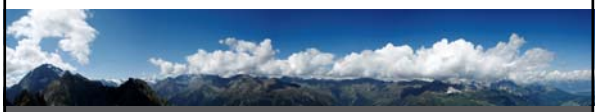
STI · INNSBRUCK

Semantic Web Services

Service Science

Lecture III – 19th March 2009

Dieter Fensel



Where are we?

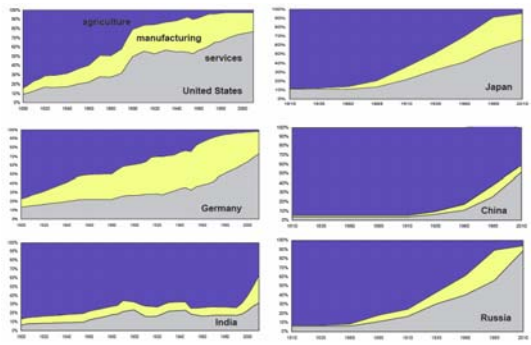
#	Date	Title
1	5th March	Introduction
2	12th March	Web Science
3	19th March	Service Science
4	26th March	Web Services (WSDL. SOAP, UDDI, XML)
5	2nd April	Web 2.0 and RESTful services
6	23rd April	WSMO
7	30th April	WSML
8	7th May	WSMX
9	14th May	OWL-S and others
10	28th May	WSMO-Lite, MicroWSMO
11	4th June	SWS Use Cases
12	18th June	seekda: the business point of view
13	25th June	Mobile services
14	2nd July	Exam Preparation

Outline

- What is a Service?
- What is Service Science?
- The pillars of Service Science
  - Service Oriented Architecture
  - Semantic Enables Service Oriented Architecture

What is a Service?

The Rise of the Service Economy



Services

- Wide range of services:
  - Communication Service
  - Ticket Reservation Service
  - Transport Service
  - Information Service
  - Finance Service
  - E-government Service
  - …

But what is a Service?

What is a service?

Main Entry: ser·vice
Function: *noun*
Etymology: Middle English, from Anglo-French *servise,* from Latin *servitium* condition of a slave, body of slaves, from *servus* slave
1 **a:** the occupation or function of serving <in active *service*> **b:** employment as a servant <entered his *service*>
2 **a:** the work performed by one that serves <good *service*> **b:** help , use , benefit <glad to be of *service*> **c:** contribution to the welfare of others **d:** disposal for use <I'm entirely at your *service*>
3 **a:** a form followed in worship or in a religious ceremony <the burial *service*> **b:** a meeting for worship —often used in plural <held evening *services*>
4**:** the act of serving: as **a:** a helpful act <did him a *service*> **b:** useful labor that does not produce a tangible commodity —usually used in plural <charge for professional *services*> **c:** serve
5**:** a set of articles for a particular use <a silver tea *service*>
6 **a:** an administrative division (as of a government or business) <the consular *service*> **b:** one of a nation's military forces (as the army or navy)
7 **a:** a facility supplying some public demand <telephone *service*> <bus *service*> **b:** a facility providing maintenance and repair <television *service*>
8**:** the materials (as spun yarn, small lines, or canvas) used for serving a rope
9**:** the act of bringing a legal writ, process, or summons to notice as prescribed by law
10**:** the act of a male animal copulating with a female animal11**:** a branch of a hospital medical staff devoted to a particular specialty <obstetrical *service*>

Merriam-Webster Online, http://www.m-w.com

What is a service?

- For different people the term Service has different meaning:
- **In Computer Science:**
  - the terms service and Web service are often regarded as interchangeable to name a software entity accessible over the Internet.
  - a service is seen software system designed to support interoperable machine-to-machine interaction over a network

What is a service?

- **In Business and Economics:**
  - a service is seen as a business activity that often results in intangible outcomes or benefits
  - a service is the non-material equivalent of a good. Service provision has been defined as an economic activity that does not result in ownership, and this is what differentiates it from providing physical goods.
  - a process that creates benefits by facilitating either a change in customers, a change in their physical possessions, or a change in their intangible assets.

Service vs. Web Service

- Service
  - A provision of value in some domain (not necessarily monetary, independent of how service provider and requestor interact)
- Web Service
  - Computational entity accessible over the Internet (using Web Service Standards & Protocols), provides access to (concrete) services for the clients.

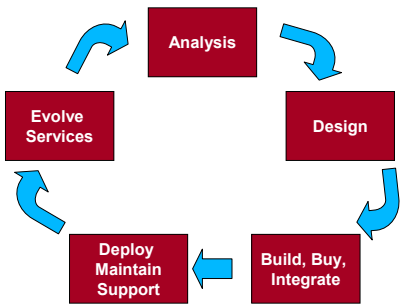
Service properties

- *Functional*
  - contains the formal specification of ***what*** exactly the service can do.
- *Behavioral*
  - ***how*** the functionality of the service can be achieved in terms of interaction with the service and as well in terms of functionality required from the other Web services.
- *Non-functional properties*
  - captures constraints over the previous mentioned properties

Service related tasks

- **Discovery:** "*Find services that matches to the service requester specification*" .
- **Selection and Ranking**: "*Choose the most appropriate services among the available ones*"
- **Composition**: "*Assembly of services based in order to achieve a given goal and provide a higher order of functionality*".
- **Mediation**: *"Solve mismatches among domain knowledge used to describe the services, protocols used in the communication, data exchanged in the interaction (types used, and meaning of the information) and business models of the different parties".*
- **Execution**: "*Invocation of a concrete set of services, arranged in a particular way following programmatic conventions that realizes a given task*".
- **Monitoring**: "*Supervision of the correct execution of services and dealing with exceptions thrown by composed services or the composition workflow itself*".
- **Handover**: "*Replacement of services by equivalent ones, which solely or in combination can realize the same functionality as the replaced one, in case of failure while execution*".

Service Lifecycle



What is Service Science?

Definition

- "Service Science, Management and Engineering (SSME) is a new multi-disciplinary research and academic effort that integrates aspects of established fields such as computer science, operations research, engineering, management sciences, business strategy, social and cognitive sciences, and legal sciences."
  IBM's definition
- "Service Science, Management, and Engineering (SSME) is an interdisciplinary approach to the study, design, and implementation of service systems – complex systems in which specific arrangements of people and technologies take actions that provide value for others."
  Wikipedia's definition

Service Science: multidisciplinary

- A multidisciplinary science influenced by
  - Computer science
  - Cognitive science
  - Economics
  - Organizational behavior
  - Marketing
  - Operations research
  - Policy and Law
  - …

The Goal of Service Science

- To provide concepts, methods and techniques to understand and engineer service based systems
- To ensure the social benefit of service based systems

Service Science

- Service Science is recognized as a very important emerging science:
  - **NESSI** (Networked European Software and Services Initiative) group in the **EU**: http://www.nessi-europe.com/Nessi/Workinggroups/HorizontalWorkingGroups/ServicesSciences/tabid/244/Default.aspx
  - **IBM**: Service Science, Management and Engineering http://www.ibm.com/university/ssme
  - **HP**: Center for Systems and Service Sciences http://www.services-sciences.org/
  - **Oracle:** Service Research and Innovation Initiative http://www.thesrii.org/

Service Oriented Architecture

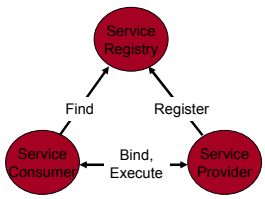
What is Service Oriented Architecture (SOA)?

- "A service-oriented architecture is essentially a collection of services. These services communicate with each other. The communication can involve either simple data passing or it could involve two or more services coordinating some activity. Some means of connecting services to each other is needed."[1]
- "Service-oriented architecture (SOA) provides methods for systems development and integration where systems group functionality around business processes and package these as *interoperable services*. An SOA infrastructure allows different applications to exchange data with one another as they participate in business processes. SOA separates functions into distinct units, or services, which developers make accessible over a network in order that users can combine and reuse them in the production of business applications "
  Wikipedia

[1] http://www.service-architecture.com/web-services/articles/service-oriented_architecture_soa_definition.html

What is Service Oriented Architecture (SOA)?

- Is not a computing architecture but a style of programming
- An SOA application is a composition of services
- A "service" is the building block/ unit of an SOA
- Services encapsulate a business process
- Service Providers Register themselves
- Service use involves: Find, Bind, Execute
- Most well-known instance is Web Services



SOA Actors

- Service Provider
  - From a business perspective, this is the owner of the service. From an architectural perspective, this is the platform that provides access to the *service*.
- Service Registry
  - This is an information space of service *descriptions* where service providers publish their services and service requesters find services and obtain binding information for services.
  - Allows service consumers to locate service providers that meet required criteria
- Service Consumer
  - From a business perspective, this is the business that requires certain function to be fulfilled. From an architectural perspective, this is the *client* application that is looking for and eventually invoking a service.

Principles of SOA

- Formal contract
- Loose coupling
- Abstraction
- Reusability
- Autonomy
- Statelessness
- Discoverability
- Composability

SOA Lifecycle

- Initiate
  - decide which business function and underlying processes
- Develop Roadmap
  - process for conducting an SOA assessment
  - developing the SOA principles
  - defining the reference architecture (future state)
  - making the transition from the current situation to the future state
- Execute Plan
  - Projects
  - Organisation and Governance
- Review and update roadmap

Self-* Properties

- Most service architectures aim for „self *“ properties to reduce management load by design:
  - Self-Configuration
  - Self-Organization
  - Self-Healing
  - Self-Optimization
  - Self-Protection

Self-Configuration

- Service architectures comprise of a huge amount of different components (services and hardware). Configuration is a challenging task in such environments.
- The idea of **self-configuration**is the adoption of the self-organization and fully distributed cooperation capabilities known from groups with cooperative social behavior which collaborate to solve a problem. Every member of the group can decide which part of the problem it can solve and which “QoS” it can provide.

Self-Organization

- A system is **self-organizing** if it automatically, dynamically and autonomously adapts itself to achieve global goals more efficiently under changing conditions.

Self-Healing

- The task of **self-healing** is to assure that a system meets some defined conditions as far as possible, i.e. to guarantee that all services running in the framework stay available, even in the case of partial outages in the system.

Self-Optimization

- The **self-configuration** is responsible to find a good distribution of the services in terms of the given resources of the service description. The target of the **self-optimization** is to distribute the services of the application in a way that the considered resources are utilized evenly.
- A typical approach is to find an adequate configuration at the beginning and to optimize the application during runtime.

Self-Protection

- **Self-protection** techniques cope with intentionally or unintentionally malicious peers or services in a framework. The behave as the "immune system" of a service framework as they are permissive to good natured services and messages but can detect appearing malicious events.

SOA Benefits

Business Benefits
- Focus on Business Domain solutions
- Leverage Existing Infrastructure
- Agility

Technical Benefits
- Loose Coupling
- Autonomous Service
- Location Transparency
- Late Binding

Semantically Enabled Service Oriented Architecture

SESA (1)

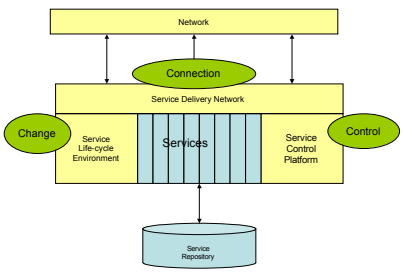
- Currently, computer science is in the next period of abstraction.
- A generation ago we learnt to abstract from hardware and currently we learn to abstract from software in terms of SERVICE oriented architectures (SOA).
- It is the service that counts for a customer and not the specific software or 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.

SESA (2)

- Service-oriented architectures will become quickly the leading software paradigm.
- However, SOAs will not scale without signification 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

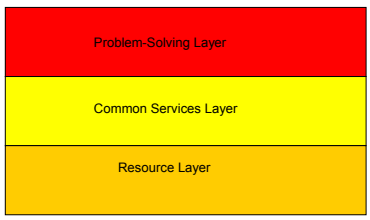
SESA Manifesto (M. Brodie et al.)

- *Change, Connection and Control*



M. Brodie et al. – SESA Manifesto

Three Layers of SESA



Problem solving layer

- This layer turns a service-oriented architecture into a domain specific problem-solving environment.
- It represents the transparent interface to the user(s), where all computing resources are turned into or expressed as services
- Supports the full set of operations from an e-commerce framework: information negotiation, etc.
- Provides clear separation between business/process logic on one hand and the common service layer

Common Services Layer

- Provides an adaptive ***execution environment*** and the supporting infrastructure that maps the problem descriptions generated at the Problem Solving Layer to the services that can solve the problems.
- Existing architectures and standards from Web service and Grid areas (e.g. OGSA, WSRF, WSDL) which operate only at a syntactic level are semantically enriched and integrated into this layer.
- Semantically enrichment of SOAs that implement the Common Service Layer capabilities will help to automate: service discovery, service adaptation, negotiation, service composition, etc.
- This layer could be implemented using the W<Triple> technology

Resource Layer

- Responsible for actual execution of applications.
- All tasks that involve resources such as resource discovery, selection and negotiation for advanced or on-the-fly reservation of resources are supported and implemented in this layer.
- Covers the deployment and provisioning of physical resource (e.g. computers, data servers, and networks, usually connected into a Grid) and logical resources (e.g. application components or common services).
- This layer may relay on two prominent and widely discussed areas that deal with distributed resources in the context of service oriented computing are Ubiquitous Computing and Grid Computing

W<Triple>

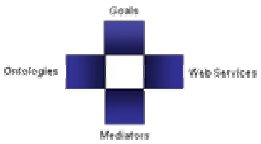
W<Triple> which stands for:

- ***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

W<Triple>: WSMO

Objectives that a client wants to achieve by using Web Services

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

More on Lecture 6

W<Triple>: WSML

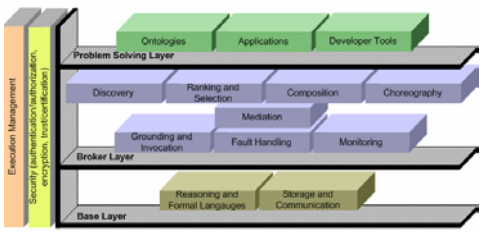
A set of concrete languages for the various tasks:

- 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)
  - or data (data mediation)

More on Lecture 7

W<Triple>: WSMX

WSMX: The **Web Service EXecution Environment**
- A service oriented architecture.
- Reference implementation of SESA and WSMO



More on Lecture 8

W<Triple>: Triple Space Computing

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

W<Triple>: Triple Space Computing

- 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

W<Triple>: Triple Space Computing

- 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.

Service Science: why this matter?

- Service oriented architectures are an arising software paradigm with big potential for the IT market.
- Bringing service orientation to its full potential requires its combination with semantics to mechanize important aspects and make it scalable.

Next Lecture

#	Date	Title
1	5th March	Introduction
2	12th March	Web Science
3	19th March	Service Science
4	**26th March**	**Web Services (WSDL. SOAP, UDDI, XML)**
5	2nd April	Web 2.0 and RESTful services
6	23rd April	WSMO
7	30th April	WSML
8	7th May	WSMX
9	14th May	OWL-S and others
10	28th May	WSMO-Lite, MicroWSMO
11	4th June	SWS Use Cases
12	18th June	seekda: the business point of view
13	25th June	Mobile services
14	2nd July	Exam Preparation

Questions?

STI · INNSBRUCK