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 (July 2005)

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

<table>
<thead>
<tr>
<th>Project</th>
<th>Funding (M€)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASG</td>
<td>12</td>
</tr>
<tr>
<td>DERI Lion</td>
<td>12</td>
</tr>
<tr>
<td>DSSE</td>
<td>0.4</td>
</tr>
<tr>
<td>Esperonto</td>
<td>3</td>
</tr>
<tr>
<td>Grisino</td>
<td>0.5</td>
</tr>
<tr>
<td>Infrawebs</td>
<td>3</td>
</tr>
<tr>
<td>LBSCULT</td>
<td>0.2</td>
</tr>
<tr>
<td>Musing</td>
<td>15</td>
</tr>
<tr>
<td>Ontoweb</td>
<td>2</td>
</tr>
<tr>
<td>Salero</td>
<td>14</td>
</tr>
<tr>
<td>SEKT</td>
<td>13</td>
</tr>
<tr>
<td>Semantic Gov</td>
<td>5</td>
</tr>
<tr>
<td>SemNetMan</td>
<td>0.5</td>
</tr>
<tr>
<td>SUPER</td>
<td>15</td>
</tr>
<tr>
<td>SWWS</td>
<td>3</td>
</tr>
<tr>
<td>TRIPCOM</td>
<td>5</td>
</tr>
<tr>
<td>Semantic Service Engine</td>
<td>4</td>
</tr>
<tr>
<td>GGP Testbed</td>
<td>0.15</td>
</tr>
<tr>
<td>Multiagent Decisions (MAD)</td>
<td>12.5</td>
</tr>
<tr>
<td>ONTOESC</td>
<td>2</td>
</tr>
<tr>
<td>COG</td>
<td>2</td>
</tr>
<tr>
<td>DIP</td>
<td>18</td>
</tr>
<tr>
<td>Eastweb</td>
<td>1</td>
</tr>
<tr>
<td>E-Swan</td>
<td>0.5</td>
</tr>
<tr>
<td>Eastweb</td>
<td>1</td>
</tr>
<tr>
<td>H-techsight</td>
<td>3</td>
</tr>
<tr>
<td>Knowledge Web</td>
<td>8</td>
</tr>
<tr>
<td>M3PE</td>
<td>0.2</td>
</tr>
<tr>
<td>Nepomuk</td>
<td>16</td>
</tr>
<tr>
<td>RW2</td>
<td>0.4</td>
</tr>
<tr>
<td>SAOR</td>
<td>0.3</td>
</tr>
<tr>
<td>SEEMP</td>
<td>5</td>
</tr>
<tr>
<td>Sembiz</td>
<td>0.6</td>
</tr>
<tr>
<td>Sense</td>
<td>0.7</td>
</tr>
<tr>
<td>Swing</td>
<td>4</td>
</tr>
<tr>
<td>SystemOne</td>
<td>2.1</td>
</tr>
<tr>
<td>Interoperable EHR</td>
<td>16</td>
</tr>
<tr>
<td>TSC</td>
<td>0.3</td>
</tr>
<tr>
<td>Learning Evaluation</td>
<td>1.85</td>
</tr>
</tbody>
</table>
Making Semantic Web real.

The technological Vision
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.

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 operation system of the 21st century based on semantics
Semantic Web and Web Services

500 million users
more than 3 billion pages

Static WWW
URI, HTML, HTTP

Semantic Web
RDFS, OWL

Serious Problems in:
- information finding,
- information extracting,
- information representing,
- information interpreting, and
- information maintaining.

Making Semantic Web real.
Semantic Web and Web Services

Static

Dynamic

Web Services

UDDI, WSDL, SOAP

Bringing the computer back as a device for computation

WWW

URI, HTML, HTTP

Semantic Web

RDF, RDF(S), OWL

Bringing the Web to its full potential

Dynamic

Intelligent Web Services

Static

Semantic Web

RDF, RDF(S), OWL

Making Semantic Web real.
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
DERI – W<Triple>

W<Triple> which stands for:

- **WSMO**: A conceptual model for describing service oriented architectures
- **WSML**: A formal language for describing service oriented architectures
- **WSMX**: 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
- Ontological Role Separation
- Centrality of Mediation
- Execution Semantics
- Description versus Implementation
Objectives that a client may have when consulting a Web Service

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

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
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

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**

<table>
<thead>
<tr>
<th>Web Services</th>
<th><strong>Triple Space</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>email</td>
<td>web</td>
</tr>
</tbody>
</table>

---

**Making Semantic Web real.**
**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

**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 Architecture (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
“Ubiquitous Computing Paradigm“:
- the right service
- at the right place
- at the right time
- (at the right cost)

„magic beyond the scene“

You have been driving for 10 hours now and the weather is becoming bad. How about a stopover at a quiet hotel at the lake nearby?

Concepts – Resource Management Functionality:
Ubiquitous Computing (1)

Ubiquitous Services
- Semantics (SWIS)
- Context (Modeling & Retrieval)
- Protocol (Choreography & Orchestration)
- Syntax

GRID Services
- Full Reasoning
- Context-Awareness
- Personalization
- Mobility (Location)
- Load Distribution
- Virtual Organization
- Negotiation & Adaption
- (Functional and Non-Functional Properties)

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

Triple Space
- Negotiation & Adaption
- (Functional and Non-Functional Properties)

Semantic GRID
- 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
• 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.
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 SWS ARCHITECTURE

WSMX – Components
**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 mechanism
- 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

- 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
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 Brodie1, Christoph Bussler2, Jos de Bruijn3, Thomas Fahringer4, Dieter Fensel3,5, Martin Hepp3, Holger Lausen3, Dumitru Roman3, Thomas Strang3, Hannes Werthner6, and Michal Zaremba3,5

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