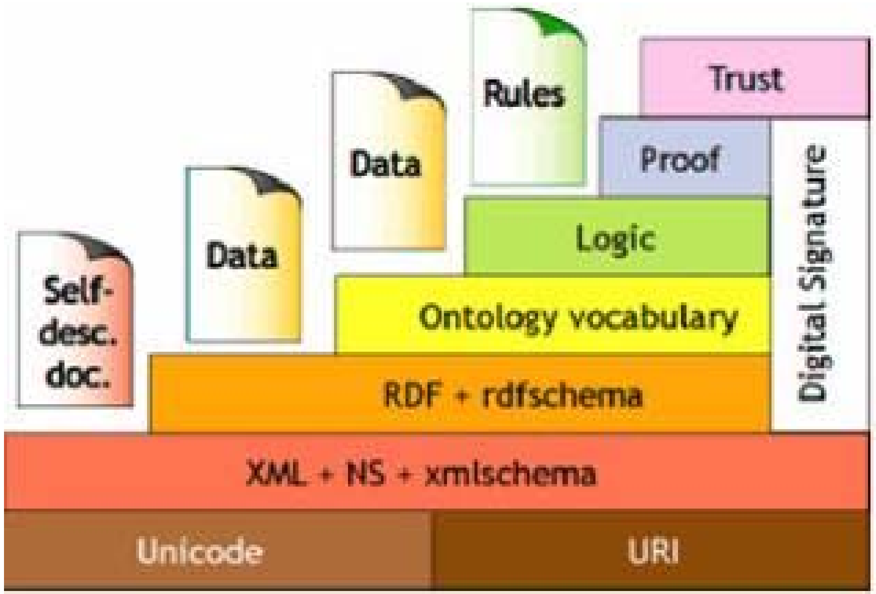
A first cousin is a child of a parent's sibling

$\forall x, y \;\; FirstCousin(x, y) \;\Leftrightarrow\; \exists p, ps \;\; Parent(p, x) \wedge Sibling(ps, p) \wedge Parent(ps, y)$

```
sibling(X, Y)      :- parent_child(Z, X), parent_child(Z, Y).
parent_child(X, Y) :- father_child(X, Y).
parent_child(X, Y) :- mother_child(X, Y).
mother_child(trude, sally).
father_child(tom, sally).
father_child(tom, erica).
father_child(mike, tom).
```

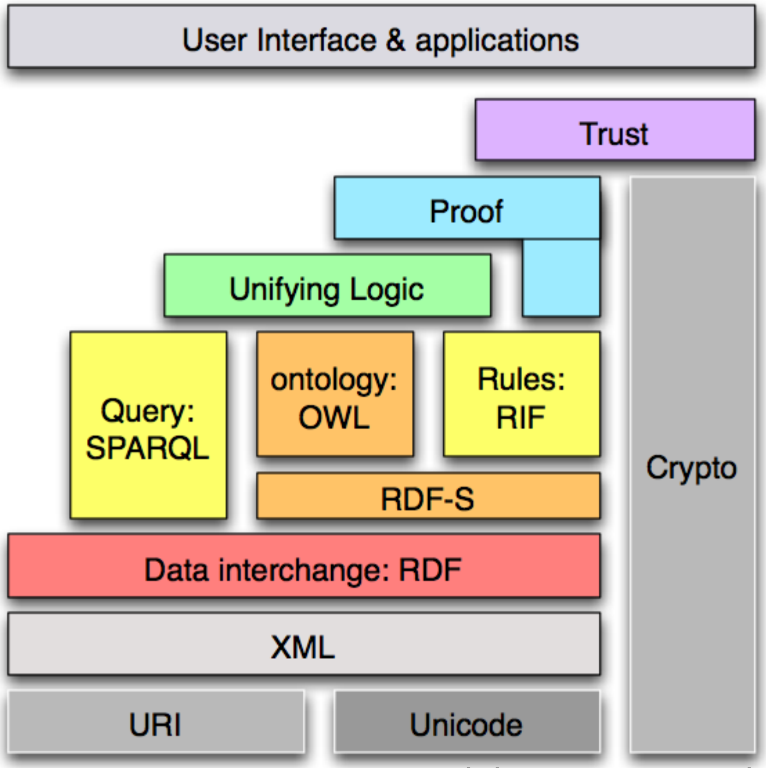


The Evolution of the Semantic Web



(Tim Berners-Lee)

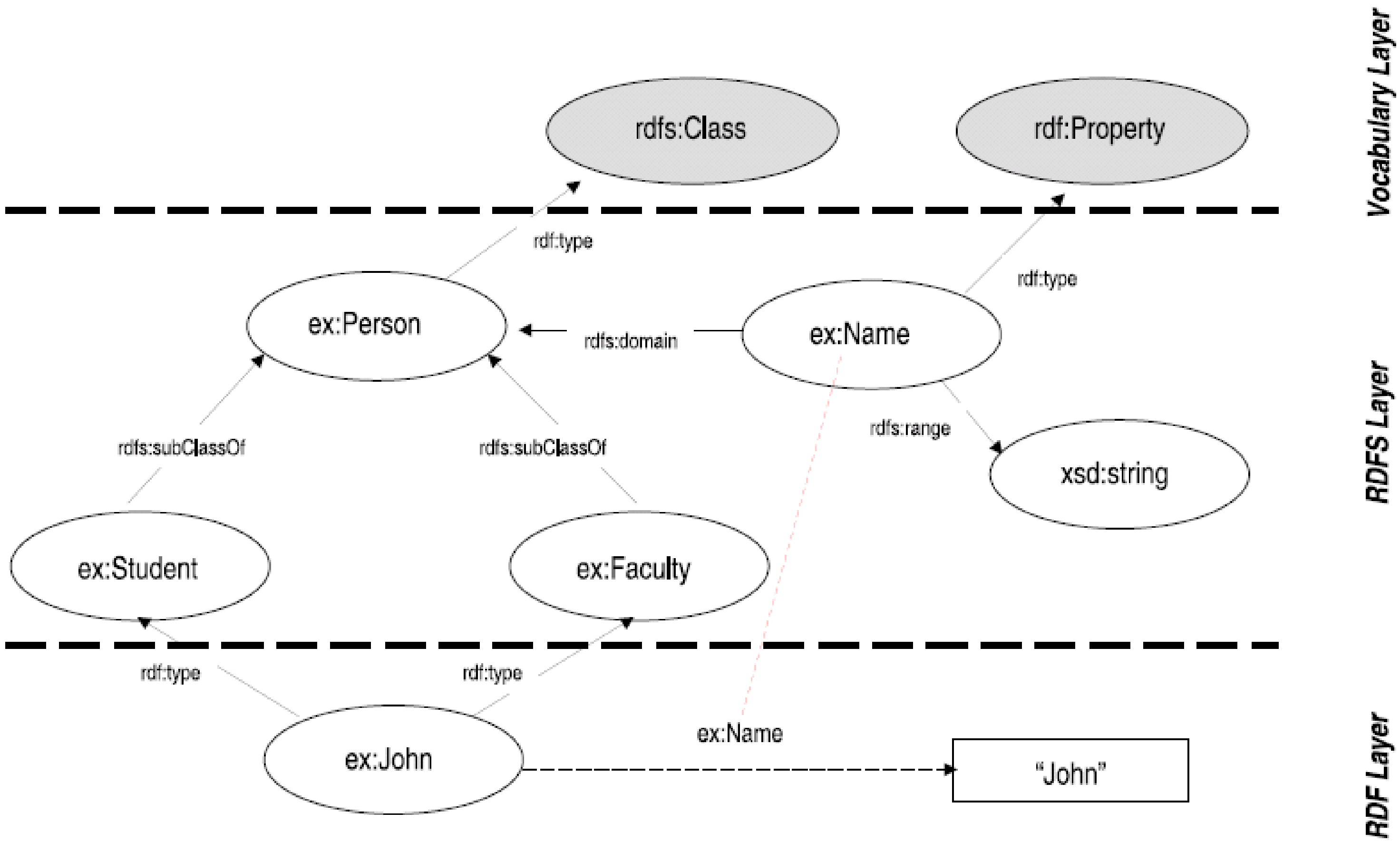
2001



(Tim Berners-Lee)

2006

RDF and RDF(S) – example



RDFS Entitlement - example

<http://example.org/#john> rdf:type <http://example.org/#Student>
<http://example.org/#Student> rdfs:subClassOf <http://example.org/#Person>

entails

<http://example.org/#john> rdf:type <http://example.org/#Person>

<http://example.org/#hasName> rdfs:domain <http://example.org/#Student>
<http://example.org/#mary> <http://example.org/#hasName> “Mary”

entails

<http://example.org/#mary> rdf:type <http://example.org/#Student>

<http://example.org/#john> <http://example.org/#hasMother> <http://example.org/#mary>
<http://example.org/#hasMother> rdfs:subPropertyOf <http://example.org/#hasParent>

entails

<http://example.org/#john> <http://example.org/#hasParent <http://example.org/#mary>

Outline

- Semantic Web
- **Web Services**
- Semantic Web Services (SWS)
  - Tasks to be Automated
- Existing Approaches to SWS
  - OWL-S, SWSF, IRS-III, WSDL-S
  - The WSMO Approach: WSMO, WSML, WSMX
- Conclusions
- Proposed Challenge for Measuring Success of SWS

Web Services – distributed computing background

- Evolution of distributed computing

Downsize
- Client/Server

Components
- CORBA
- DCOM
- EJB

Messaging
- MOM

- Problems with existing models
  - Proprietary protocols
  - Interoperability
  - Platform lock-in
  - Flexibility

SOA
Web Services

Web Services

- W3C: “The World Wide Web is more and more used for application to application communication. The programmatic interfaces made available are referred to as *Web services*”
- A multitude of Web services specifications “WS*-”:

SOAP ASAP WS-Addressing
MTOM WS-Enumeration WS-Eventing
WS-Transfer

WSDL WSRF
WS-Remote Portlet

WS-Reliability WS-ReliableMessaging
WS-Acknowledgment WS-Coordination
WS-Atomic Transaction
WS-BusinessActivity
WS-TransactionManagement

WS-Security : SOAP Message Security
WS-Security: Username Token Profile
WS-Security: X.509 Certificate Token Profile
WS-Security Kerberos Binding
WS-Security Minimalist Profile
WS-SecureConversation WS-Authorization SAML
WS-Security Policy
WS-Encryption WS-Signature
WS-Trust WS-Federation

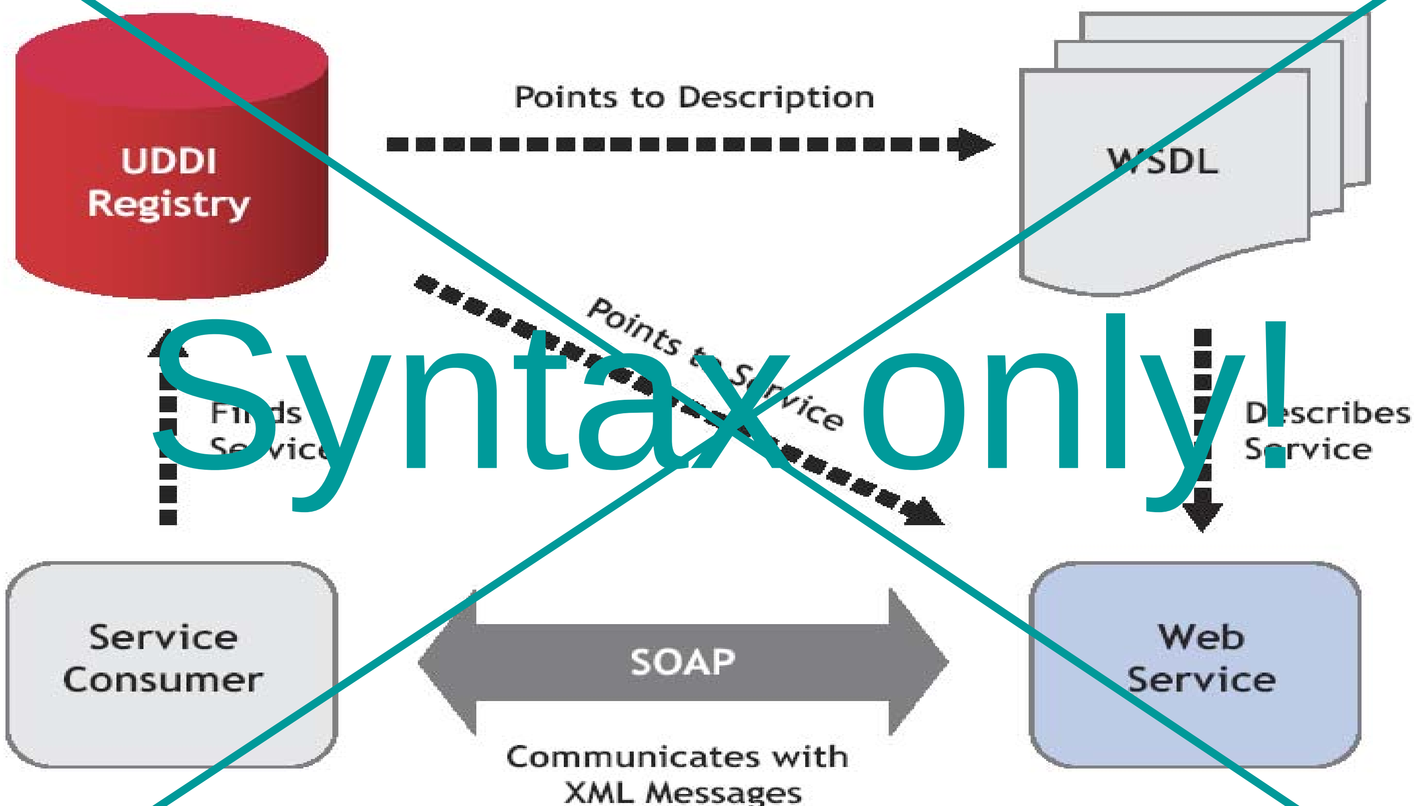
WS-Policy
WS-PolicyAssertions
WS-PolicyAttachment
WS-MetadataExchange
WS-MessageData

UDDI WS-Discovery
ebXML Registry
WSIL WS-Notification

WS-BPEL
WS-CDL WSCI

WS-Management WS-Management Catalog
WS-Manageability WS-Provisioning
WS-Distributed Management (WSDM)

Basic Web Services



Syntax only!

Some deficiencies of WS technology

- Current technologies allow usage of Web Services
- But:
  - only syntactical information descriptions
  - syntactic support for discovery, composition and execution

  => *Web Service usability, usage, and integration needs to be inspected manually*
  - no semantically marked up content / services
  - no support for the Semantic Web

=> Current Web Service Technology Stack did not realize the promise of Web Services

Outline

- Semantic Web
- Web Services
- **Semantic Web Services (SWS)**
- Existing Approaches to SWS
  - OWL-S, SWSF, IRS-III, WSDL-S
  - The WSMO Approach: WSMO, WSML, WSMX
- Conclusions
- Proposed Challenge for Measuring Success of SWS

It's all about automation !!!

Bringing the Web to its full potential

Dynamic

Web Services
UDDI, WSDL, SOAP

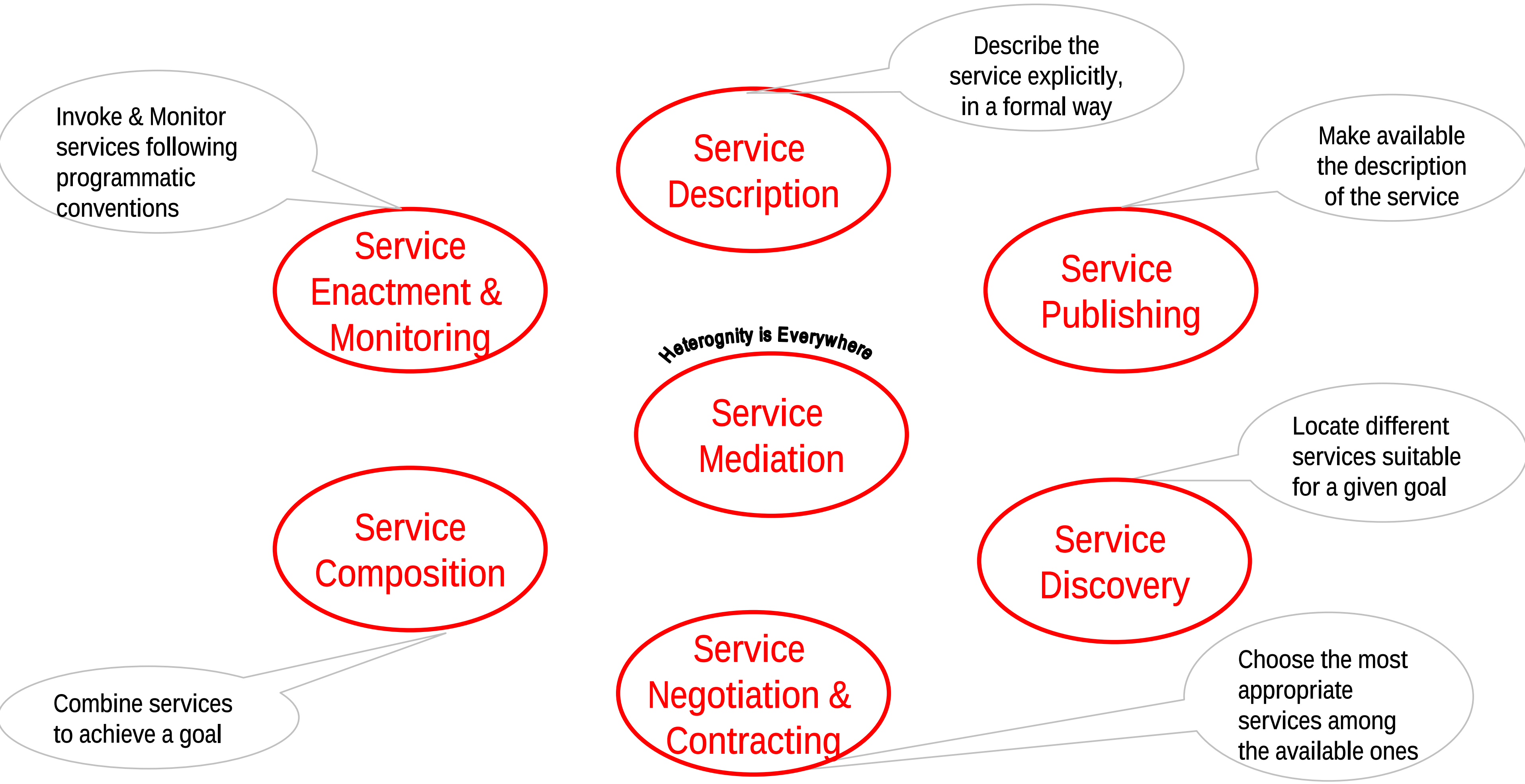
Semantic Web Services

Static

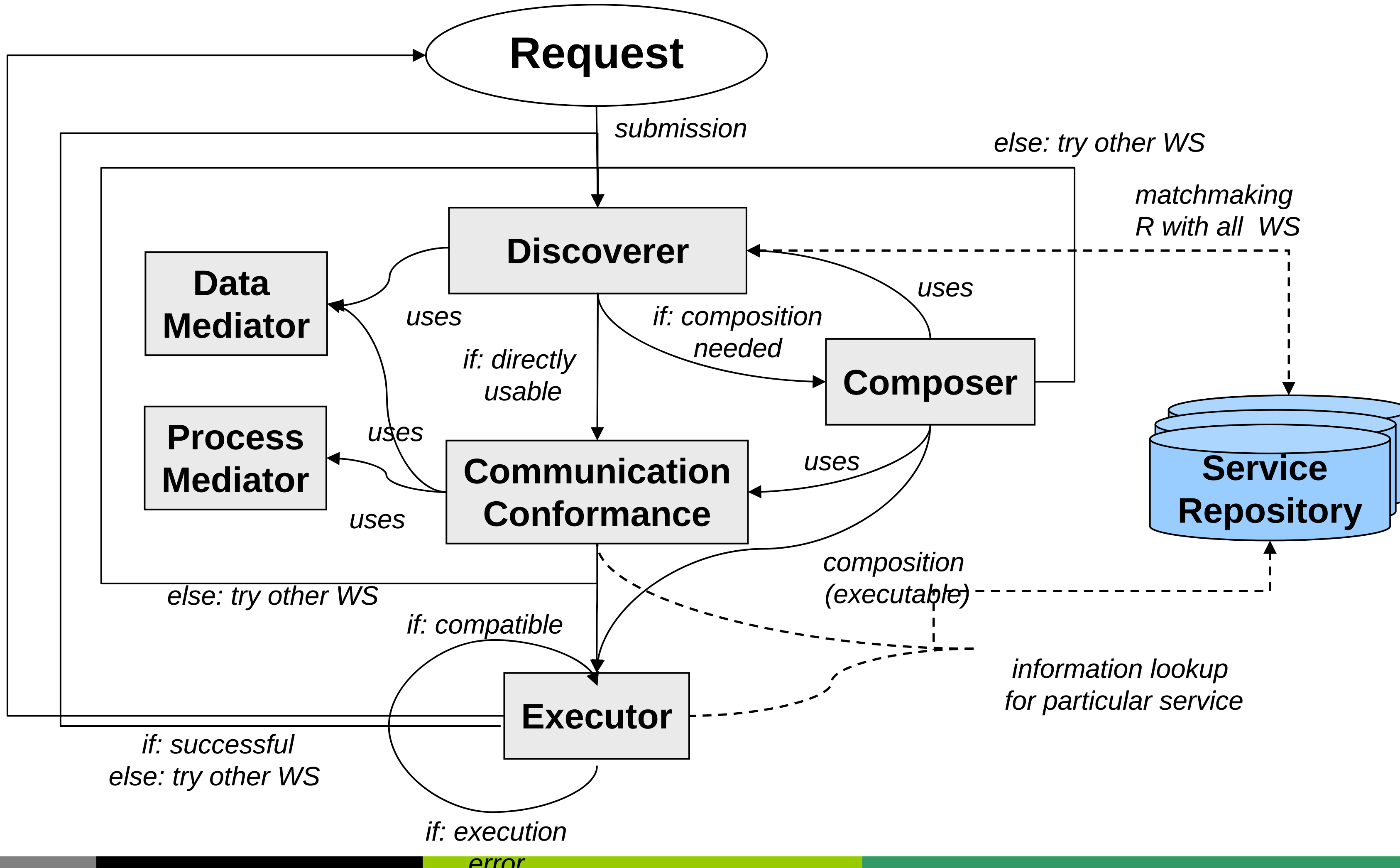
WWW
URI, HTML, HTTP

Semantic Web
RDF, RDF(S), OWL, etc.

SWS – Tasks to be automated



An Example of a SWS Usage Process



So what is needed?

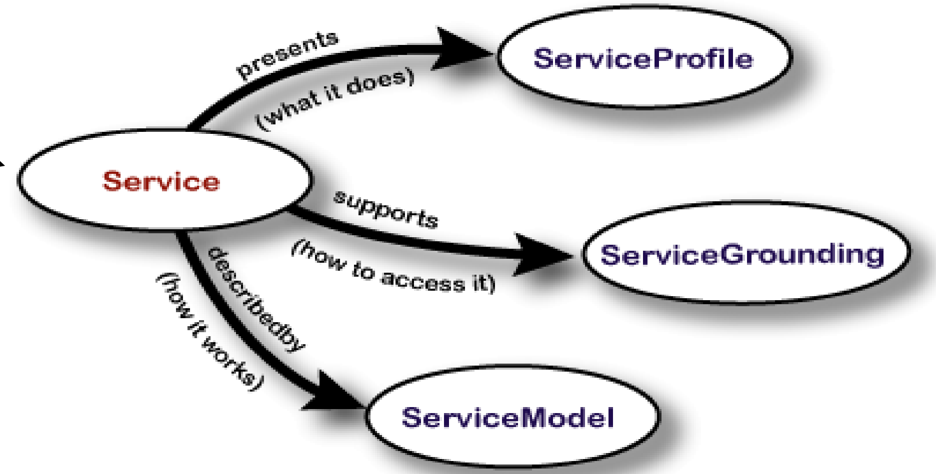
- **Mechanized support** is needed for
  - Annotating/designing services and the date they use
  - Finding and comparing service providers
  - Negotiating and contracting services
  - Composing, enacting, and monitoring services
  - Dealing with numerous and heterogeneous data formats, protocols and processes, i.e. mediation

=> Conceptual Models, Formal Languages, Execution Environments

Outline

- Semantic Web
- Web Services
- Semantic Web Services (SWS)
  - Tasks to be Automated
- **Existing Approaches to SWS**
  - OWL-S, SWSF, IRS-III, WSDL-S
  - The WSMO Approach: WSMO, WSML, WSMX
- Conclusions
- Proposed Challenge for Measuring Success of SWS

OWL-based Web service ontology (OWL-S)



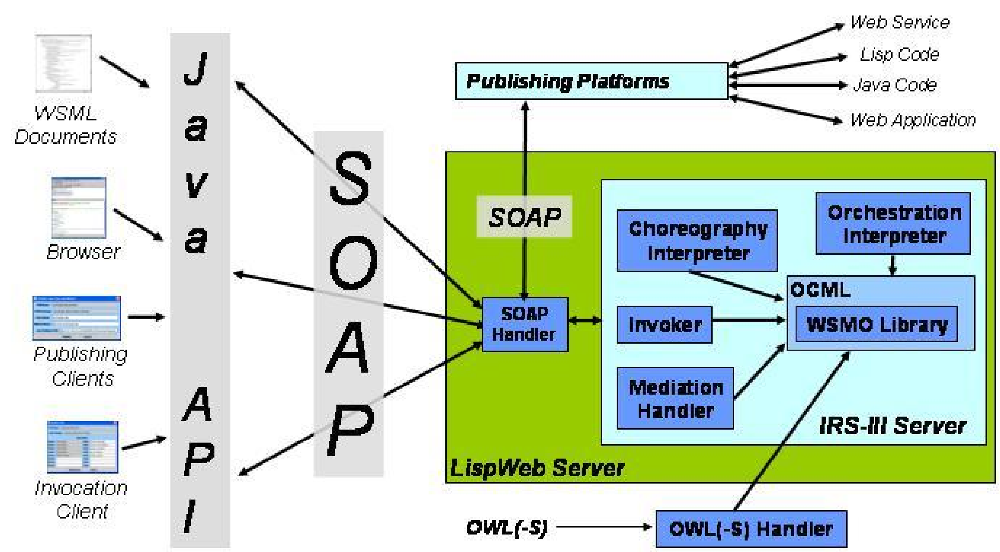
- Conceptual Model
  - A set of ontologies used to describe different aspects SWS
- Language: OWL
- Some OWL-S drawbacks
  - OWL not sufficiently expressive for all aspects of a service
    - more expressive languages have been syntactically integrated: SWRL, KIF, DRS, and PDDL – how do these languages interoperate?
  - Inherits some of the drawbacks of OWL (e.g. lack of proper layering, improper use of OWL for describing and reasoning about processes)
  - No explicit support for Mediation in the language

Semantic Web Services Framework (SWSF)

- Two major components: an ontology and a language used to axiomatize it
- Semantic Web Services Ontology (SWSO) – an extension of OWL-S conceptual model, e.g. a rich behavioural process model based on PSL
  - FLOWS – First-Order Logic Ontology for Web Services
  - ROWS - Rule Ontology for Web Services
- The Semantic Web Services Language (SWSL)
  - SWSL-FOL - based on First Order Logic; includes features from HiLog and F-Logic
  - SWSL-Rules - a logic programming language; includes features from Courteous logic programs, HiLog, and F-Logic
- Some SWSF drawbacks
  - unclear how all the paradigms part of this approach work together
  - first-order logic ontology for Web services, but not a Web language

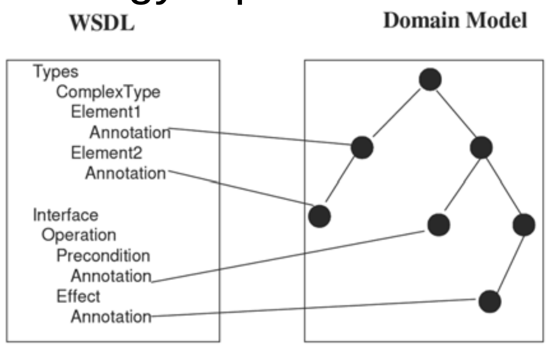
Internet Reasoning Service (IRS-III)

- A platform which acts as a broker mediating between the goals of a user or client and available deployed web services
- Not a SWS framework on its own but uses WSMO as its ontology and follows the WSMO design principles
- IRS Architecture:



Web Service Semantics - WSDL-S

- A mechanism to augment WSDL descriptions with semantics
  - a set of annotations can be created to semantically describe the inputs, outputs and operations of a Web service.
  - keeps the semantic model outside WSDL, making the approach agnostic to any ontology representation language

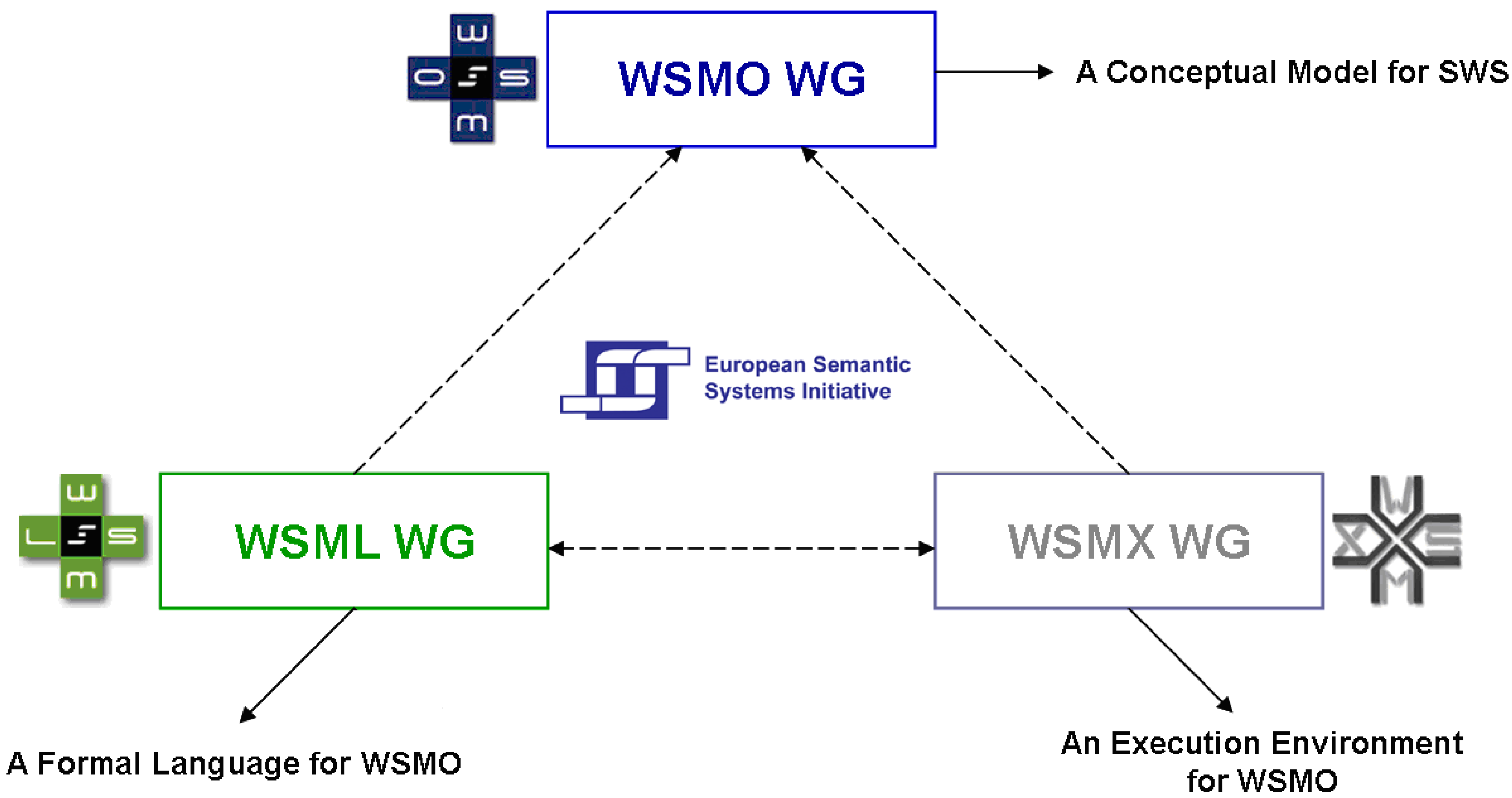


- WSDL-S doesn't provide a conceptual model and language for SWS
  - a bottom up approach to SWS (annotating existing standards with metadata)
- Could be used as a grounding mechanism for SWS

Outline

- Semantic Web
- Web Services
- Semantic Web Services (SWS)
  - Tasks to be Automated
- **Existing Approaches to SWS**
  - OWL-S, SWSF, IRS-III, WSDL-S
  - The WSMO Approach: WSMO, WSML, WSMX
- Conclusions
- Proposed Challenge for Measuring Success of SWS

The WSMO Approach



The Web Service Modeling Ontology (WSMO)

Design Principles

Web Compliance

Ontology-Based

Strict Decoupling

WSMO

Ontological Role Separation

Centrality of Mediation

Execution Semantics

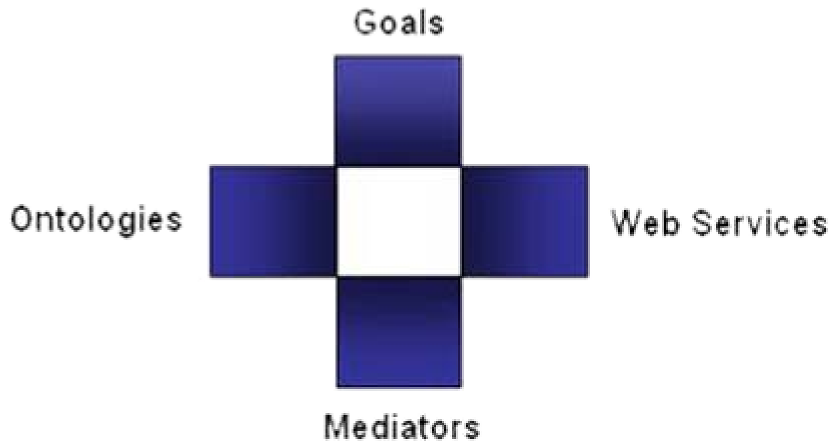
Description versus Implementation

Top-level elements defined by WSMO

(http://www.wsmo.org)

Objectives that a client may have when consulting a Web Service

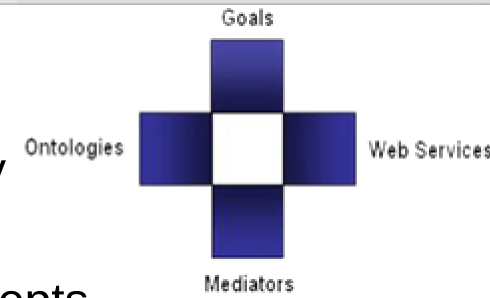
Provide the formally specified terminology of the information used by all other components



Semantic description of Web Services:
- **Capability** *(functional)*
- **Interfaces** *(usage)*

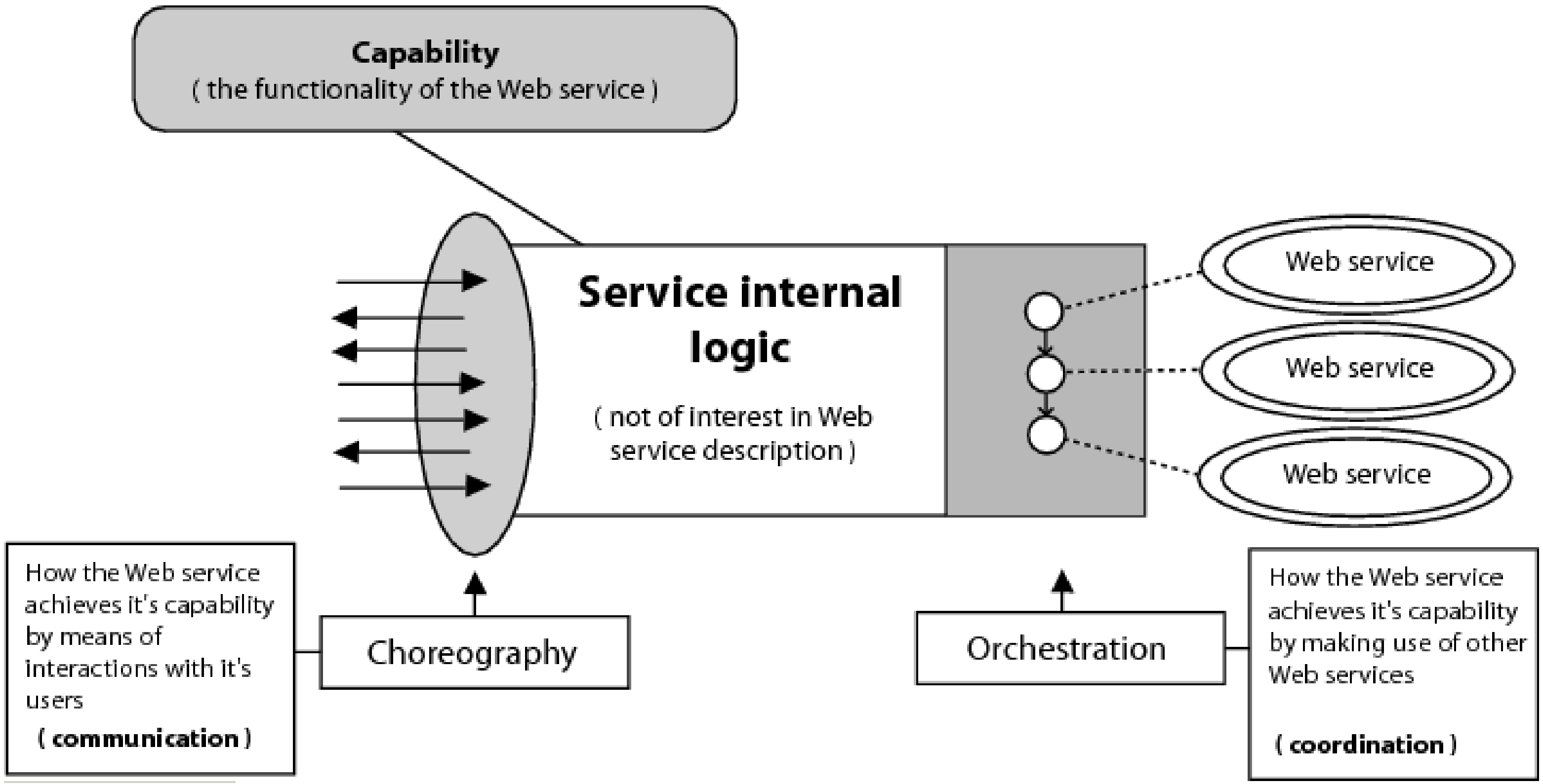
Connectors between components with mediation facilities for handling heterogeneities

WSMO – the Ontology element

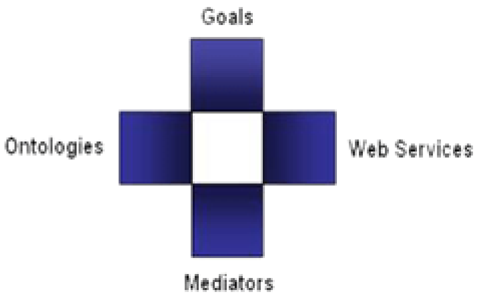


- Ontology elements:
  - **Concepts** - set of concepts that belong to the ontology
    - **Attributes** - set of attributes that belong to a concept
  - **Relations** - define interrelations between several concepts
  - **Instances** - set of instances that belong to the represented ontology
  - **Axioms** - axiomatic expressions in ontology (logical statements)
  - **Non-functional properties**
  - **Imported ontologies** - importing existing ontologies where no heterogeneities arise
  - **Used mediators** - ontology import with terminology mismatch handling
- Ontologies - used as the ‘data model’ throughout WSMO
  - all WSMO element descriptions rely on ontologies
  - all data interchanged in Web Service usage are ontologies
  - Semantic information processing & ontology reasoning

WSMO – the Web service element



The big challenges of defining a WSMO service



- **Capabilities**
  - What is a service able to do?
  - What are the requirements on the input and output?

  ➔ **Preconditions, Assumptions, Postconditions and Effects need to be defined.**

- **Interfaces**
  - How can a service be accessed?
  - How does a service solve its task?

  ➔ **Choreography and Orchestration of services need to be defined.**

WSMO – Goals, Mediators

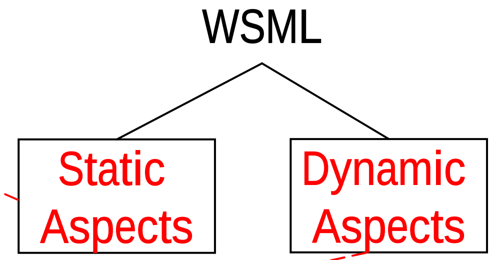
- Goals
  - Defined in a similar way as WSMO Web services
- Mediation
  - Data Level - mediate heterogeneous Data Sources
  - Protocol Level - mediate heterogeneous Communication Patterns
  - Process Level - mediate heterogeneous Business Processes
- WSMO Mediators:
  - *OO Mediators* - terminology import with data level mediation
  - *WW Mediators* - enable interoperability of heterogeneous Web Services
  - *WG Mediators* - link a Web Service to a Goal and resolve occurring mismatches
  - *GG Mediators* – Support specs of goals by reusing exiting goals

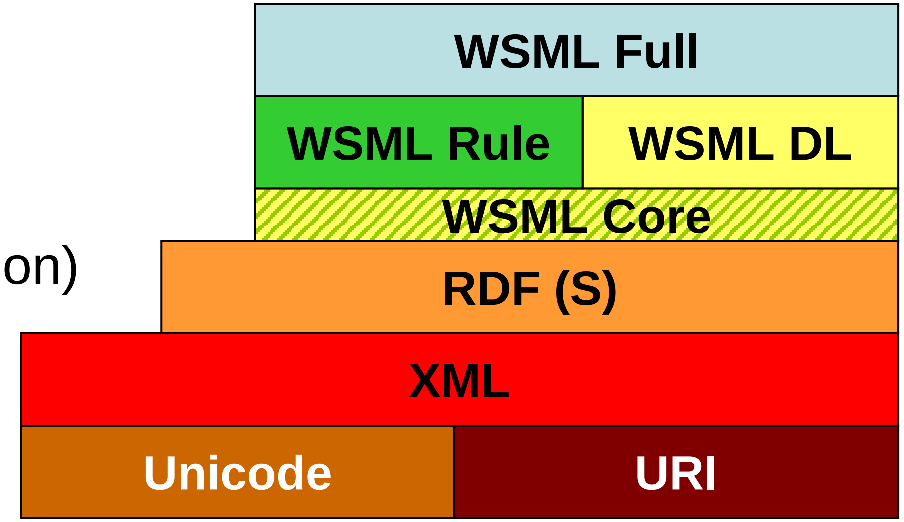
WSMO and the other SWS approaches

	OWL-S	WSMO	SWSF	WSDL-S
Scope	description model for semantically describing Web services	description model & language for core elements of Semantic Web service technologies	extension of the OWL-S Process Model	semantic annotation of WSDL descriptions
Top Level Elements	Service Profile, Process Model, Grounding	Ontologies, Goals, Web Services, Mediators	Processes	Operations / WSDL descriptions
Service Level Description	non-functional aspects IOPE for service-level functional description	capability (PAPE) for provided and requested functionality	not in the scope	keyword classification (ontology-based)
Operation Level Description	IOPE for processes	interfaces for consumption (choreography) and interaction (orchestration)	internal behavior (atomic and composite processes)	preconditions & effects for WSDL operations
Language (static)	OWL	WSML	SWSLFOL & SWSLRules	not specified
Language (dynamic)	Process Model and OWL	Abstract State Machines	FLOWS	not specified

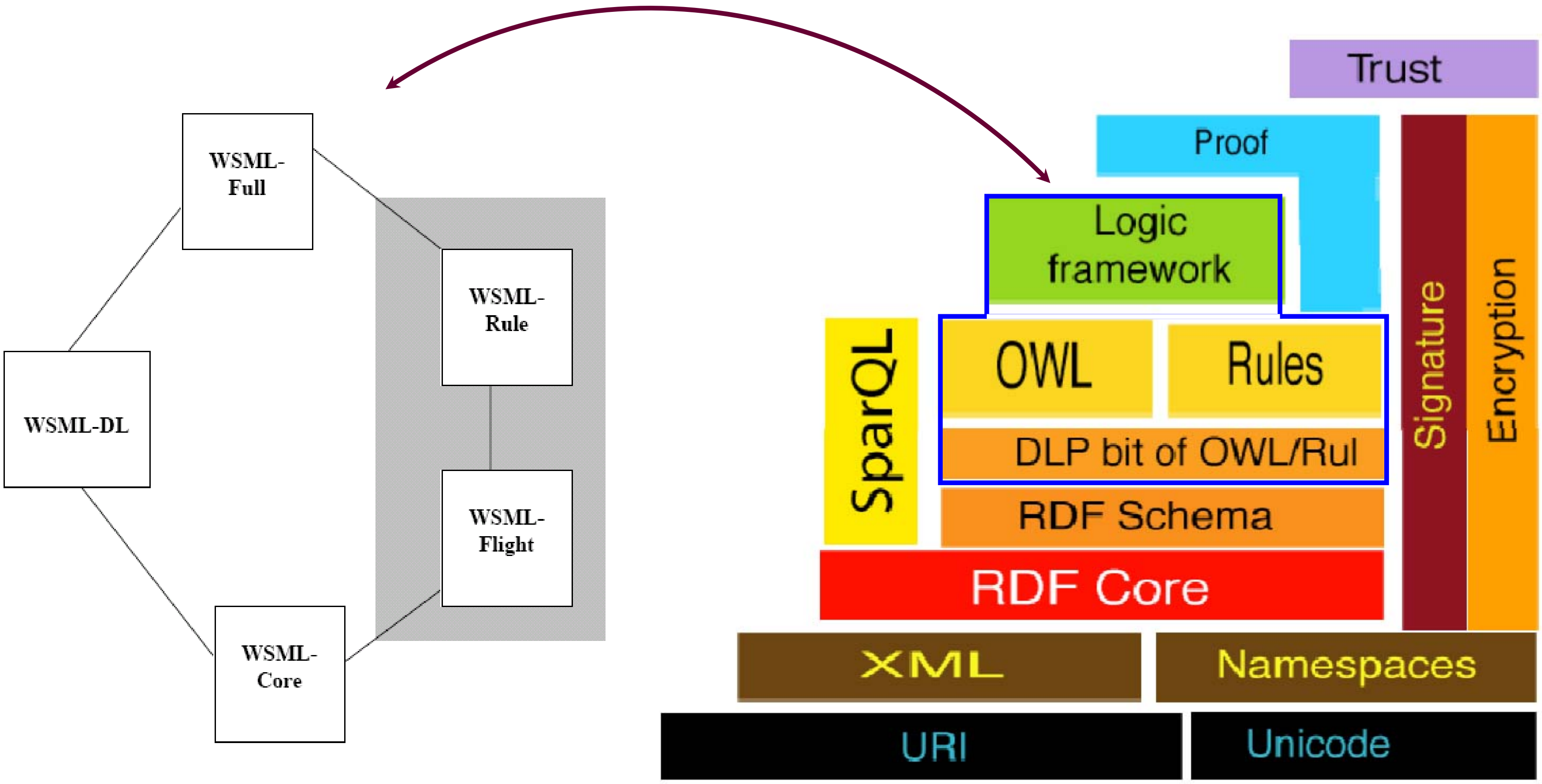
The Web Service Modelling Language (WSML)

- Ontology / Rule Languages
  - WSML Core: efficiency and compatibility
  - WSML DL: decidability, open world semantics
  - WSML Rule: efficient existing rule engines
  - WSML Full: unifying language, theorem proving
- Languages for dynamics
  - Transaction Logic over ASMs
- Mapping languages
  - for dynamics (process mediation)
  - for data (data mediation)





WSML – relation to SW standards



WSMO/WSML – Some Modelling Examples

- Concept example

```
concept phoneNumber
   nonFunctionalProperties
     dc#description hasValue "concept of a
          phone number"
   endNonFunctionalProperties
   countryCode ofType _string
   areaCode ofType _string
   number ofType _string
```

- **Sub-concept example**

```
concept mobilePhoneNumber subConceptOf
      phoneNumber
   nonFunctionalProperties
     dc#description hasValue "concept of a
          mobile phone number"
   endNonFunctionalProperties
   mobileProvider ofType Provider
```

- Relation example

```
relation hasRoute(ofType routeDescription, ofType route)
   nonFunctionalProperties
     dc#description hasValue "Relation that holds between
          a route description and a route"
   endNonFunctionalProperties
```

- Instance example

```
instance myPhoneNumber memberOf phoneNumber
   countryCode hasValue "43"
   areaCode hasValue "664"
   number hasValue "49322607"
```

- Axiom example

```
axiom ValidInformationQuality
   definedBy
     forall {?x} (
       ?x memberOf informationQualityType
     implies
       ?x[value hasValue "low"] or
       ?x[value hasValue "high"]).
```

WSMO/WSML – Some Modelling Examples (cont')


```
webService _"https://asg-platform.org/AttractionBooking/MobtelPhoneLocationService"
    nfp
        dc#title hasValue "MobtelPhoneLocationService"
        dc#publisher hasValue "Mobtel“
        dO#informQualityType hasValue "high"
    endnfp
    importsOntology _"https://asg-platform.org/AttractionBooking/domainOntology.wsml"
    capability MobtelPhoneLocationServiceCapability
        sharedVariables {?P}
        precondition
            definedBy
                ?P memberOf dO#phoneNumber.
        postcondition
            definedBy
                ?L memberOf dO#location
                 and
                dO#hasLocation(?P,?L).
    interface MobtelPhoneLocationServiceInterface
        choreography MobtelPhoneLocationServiceChoreography
            stateSignature
              in
                dO#phoneNumber withGrounding
        ssWSDL#wsdl.interfaceMessageReference(MobtelPhoneLocationServicePortType/doIt/In)
             out
                dO#location withGrounding
        ssWSDL#wsdl.interfaceMessageReference(MobtelPhoneLocationServicePortType/doIt/Out)
            transitionRules
                forAll{?P} with (?P memberOf dO#phoneNumber) do
                    add(?L memberOf dO#location and dO#hasLocation(?P,?L))
                endForall
```

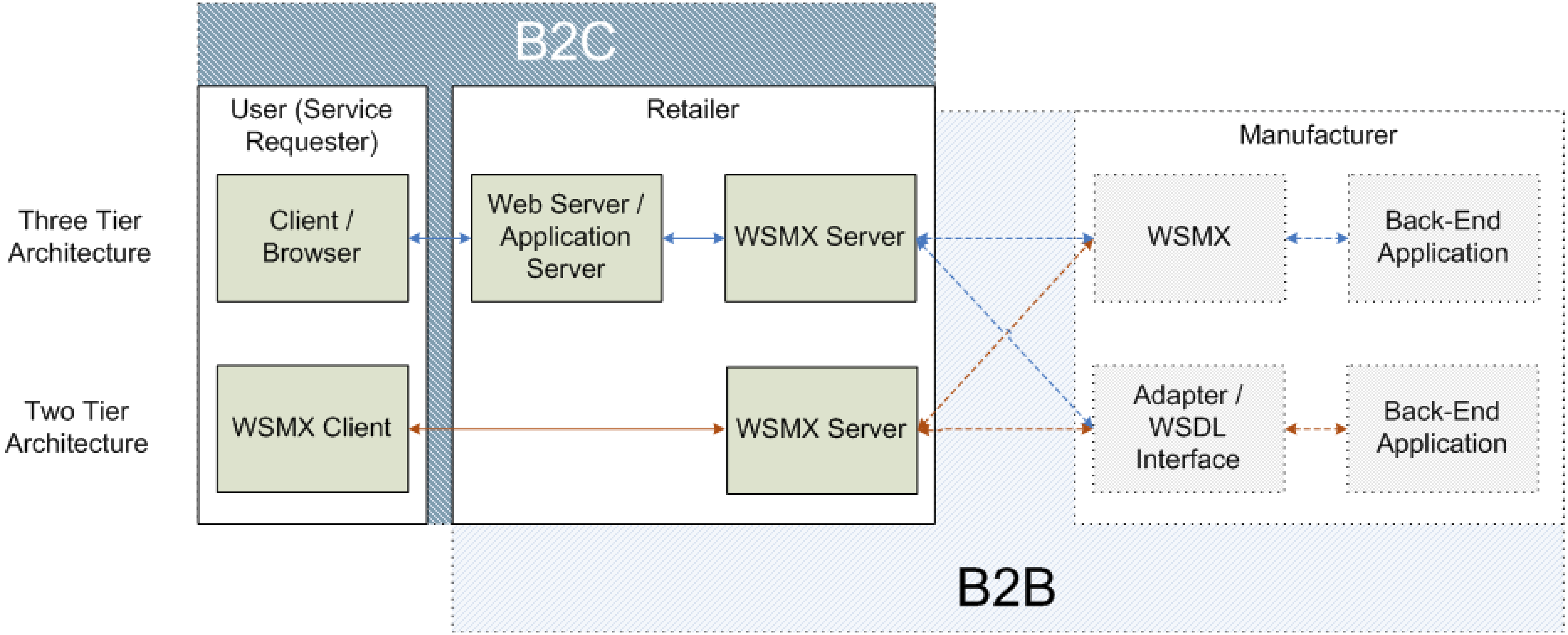
The Web Service Execution Environment (WSMX)

- A software framework for runtime binding of service requesters and service providers
- WSMX interprets service requester's goal to
  - discover matching services
  - select (if desired) the service that fits best
  - provide mediation (if required)
  - make the service invocation
- Is based on the conceptual model provided by WSMO
- Has a formal execution semantics
- SO and event-based architecture based on microkernel design using technologies as J2EE, Hibernate, JMX, etc.

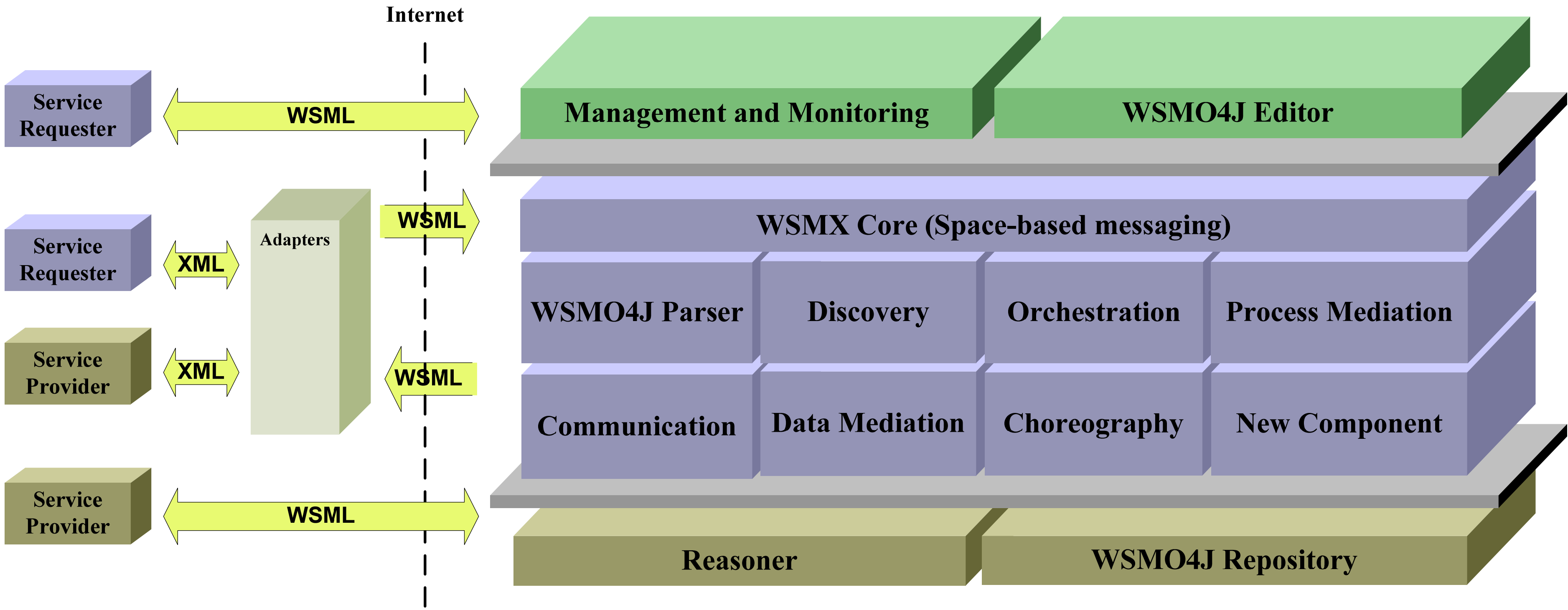
WSMX Motivation

- Provide middleware ‘glue’ for Semantic Web Services
  - Allow service providers focus on their business
- Provide a reference implementation for WSMO
- Provide an environment for goal based service discovery and invocation
  - Run-time binding of service requester and provider
- Provide a flexible Service Oriented Architecture
  - Add, update, remove components at run-time as needed
- Keep open-source to encourage participation
  - Developers are free to use in their own code
- Define formal execution semantics
  - Unambiguous model of system behaviour

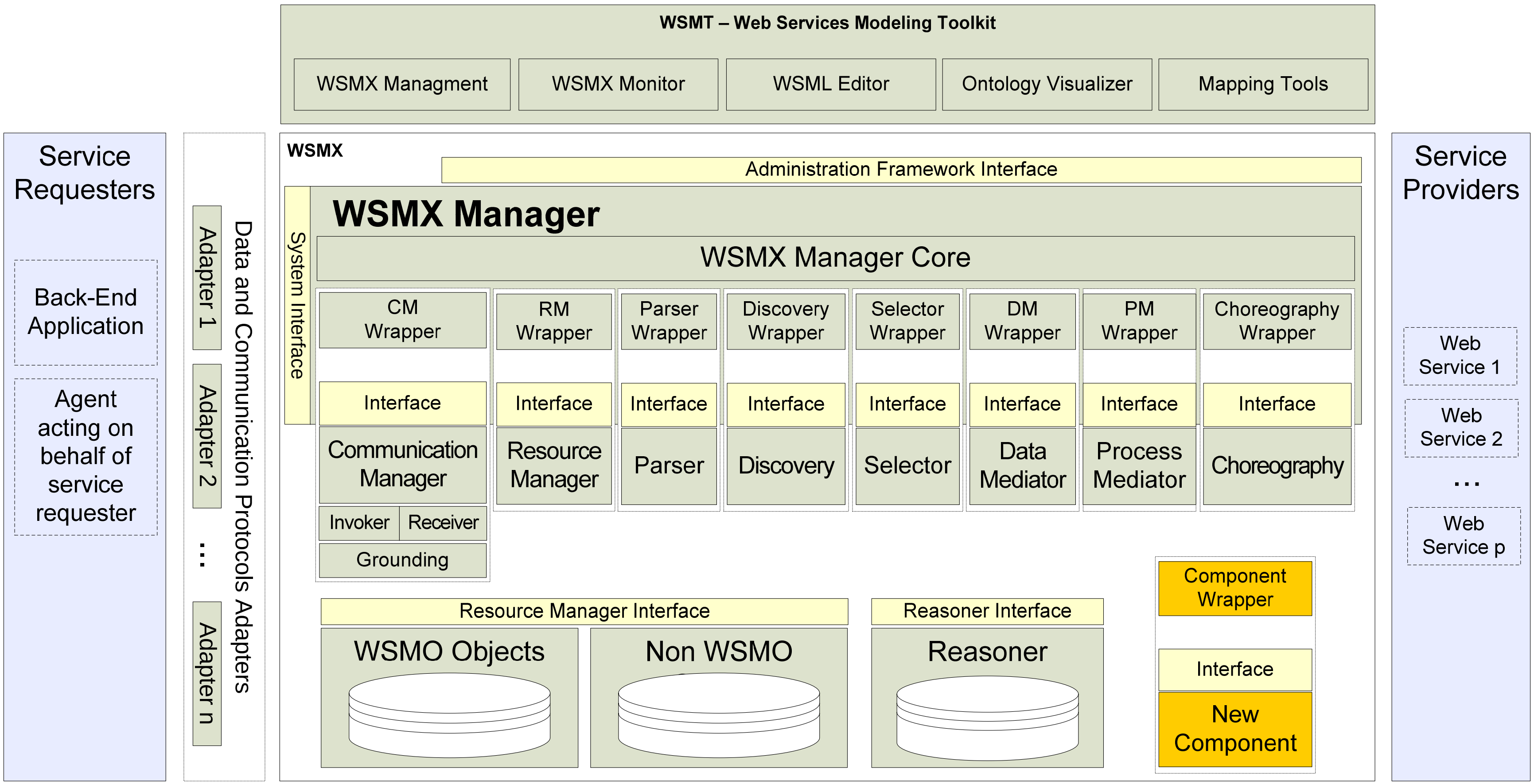
WSMX Usage Scenario



WSMX Components



WSMX Architecture



WSMX Discovery Component

- Functionality
  - Identify possible web services W which are able to provide the requested service S for its clients
- An important issue …
  - „being able to provide a service“ has to be determined based on given descriptions only (WS, Goal, Ontos)
  - Discovery can *only be as good* as these descriptions
    - *Very detailed WS descriptions*: are precise, enable highly accurate results, are more difficult to provide; in general, requires interaction with the provider (outside the pure logics framework)
    - *Less detailed WS descriptions*: are easy to provide for humans, but usually less precise and provide less accurate results

Ease of provision

Possible Accuracy

WSMO Discovery – Levels of abstractions

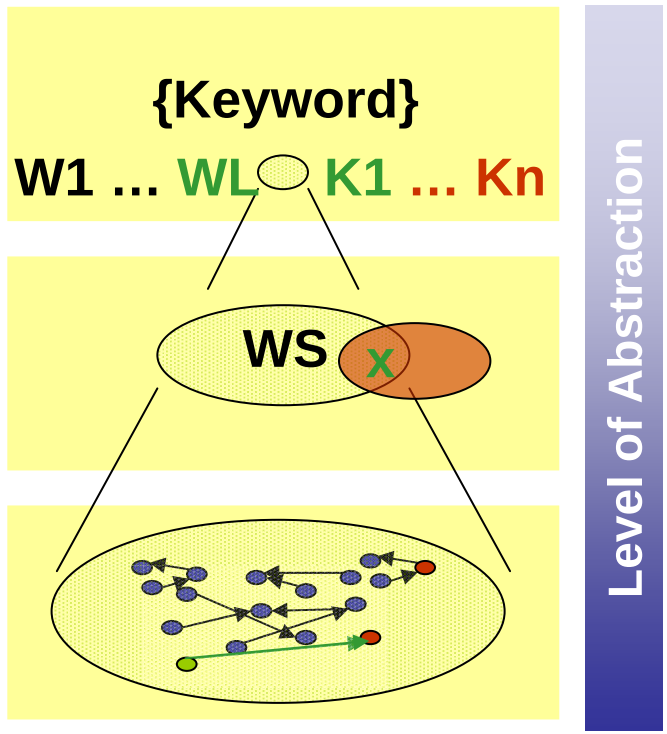
- Support a wide-variety of applications wrt. needed accuracy
- Basic possibilities for the description of web services:
  - Syntactic approaches → WS as a set of keywords
    - Keyword-based search, natural language processing techniques, Controlled vocabularies
  - Lightweight semantic approaches → WS as a set of objects
    - Ontologies, What does W provide (not how)?, Action-Object-Modelling, Coarse-grained semantic description of a service
  - Heavyweight semantic approaches → WS as a set of state-changes
    - Describes the service capability in detail, Pre/Post-Cond, takes „in-out“ relationship into account, Fine-grained web service description

Level of Abstraction

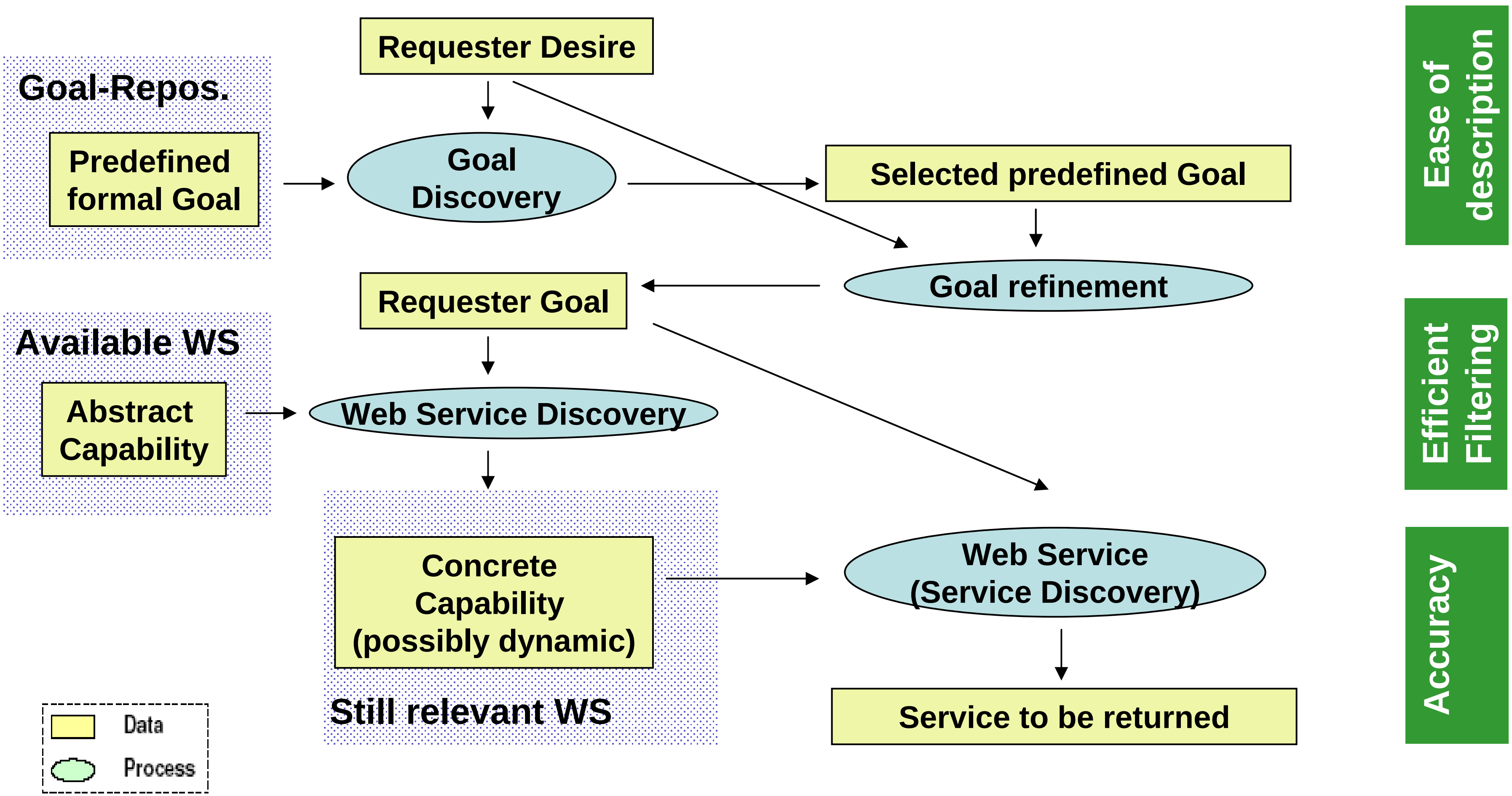
Common keywords — **Syntactic**

Set-theoretic relationship — **Semantic („Light“)**

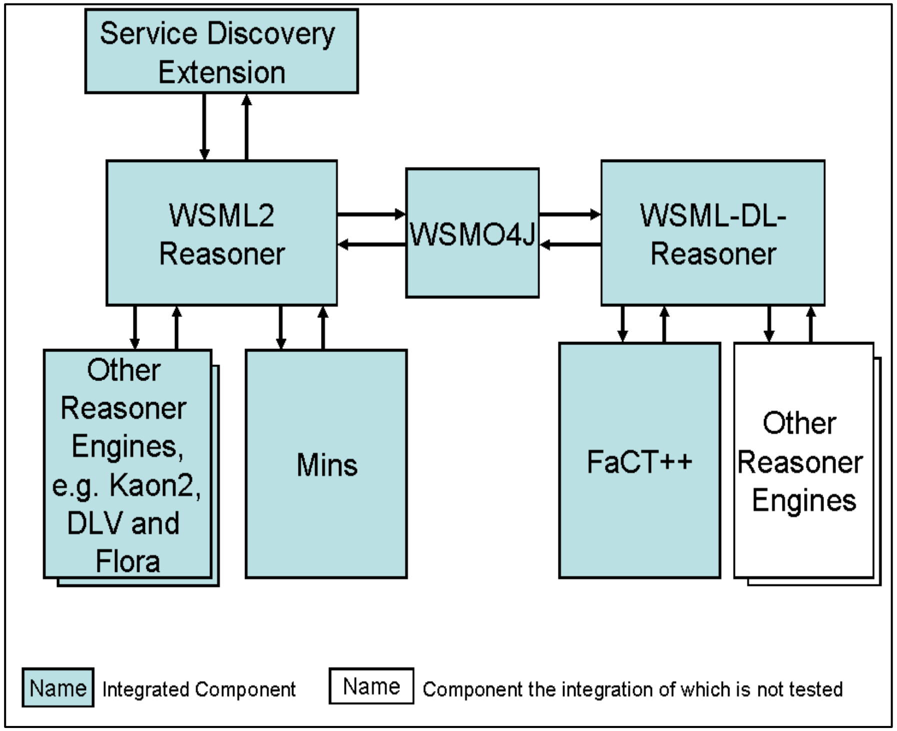
Adequate (common) execution/ state-transition — **Semantic („Heavy“)**



WSMO Discovery Process



WSMX Reasoner Component – an Overview



- Mins
  - Datalog + Negation + Function Symbols Reasoner Engine
  - Features
    - Built-in predicates
    - Function symbols
    - Stratified negation

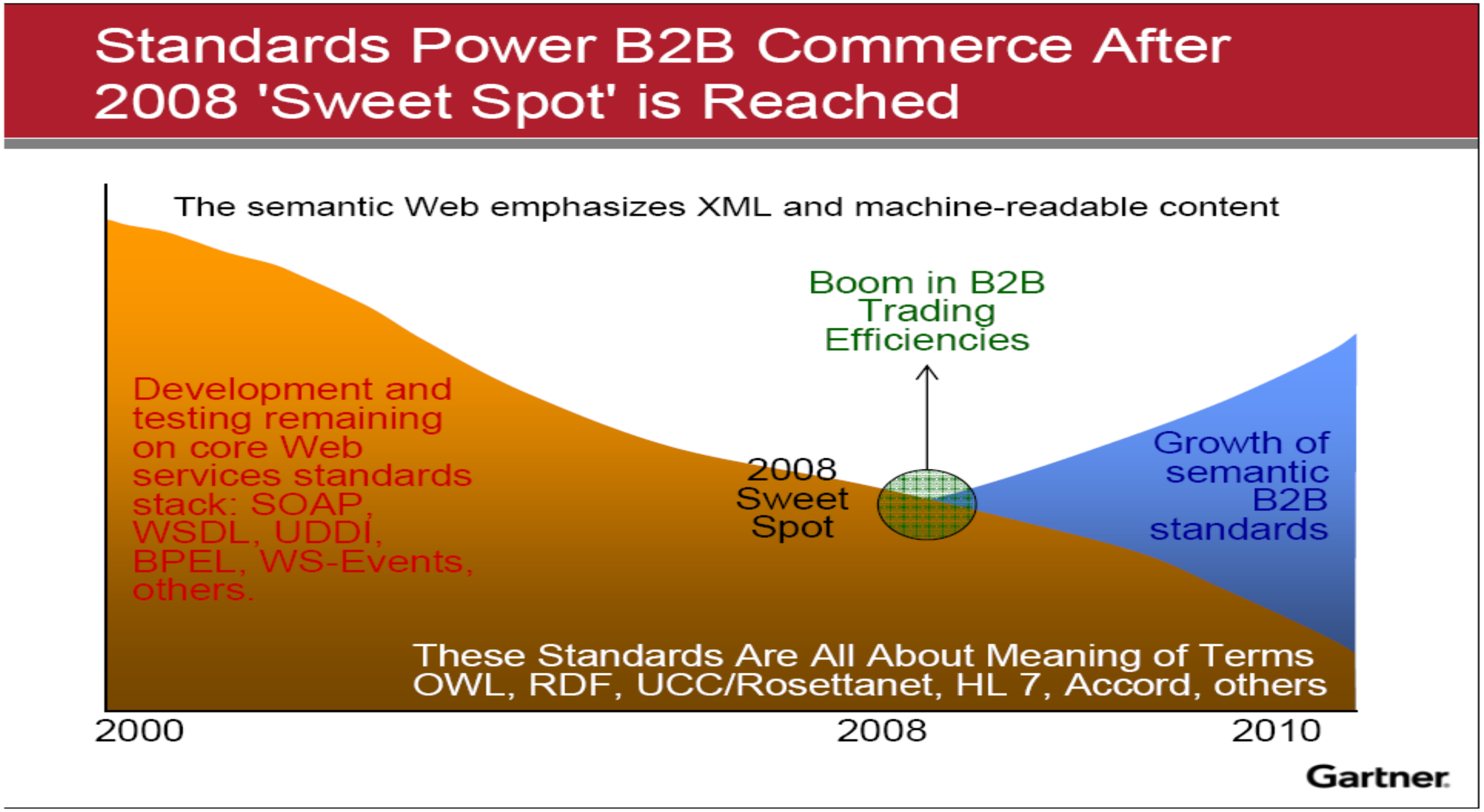
- WSMO4J
  - validation, serialization and parsing
- WSML2Reasoner
  - Reasoning API
    - mapping fromWSML to a vendor-neutral rule representation
  - Contains:
    - Common API for WSML Reasoners
    - Transformations of WSML to tool-specific input data (query answering or instance retrieval)
- WSML-DL-Reasoner features:
    - T-Box reasoning (provided by FaCT++)
    - Querying for all concepts
    - Querying for the equivalents, for the children, for the descendants, for the parents and for all ancestors of a given concept
    - Testing the satisfiability of a given concept with respect to the knowledge base
    - Subsumption test of two concepts with respect to the knowledge base
    - Wrapper of WSML-DL to the XML syntax of DL used in the DIG interface

Outline

- Semantic Web
- Web Services
- Semantic Web Services (SWS)
  - Tasks to be Automated
- Existing Approaches to SWS
  - OWL-S, SWSF, IRS-III, WSDL-S
  - The WSMO Approach: WSMO, WSML, WSMX
- **Conclusions**
- Proposed Challenge for Measuring Success of SWS

But first what about the predicted impact of semantic technologies? (Gartner, Oct. 2005)

Strategic Planning Assumption: Enterprise clusters that adopt robust semantic B2B specifications and Web services will achieve savings of 10 percent or more in trading processes by year-end 2007 (0.7 probability).



Semantic Web – Conclusions

- The Semantic Web is **real** – tremendous progress in the past five years
  - Growing support in industry and govt use
  - Lots of tools out there:
    - **Browsers**: mSpace, Longwell, OINK, BrownSauce, Piggy Bank, Tabulator, etc
    - **Annotators**: Annotea, Clipmarks, PhotoStuff, M-OntoMat-Annotizer, KIM, WSMT
    - **Storages**: Oracle Spatial 10g, Kowari, Jena, Yars, 3Store, AllegroGraph, Joseki, ARC RDF Store
    - **Ontology Mappers**: OntoMerge, HMARFA, CMS
    - **Reasoners**: BOR, Bossam, FaCT++, Jess, OWLJessKB, RacerPro
    - **Composite Applications/Frameworks**: Cerbera, Corse, IODT, Jena, TopBraid Composer, KAON
- New languages under way
  - GRDDL/RDFa – integration of HTML world and Semantic Web
  - RIF (Rules interchange format) – representing rules on the Web
  - And more: Multimedia annotation, Web-page Metadata annotation, Health Care and Life Science, Privacy
- Easy to get involved – many open source tools; new languages and techniques reaching critical mass
- …and research opportunities still abound: scaling, inconsistency, access and acquisition, etc.

Semantic Web Services – Conclusions

- Although lots of progress in the last couple of years, SWS are still immature technologies; not too many use cases
  - But high potential in BPM: B2B, EAI, eCommerce, etc.
- The WSMO Approach to SWS looks promising
  - Covers many aspects of SWS; unifying approach
  - Large-scale ongoing initiative supported by both industry and academia
- Standardization activities are emerging in this area
  - OWL-S, SWSF, WSDL-S, WSMO – submitted to W3C
  - OASIS SEE technical committee formed (based on WSMX)
  - W3C SAWSDL Working Group formed; close to recommendation
- More collaboration is needed between research community and the industrial community
- The biggest challenge for the future: the movement to service-orientation and the semantic enablement of industrial scale infrastructures and applications

Outline

- Semantic Web
- Web Services
- Semantic Web Services (SWS)
  - Tasks to be Automated
- Existing Approaches to SWS
  - OWL-S, SWSF, IRS-III, WSDL-S
  - The WSMO Approach: WSMO, WSML, WSMX
- Conclusions
- **Proposed Challenge for Measuring Success of SWS**

Proposed Challenge for Measuring Success of SWS

- An industrial-scale application that has run successfully in a production environment for at least six months; the application must:
  - Support some form of collaboration substantially involving at least five internal or external, but separate, organizations
  - Consist of at least 1,000 entity types and 1,000 service types with at least 2 subscribers per service type.
  - Have an average daily service transaction rate of 10 million service executions.
  - Comply with at least three industrial standard ontologies and the majority of relevant SOA and Web service standards.
  - Support at least 20 concurrent domain-specific problem solvers.
  - Support problem solving in at least two distinct aspects (e.g., ordering and billing) of standard problem domain, e.g., manufacturing, financial services, health care, inventory, and tourism. This requires at least:
    - 10 industry-specific tools of which 5 must be standard industry practice
    - 10 industry-specific problem solving capabilities supported by the tools
    - Automatic workspace configuration: The application must automatically configure the workplace for each user with the tools and capabilities required by each user as defined in their profile which defines their roles, responsibilities, and user-specified or system-deduced configuration preferences.

Proposed Challenge for Measuring Success of SWS (cont')

- At least 50% of service discovery, selection, negotiation, adaptation, composition, invocation, and monitoring, as well as service interaction requiring data, protocol, and process mediation -- are fully automated, with no human intervention.
  - At a minimum, this must dynamically and automatically address and resolve the conflicting non-functional business aspects that arise when a consumer discovers a service offered by a producer with whom there is no business agreement for the discovered service.
- Normal business: All normal business conventions must apply. There must be significant, e.g., legal and financial, consequences should there be a failure in any of the above automated service operations.
- All service offerings and requests are expressed in terms of service descriptions that contain:
  - a functional and behavioural specification expressed in semantic terms consistent with one or more industrial standard ontologies, and
  - a non-functional specification consisting of at least 5 non-functional terms such as price, promised service levels (SLAs), and performance characteristics.

STI2

- **STI2** - Semantic Technology Institutes International
  - the leading international think tank in this field unifying three major initiatives:

knowledgeweb

European Semantic Systems Initiative

DERI

  - DERI Innsbruck – a founding member of STI2

STI2 Services

Research	Technology	Realisation
Ontology Engeneering	Standardisation	Education
Reasoning	Architecture	Dissemination
Semantic Web Services	Software	Commercialisation
	Testbeds	
	Challenges	

More details: www.STI2.org

Thank you!

Questions?

Acknowledgements

- This tutorial is based on the WSMO Tutorial Series, series to which several people contributed. For the list of contributors to the WSMO tutorials, and the various WSMO tutorials, please check http://www.wsmo.org/TR/d17/
- Some of the slides related to ontologies are based on the Semantic Web lecture at University of Innsbruck, slides which were initially developed by Jos de Brujin and Stijn Heymans