Semantically Enabled Service-Oriented Architectures: A Framework

Dieter Fensel, Michal Zaremba

DERI International

Dieter Fensel, Michal Zaremba
<firstname.lastname>@deri.org

INTAP Semantic Web Conference 2006

But first a few words about us…

- Digital Enterprise Research Institute (DERI) - our vision is to make the Semantic Web and Semantic Web Services a reality enabling fully flexible eCommerce for small, medium-sized and large enterprises.
  - Semantic Web Services have the potential to become a key-enabling infrastructure for Knowledge Management and eWork, Enterprise Application Integration, and eCommerce
  - In consequence, Semantic Web Services are one of the key areas of applied computer science

DERI International – Status

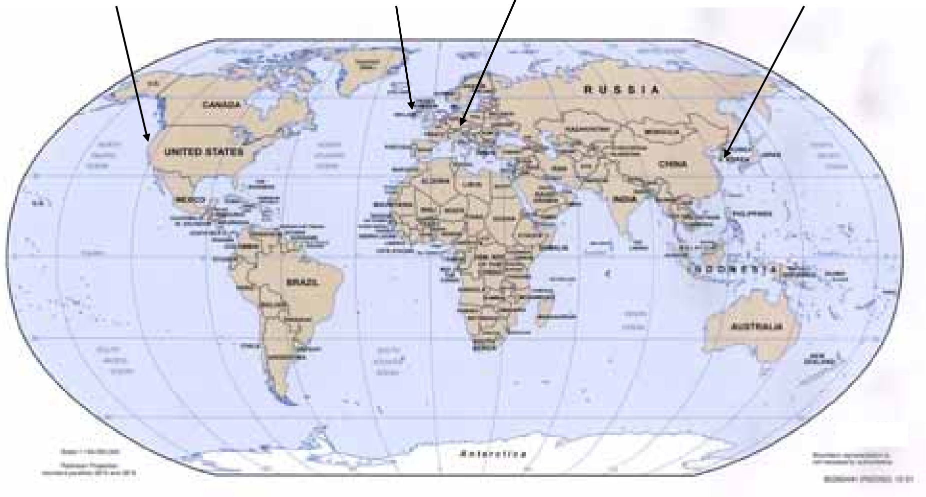
Digital Enterprise Research Institute www.deri.org

DERI Stanford

DERI Galway

DERI Innsbruck

DERI Seoul



DERI International – Status

Digital Enterprise Research Institute www.deri.org

DERI International € 42 Mio

DERI Stanford € 2 Mio

- DARPA, Defence Advanced Research Projects Agency: 0,15M€
- DARPA, subcontracted through NRL (Naval Research Laboratory: 1,85M€

DERI Seoul € 5,5 Mio

- Korean Ministry of Health & Welfare: 4M€
- Ministry of Information And Communication: 1,2M€

DERI Innsbruck € 15,5 Mio

- European Commission: €10,1M€
- Tiroler Zukunftsstiftung: 2M€
- TransIt: 1,4M€
- FFF/FFG/WWFF: 1,5M€
- BMBWK: 0,5M€

DERI Galway € 19 Mio

- SFI: 12M€
- European Union: €6,6M
- Enterprise Ireland: 0,3M€

DERI International – Status (July 2005)

Projects, DERI is involved in (around 188M€):

• ASG:	12 M€	COG:	2 M€
• DERI Lion:	12 M€	DIP:	18 M€
• DSSE:	0.4M€	Eastweb:	1 M€
• Esperonto:	3 M€	E-Swan:	0,5M€
• Grisino:	0,5M€	H-techsight:	3M€
• Infrawebs:	3M€	Knowledge Web:	8 M€
• LBSCULT:	0,2M€	M3PE:	0,2 M€
• Musing:	15M€	Nepomuk:	16 M€
• Ontoweb:	2M€	RW2:	0.4M€
• Salero:	14M€	SAOR:	0,3 M€
• SEKT:	13M€	SEEMP:	5M€
• Semantic Gov:	5 M€	Sembiz:	0,6M€
• SemNetMan:	0.5M€	Sense:	0,7M€
• SUPER:	15 M€	Swing:	4 M€
• SWWS:	3 M€	SystemOne:	2.1M€
• TRIPCOM:	5M€	Interoperable EHR:	16M€
• Semantic Service Engine	4M€	TSC:	0.3M€
• GGP Testbed	0,15M€	Learning Evaluation:	1.85M€

DERI – Team 2005

DERI 2006 - 2007 (Future Plans ☺)

The technological Vision

Semantically Empowered Service-oriented Architectures (SESA)

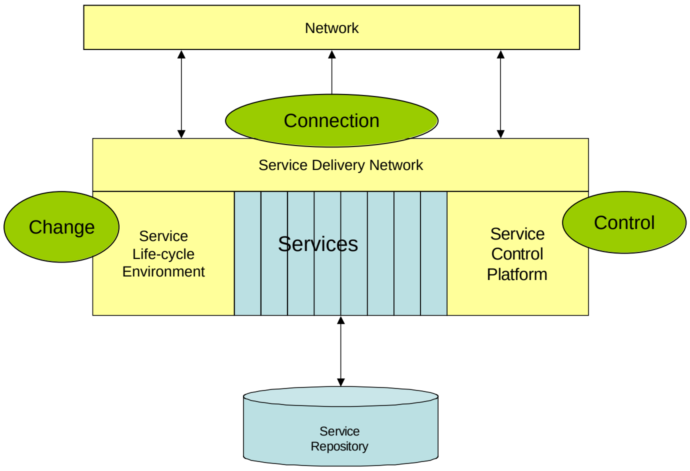
- Currently, computer science is in a new period of abstraction.
- A generation ago we learnt to abstract from hardware and currently we learn to abstract from software in terms of SERVICE oriented architectures (SOA).
- It is the service that counts for a customer and not the specific software or hardware that is used to implement the service.
- In a later stage, we may even talk in terms of problem-oriented architectures (or more positively expressed in terms of problem-solving oriented architectures) because SOAs are biased towards the service provider and not towards the customer that has a problem that needs to be solved.

Semantically Empowered Service-oriented Architecture (SESA)

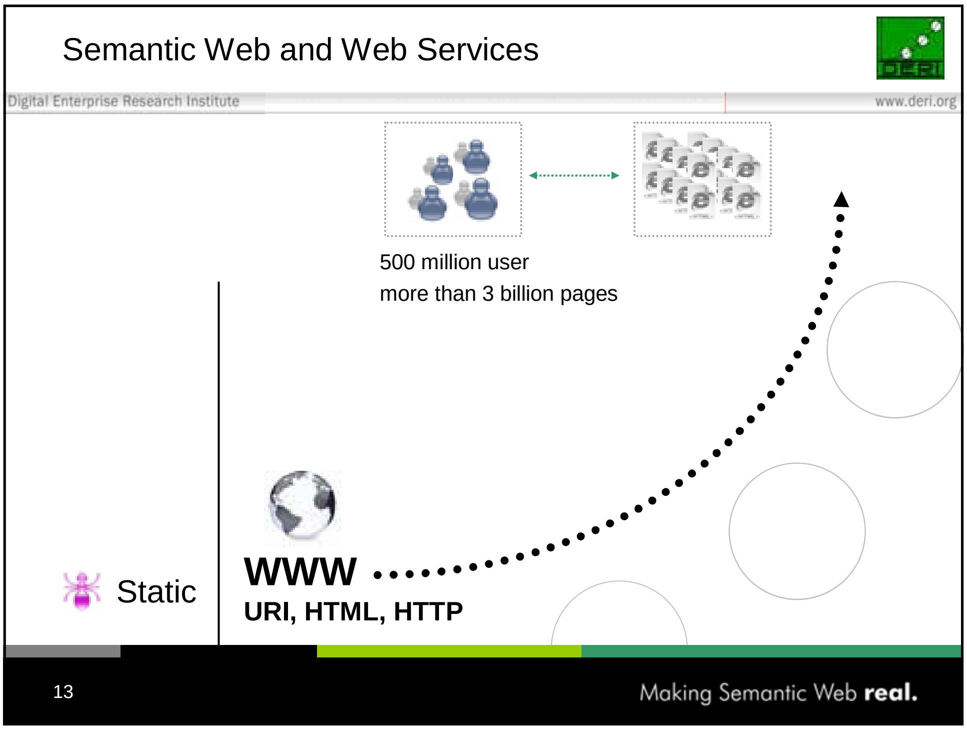
- Service-oriented architectures will become quickly the leading software paradigm
- However, SOAs will not scale without significant mechanization of
  - Service discovery, service adaptation, negotiation, service composition, service invocation, and service monitoring; and
  - Data and process mediation
- Therefore, machine processable semantics needs to be added to bring SOAs to their full potential
- Development of open standards (languages) and open source architectures and tools that add semantics to service descriptions

The SESA Manifesto (M. Brodie et al.)

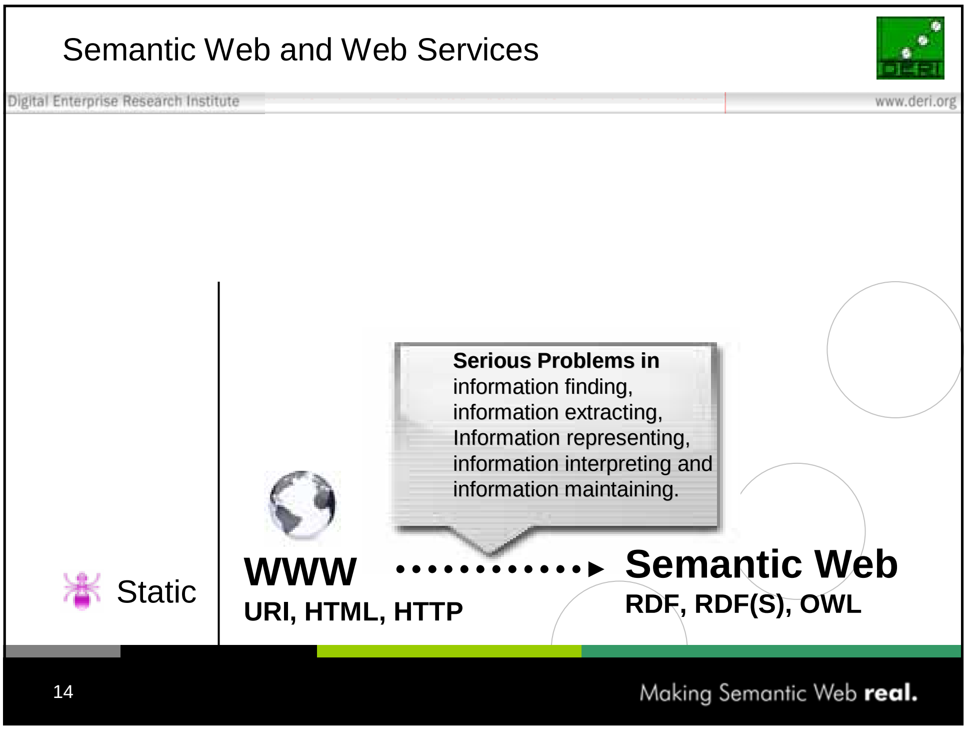
The operation system of the 21st century based on semantics



Background



Semantic Web and Web Services
Digital Enterprise Research Institute
www.deri.org
500 million user
more than 3 billion pages
Static
WWW
URI, HTML, HTTP
13
Making Semantic Web real.



Semantic Web and Web Services
Digital Enterprise Research Institute
www.deri.org
Serious Problems in
information finding,
information extracting,
Information representing,
information interpreting and
information maintaining.
Static
WWW
URI, HTML, HTTP
Semantic Web
RDF, RDF(S), OWL
14
Making Semantic Web real.

Semantic Web and Web Services

Dynamic

Static

Web Services
UDDI, WSDL, SOAP

Bringing the computer back as a device for computation

WWW
URI, HTML, HTTP

Semantic Web
RDF, RDF(S), OWL

Semantic Web and Web Services

Digital Enterprise Research Institute www.deri.org

Bringing the Web to its full potential

Dynamic

Static

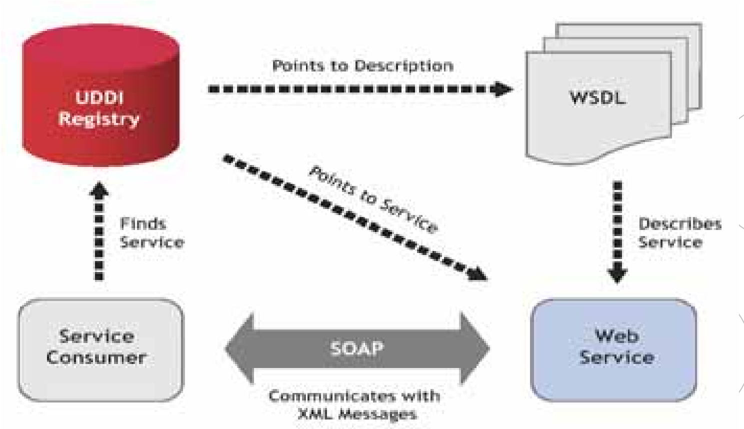
Web Services
UDDI, WSDL, SOAP

Intelligent Web Services

WWW
URI, HTML, HTTP

Semantic Web
RDF, RDF(S), OWL

State-of-the-Art in (non-semantic) Web Services



Usage Process

- Publication: Make available the description of the capability of a service
- Discovery: Locate different services suitable for a given task
- Selection: Choose the most appropriate services among the available ones
- Composition: Combine services to achieve a goal
- Mediation: Solve mismatches (data, process) among the combined
- Execution: Invoke services following programmatic conventions

Usage Process – execution support

- Monitoring: Control the execution process
- Compensation: Provide transactional support and undo or mitigate unwanted effects
- Replacement: Facilitate the substitution of services by equivalent ones
- Auditing: Verify that service execution occurred in the expected way

Mechanization of Finding, Comparing, Data and Process Mediation

- Mechanized support is needed in **finding and comparing service providers** and their offers
  - Machine processable semantics of information allow to mechanize these tasks
- Mechanized support is needed in dealing with **numerous and heterogeneous data formats**
  - Ontology technology is required to define such standards better and to map between them
- Mechanized support is needed in dealing with **numerous and heterogeneous business and application logics**
  - Mediation is needed to compensate these differences, allowing partners to cooperate properly

Concepts

DERI – W<Triple>

W<Triple> which stands for:

- ***W***SMO: A conceptual model for describing service oriented architectures
- ***W***SML: A formal language for describing service oriented architectures
- ***W***SMX: A service oriented architecture
- ***Triple*** space: A shared space for heterogeneous services that communicate via persistent publication

WSMO is…

- A conceptual model for Semantic Web Services :
  - Ontology of core elements for Semantic Web Services
  - a formal description language (WSML)
  - execution environment (WSMX)
- … derived from and based on the Web Service Modeling Framework WSMF
- an ESSI-Cluster Working Group
  (joint European research and development initiative)

WSMO Design Principles

Web Compliance **Ontology-Based**

Strict Decoupling **WSMO** **Ontological Role Separation**

Centrality of Mediation **Execution Semantics**

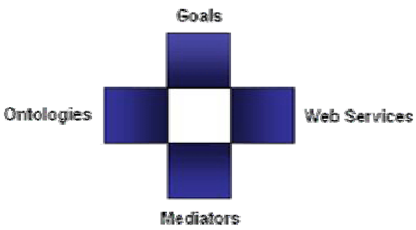
Description versus Implementation

Concepts – Technological basis: WSMO (1)

(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

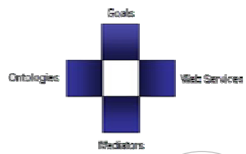


Semantic description of Web Services:
- **Capability** *(functional)*
- **Interfaces** *(usage)*

Connectors between components with mediation facilities for handling heterogeneities

Four top level elements – cornerstone of conceptual model

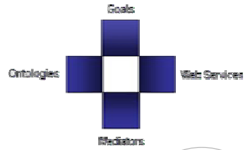
Concepts – Technological basis: WSMO (2)



WSMO V2.0; topics for model refinement:

- **Goals**
  - Goal repositories, goal decomposition, non-functional properties
- **Semantic Web services**
  - Relationship to WSDL, non-functional properties
- **Mediators**
  - Deeper understanding of nature of OO, GG, WG, WW mediators
- **Ontologies**
  - Develop for various domains (e.g. EDI), measure usage

Concepts – Technological basis: WSMO (3)



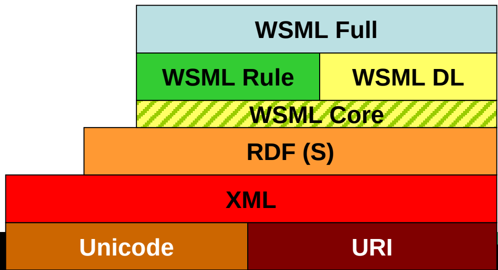
The big challenge of defining a Semantic Web service

- **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.**

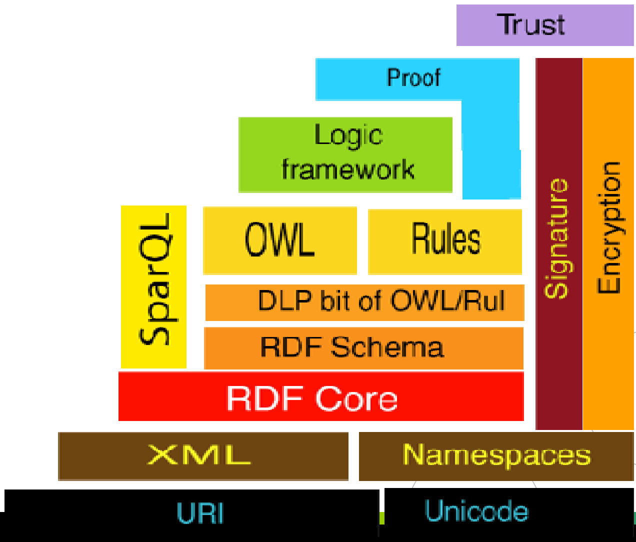
Technological basis: WSML

WSML: The **Web Service Modeling Language**

- A family of language layered on top of XML and RDF.



Technological basis: WSML



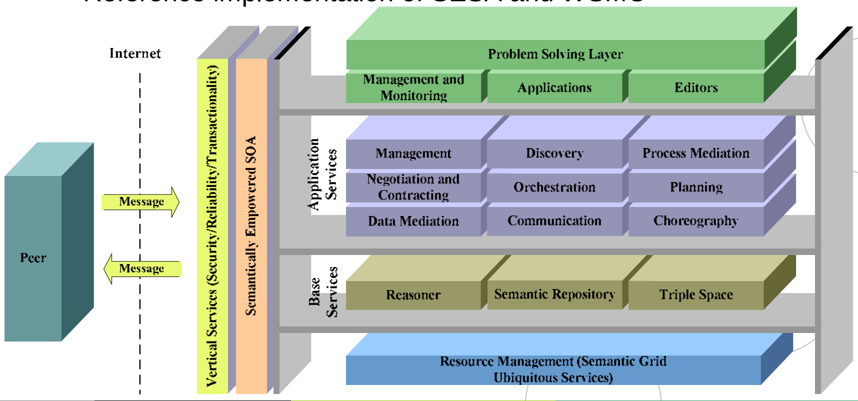
Languages

- A set of concrete languages is needed for the various tasks:
- Ontology / Rule Languages (static view)
  - 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)

Concepts – Technological basis: WSMX

WSMX: The **Web Service EXecution Environment**

- A service oriented architecture.
- Reference implementation of SESA and WSMO



Concepts – Technological basis: Triple Space (1)

	Message	Publishing
Machine net	Web Services	**Triple Space**
Human net	email	web

DERI – Triple Space

Communication platform for Semantic Web services based on Web principles:

"Persistently publish and read semantic data that is denoted by unique identifiers"

Fundamentals:

- Space-based computing – sharing information, knowledge
- RDF triples of the form: <subject, predicate, object>
- URI – Uniform Resource Identifier

Concepts – Technological basis: Triple Space (3)

Digital Enterprise Research Institute www.deri.org

Triple Spaces allow for:

- Time autonomy
- Location autonomy
- Reference autonomy
- Vocabulary autonomy

➔ Triple Spaces provide a communication paradigm for *anonymous*, *asynchronous* information exchange that ensure the *persistency* and *unique identification* of the communicated semantic data.

Concepts - Resource Management Functionality

- **Semantic Grid**
- **Ubiquitous Services**

Concepts – Resource Management Functionality: Semantic Grid (1)

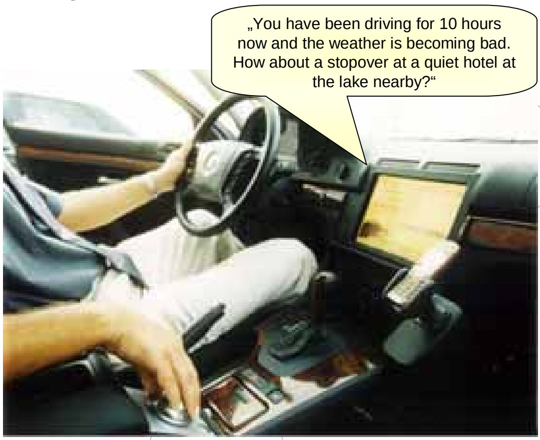
- Basic Entities:
- Grid
  - Co-ordinated resource sharing over the Web
- Web Services Resource Framework (WSRF)
  - Specifications linking resources to Web services
- Open Grid Service Archtecture (OGSA)
  - A SOA for grid computing
- Semantic Web Services
  - Provide the Web endpoints for Grid resources
  - Facilitate discovery, composition, mediation
- Non functional properties:
  - Require particular focus e.g. reliability, price, availability

Concepts – Resource Management Functionality: Ubiquitous Computing (1)

„Ubiquitous Computing Paradigm":

- the right service
- at the right place
- at the right time
- (at the right cost)

„magic beyond the scene"



Concepts – Resource Management Functionality: Ubiquity vs. GRID: covering both ends

Architectural Support

… **Discovery** **Execution** …

Ubiquitous Services **GRID Services**

Semantics (SWS)
Context (Modeling & Retrieval)
Protocol (Choreography & Orchestration)
Syntax

Pervasive Spaces **Triple Space** **Semantic GRID**

challenges

Partial Reasoning
Compensation (Handover)
Dynamics
- Network Reliability
- Contracting/SLA
Storage Capacity

Negotiation & Adaption
(Functional **and** Non-Functional Properties)

challenges

Full Reasoning
Context-Awareness
- Personalization
- Mobility (Location)
Load Distribution
Virtual Organization
Mediation

DERI's approach to tackle the challenges

- **International working groups**
  - WSMO Working Group → define the models
  - WSML Working Group → define the languages
  - WSMX Working Group → define and implement the execution environment
- The working groups are part of the SDK-cluster which is a joined dissemination and coordination activity of three leading European projects in the semantic web and semantic web service area.

SEKT (Semantically-Enabled Knowledge Technologies)
http://sekt.semanticweb.org/

DIP (Data, Information and Process with Semantic Web Services)
http://www.nextwebgeneration.org/projects/dip/

Knowledge Web
http://knowledgeweb.semanticweb.org/

http://www.sdk-cluster.org

Dissemination

European Semantic Systems Initiative

- The SDK (SEKT, DIP, Knowledge Web) Cluster has strategically aligned with ASG (Adaptive Services Grid) resulting in the formation of the European Semantic Systems initiative (ESSI) cluster.
- The new ESSI Cluster combines Semantic Web Services and Semantically empowered system solutions with Semantically empowered service-oriented architectures
- ASG will add value to the new ESSI cluster by providing a proof-of-concept prototype of an open platform for adaptive services discovery, creation, composition and enactment.

http://www.essi-cluster.org/

Case Study – Potential Application

Case Scenario - SWS applications in B2B (RosettaNet)

- Integration of information systems is important in cutting costs
- B2B integration even when using an XML-based standard, such as RosettaNet, the integrations can easily take e.g. six months so set up
- If a business partner gives a more formal description of how to interact with it, it is possible to use mediation technologies to adapt the interaction of the other business partner to be interoperable

Case Scenario - SWS applications in e-Banking

- Many banks offer online tools allowing customers to see current mortgage rates and the amount they could borrow. These tools are constrained by being limited to the mortgage products offered by just one bank
- There are websites that can aggregate information from multiple banks allowing the comparison of the various mortgage products. Used techniques: manual population, screen scrapping, Web Services and other
- Using SWS application/agent would automatically discover new SWS offering mortgage rate, the description of the interface would be examined automatically to determine how the application and service should communicate, once data mismatches have been resolved, the application retrieves the information about mortgages as required.

Telecom case study – realized in one of EU projects

- Service Assurance Across Organisational Boundaries
- Developed by British Telecom and NIWA
- Integration of the heterogeneous Operational Support Systems (OSS)
- Semantic descriptions of messages
- Allows semi-automatic mediation to be carried out
- Greater automation in the integration process

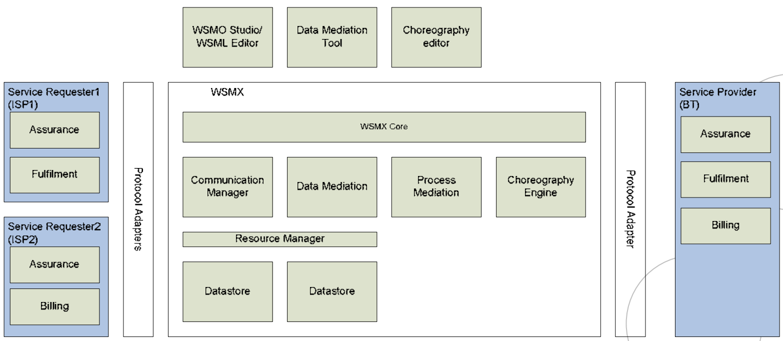
Telecom case study – Main process (1)

- A Customer informs his ISP of an error occurring in one of his products
- Error is passed to the ISP's trouble ticketing system.
- The ticketing system raises the problem with an operator
- -> a test should be carried out on the customer's line using the GUI of the OSS
- The OSS system produces a message in a specific XML format (including the data payload, describing the error and the Customer's product).

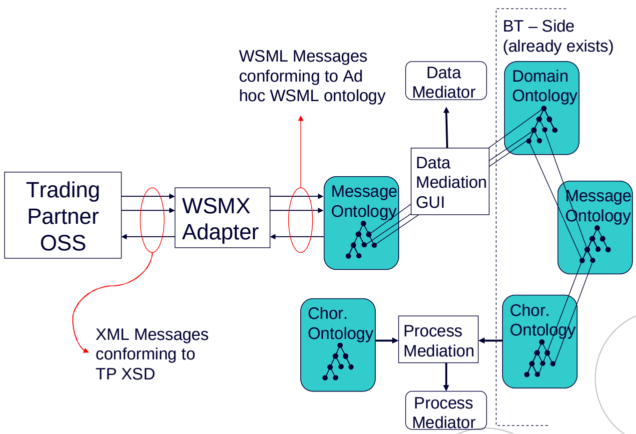
Telecom case study – Main process (2)

- The message is sent to the B2B Integration Platform
- Results in a test request being forward to BT
- BT's OSS receives the message and handles it appropriately, updating its GUI with details and status of the test.
- Upon completion of the test, the status is updated and an appropriate message is returned to the B2B Integration Platform
- A test request response being sent to the ISP which then updates it's GUI allowing the operator to see the result and act on it.

Telecom case study – prototype Architecture/Subcomponents of WSMX



Telecom case study – trading partner design time



Realization – Web Services Execution Environment OASIS Semantic Execution Environment

SEE WSMX Introduction

- 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 best fits
  - 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, Spring, JMX, etc.

WSMX Motivation

- Provide middleware 'glue' for Semantic Web Services
  - Allow service providers focus on their business
- Provide a reference implementation for WSMO
  - Eat our own cake
- 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

Design Principles

Strong Decoupling & Strong Mediation

autonomous components with mediators for interoperability

Interface vs. Implementation

distinguish interface (= description) from implementation (=program)

Peer to Peer

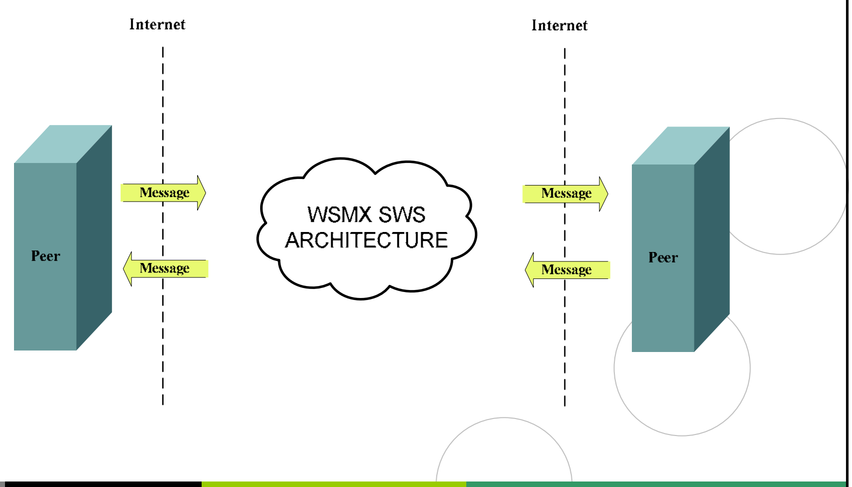
interaction between equal partners (in terms of control)

WSMO Design Principles == WSMX Design Principles == SOA Design Principles

WSMX as Service Oriented Architecture

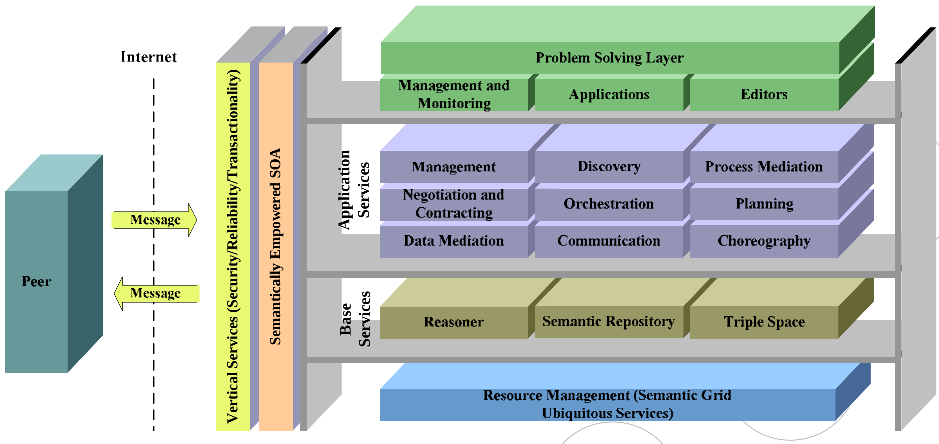
- Better reuse
  - Build new functionality (new execution semantics) on top of existing Business Services
- Well defined interfaces
  - Manage changes without affecting the Core System
- Easier Maintainability
  - Changes/Versions are not all-or-nothing
- Better Flexibility

WSMX Infrastructure



WSMX – Components

Digital Enterprise Research Institute www.deri.org



Benefits of SOA

- Better reuse
  - Build new functionality (new execution semantics) on top of existing Business Services
- Well defined interfaces
  - Manage changes without affecting the Core System
- Easier Maintainability
  - Changes/Versions are not all-or-nothing
- Better Flexibility

Service Oriented State

- The interface to the service is implementation-independent
- The service can be dynamically invoked
  - Runtime binding
- The service is self-contained
  - Maintains its own state

WSMX Uptake

- Creation of **OASIS Semantic Execution Environment SEE Technical Committee**.
- Interoperability
  - With IRS III from Open University, UK
- DIP
  - WSMX as reference implementation of DIP architecture
- Meteor-S – grounding meachanism
- Cocoon – joint contribution to OASIS SEE
- Business development
  - Vehicle for EU projects and partnerships (SEEMP, SemanticGov, Super, Swing, TripCom)

SEE WSMX Conclusions

- Conceptual model is WSMO
- End to end functionality for executing SWS
- Has a formal execution semantics
- Real implementation
- Open source code base at SourceForge
- Event-driven component architecture
- Growing functionality - developers welcome ☺

Conclusions

Conclusions

- The targets of this presentations were to:
  - understand aims & challenges within Semantic Web Services
  - understand vision of Semantically Enabled Service-Oriented Architectures
  - understand DERI approach to Semantic Web Services
  - present WSMX/SEE - future Web Service based IT middlewares
- => you should now be able to correctly assess Service Oriented Architectures and utilize these for your future work

Acknowledgements

The work presented is funded by the European Commission under the projects **ASG, DIP**, **Knowledge Web**, **SEKT**, **SWWS**, **AKT** and **Esperonto**; by **Science Foundation Ireland** under the **DERI-Lion** project; and by the Austrian government under the **FIT-IT** program.

We would like to acknowledge all the authors contributing to SESA vision: Michael Brodie[1], Christoph Bussler[2], Jos de Brujin[3], Thomas Fahringer[4], Dieter Fensel[3,5], Martin Hepp[3], Holger Lausen[3], Dumitru Roman[3], Thomas Strang[3], Hannes Werthner[6], and Michal Zaremba[3,5]

[1] Verizon, USA
[2] Cisco, USA
[3] DERI Innsbruck, Leopold-Franzens Universität Innsbruck, Austria
[4] Institute for Computer Science, Leopold-Franzens Universität Innsbruck, Austria
[5] DERI Galway, National University of Ireland, Galway, Ireland
[6] Department for Information Systems and e-tourism, Leopold-Franzens Universität Innsbruck, Austria