Web Services og SOA – hvordan vi ser for oss bruk i fremtiden

NorStella SOA- og Web Services-seminar 30.05.2008

Brian Elvesæter
brian.elvesater@sintef.no
Outline

- Service-oriented architecture (SOA)
- Interoperabilitet og systemintegrasjon
- (Semantic) Web services
- Forskning og utvikling på SOA
Service-oriented architecture (SOA)
Different kinds of architectures

- Enterprise architecture
- Integration architecture
- Business architecture
- Knowledge architecture
- Conceptual architecture
- Logical architecture
- Functional architecture
- Service-oriented architecture
- Realisation architecture
- Information architecture
- ICT architecture
- Web services architecture
- Architecture framework
Enterprise architecture (EA) is the practice of applying a method for describing a current and/or future structure and behaviour for an organization's processes, information systems, personnel and organizational sub-units, so that they align with the organization's core goals and strategic direction.
- Holistic view of the enterprise and all its important assets.

Service-oriented architecture (SOA) is a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains. [OASIS 2006]
- Architectural style for designing (technical) systems.

Web services architecture (WSA) intends to provide a common definition for understanding Web services. A Web services architecture involves many layered and interrelated technologies. [W3C 2004]
- A set of enabling Web technologies for implementing software systems.
SOA governance with EA

- Architecture is a strategic tool
  - not just high-level design
  - Architecture goes beyond ICT
  - Enterprise architecture is a key component of the IT governance process at any organization of significant size.

- Stability and flexibility
  - Seem to be contradictory, but a good architecture facilitates change!

- Communication with stakeholders
  - architects, managers, customers, engineers, …

- Analysis
  - impact-of-change
  - cost and performance

- Enterprise Architecture (EA) is a generic, abstracted and aggregated representation of the core structures and competences of an enterprise.
- EA supports laying out the main characteristics of the enterprise to be analysed and agreed before detailed technical design is started.
- It is shared and discussed enterprise-wide between all stakeholders as a common description forms, functions and features, components, properties and relationships.
Role of enterprise architecture

Describing coherence

OASIS Reference Model for Service Oriented Architecture 1.0

- OASIS
- Abstract framework.
  - Understanding significant entities and relationships between them within a service-oriented environment.
  - Development of consistent standards or specifications supporting service-oriented environment.
- Based on unifying concepts of SOA and may be used by
  - architects developing specific service-oriented architectures
  - in training and explaining SOA.
- Reference model not directly tied to any standards, technologies or other concrete implementation details
- Provide a common semantics that can be used unambiguously across and between different implementations.
- The reference model focuses on the field of software architecture.
What is an SOA

- **Service-oriented architecture** (SOA) is a paradigm for organizing and utilizing distributed **capabilities** that may be under the control of different ownership domains.

- Visibility, interaction, and effect are key concepts for describing the SOA paradigm.
  - **Visibility** refers to the capacity for those with needs and those with capabilities to be able to see each other.
  - Whereas visibility introduces the possibilities for matching needs to capabilities (and vice versa), **interaction** is the activity of using a capability.
  - The purpose of using a capability is to realize one or more **real world effects**. At its core, an interaction is “an act” as opposed to “an object” and the result of an interaction is an effect (or a set/series of effects).
Interoperabilitet og systemintegrasjon
Rationale for interoperability

- Interoperability is the key to increase competitiveness of enterprises.
- “Enterprise systems and applications need to be interoperable to achieve seamless operational and business interaction, and create networked organizations” – European Group for Research on Interoperability, 2002

Application integration license revenue

System implementation budget

(Source: the Yankee Group 2001)

The cost of non-interoperability are estimated to 40% of enterprises IT budget.
Holistic approach to interoperability

Interoperability (def.) is “the ability of two or more systems or components to exchange information and to use the information that has been exchanged” – IEEE Standard Computer Dictionary

To achieve meaningful interoperability between enterprises, interoperability must be achieved on all layers:

- **Business layer**: business environment and business processes
- **Knowledge layer**: organisational roles, skills and competencies of employees and knowledge assets
- **ICT layer**: applications, data and communication components
- **Semantics**: support mutual understanding on all layers
Motivation

Enterprise
- Challenges
  - Business agility
  - Flexibility and adaptability
- Enterprise architecture frameworks
  + Holistic approach
  + Different views of an enterprise as related (visual) knowledge models
  - Current enterprise architectures are only blueprints

ICT
- Challenges
  - Inflexible and difficult to adapt
  - Enterprise application integration (EAI)
- Service-oriented architecture (SOA)
  + Architectural style
  + Loosely coupled systems
  + Horizontal integration between different business domains
  + Use case oriented service composition
  +/- Web services (enabling technology)

Requirements
- Enterprises require operational enterprise architectures
- ICT solutions must be designed to be inherently interoperable
# Integrasjon vs. interoperabilitet

<table>
<thead>
<tr>
<th>Area</th>
<th>Integration</th>
<th>Interoperability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consistency of objectives</td>
<td>Consistency of local objectives to global ones</td>
<td>No consistence between local and global objectives</td>
</tr>
<tr>
<td>Type of coupling</td>
<td>Tightly coupled, components interdependent</td>
<td>Loosely coupled, components independent</td>
</tr>
<tr>
<td>Solution approach</td>
<td>Uniformisation of languages, methods, tools etc.</td>
<td>Network of services with identity and diversity preserved.</td>
</tr>
<tr>
<td>Organisational structure</td>
<td>Intra-enterprise, fusion, re-structuration, etc.</td>
<td>Inter-enterprise, virtual enterprise, etc.</td>
</tr>
</tbody>
</table>
History of integration

1950 – 2007: Integration = develop then integrate
  - 1950s-1970s: Simple, manual integration
  - 1970s-1980s: Distributed Computing
    - Applications (interoperation)
    - Databases (integrate)

1990s: Business Driven Integration – concepts, technologies, and tools – increased automation, internet-based computing
  - Concepts: Workflows, Processes, Web,
  - Integration solutions blossom (diverge): ETL, EAI, BPM, …

2000: SOA Emerges
  - 2000: Web services
  - 2003: Integration solution evolution accelerates, vendor chaos ensues
  - 2005: Growth in all integration categories
SOA and integration/interoperability

- Fundamental change for integration: X <-> Y
  - Pre-SOA: outside, after development
  - Post-SOA: inside, integral part of development / computational model
    - Towards loosely coupled interoperable systems
    - From static composition to (semi)-automated compositions of services
    - Process integration/interoperability

- Consequences
  - How should integration/interoperability be done?
    - Services under different ownership domains may fail
    - Monitoring and service contracts
  - Innovation and experience
  - Competition, expansion, consolidation
Application architecture vs. SOA
Integration/interoperability in SOA

- 2007 – 2012: Integration = dominant programming model
  - 2001-2010: Wrapping
  - 2005-2010: Re-Engineering
  - 2006-2008: Consolidation
  - 2006-2008: Research on Semantic SOA
  - 2007-2012: Emergence of SOA Platforms and Solutions
  - 2008-2012: Problem Solving Era: IT/integration relegated to low level function
SOA platform consolidation

- Data and information integration ➢ **Information Fabric**
  - EII: Enterprise information integration
  - ETL: Extract, transform and load
- Application integration ➢ **Integration Suite**
  - EAI: Enterprise application integration
  - B2Bg: Business-to-business gateway
  - ESB: Enterprise service bus
- Applications and Processes ➢ **Business Process Management Suite**
  - BPM: Business process management
  - B2Bi: Business-to-business integration
- Enterprise workplace ➢ **Interaction Platform**
SOA Framework: Process + Applications + Data

Interaction Platform

Business Process Management Suite

Information Fabric

Virtualized Data

Enterprise / Information Workplace

User interaction module

User interaction module

User interaction module

Business process and workflow definitions

Business service

Business service

Business service

Business service

Business service

Business service

Utility services

Composite Applications

Business Unit Services

Enterprise Services

Business service flow

Information Fabric

Databases

Files

Devices

Databases
Goal: Composite applications


Extensions: Adapter, collaboration, analysis, reporting, development, monitoring, contracts, SOA standards, …
Business process management suite & interaction services

- **Goal:** Continuous process improvement
- **Components:** BPM
  - human-centric: people-intensive processes
  - Integration-centric: system-intensive processes
Information fabric services

- Goal: Holistic view of data (information virtualisation)
- Components: DBMS, EII + ETL + replication
- Extensions: Distributed meta-data repository, distributed data access, integrated data management
(Semantic) Web services
The waves of client/server technology

First Wave
- File Servers
- Database Servers
- Groupware
- TP Monitors

Second Wave
- Distributed Objects
- OMG CORBA
- COM/OLE
- Web/Internet
- Java

Third Wave
- Server-side components
- J2EE/EJB
- COM+
- Corba
- Comp
- Agents, P2P
- FIPA
- MDA
- Web Services
- .Net

Fourth Wave
- Service-oriented architecture
- SOAP, XML
- WSDL/WSFL

Fifth Wave
- Agents, FIPA
- Grid
- P2P

Sixth Wave
- WEB 2.0
- Semantic Web Services
Web service

- Web service
  - “Applications identified by a URI, whose interfaces and bindings are capable of being defined, described and discovered as XML artefacts. A Web service supports direct interactions with other software agents using XML-based messages exchanged via Internet-based protocols.” (W3C)
  - http://www.w3.org/

- SOA ~ architectural style
- Web services stack ~ technology/protocol standards
- SOA =/= Web services
Web services architecture

- Web services can be used to implement service-oriented solutions.
- They adhere to the set of roles and operations specified by the service-oriented model.
- They have also managed to establish a standardized protocol stack.
WS-* stack to-be

- Simplified version of the to-be WS-* stack
  - Families of related specs not expanded
  - Competing spec families not shown
  - “Historical” or abandoned specs not shown
WS-* stack as-is

- Complete version of the as-is WS-* stack
  - The 3 widely-accepted specs today are the same as 5 years ago
  - BPEL and WS-Security is gaining momentum
  - Orchestration, discovery and brokering do not exist in today’s world
Next Generation of the WWW

Dynamic Web Services
- UDDI, WSDL, SOAP
- Semantic Web Services

Static
- WWW
  - URI, HTML, HTTP
- Semantic Web
  - RDF, RDF(S), OWL

Syntactic

Semantic
Semantic Web services

- Semantic Web & Ontologies
  - ontology languages
  - reasoning infrastructures
  - semantic data integration

- Semantic Web Services
  - semantic WS annotation
  - SWS technologies: automated discovery, composition, mediation, selection, compatibility analysis
  - automated WS execution
  - Web Services & SOA
Goal-driven Web Service Usage

Client
objective / problem to be solved

Goals
formal objective description

discovery, composition, mediation

discovery, composition, mediation

Semantics / SWS
Ontology
SWS description
Mediator

Execution
Web Services & Resources

Web Service
Internet

Discovery
- find candidate WS to solve a Goal

Selection & Ranking
- select best candidate / determine a priority list

Composition
- combine several WS to solve a Goal

Behavioral Compatibility
- ensure that interaction can take place

Mediation
- resolve & handle possibly occurring heterogeneities

Execution
- automatically invoke & consume WS to solve a Goal
WSMO Web Service Description

- complete item description
- quality aspects
- Web Service Management

Non-functional Properties

- Advertising of Web Service
- Support for WS Discovery

Capability

- functional description

Web Service Implementation
(not of interest in Web Service Description)

- Advertising of Web Service
- Support for WS Discovery

Choreography --- Service Interfaces --- Orchestration

client-service interaction interface for consuming WS
- external visible behavior
- communication structure
- ‘grounding’

DC + QoS + Version + financial

realization of functionality by aggregation
- functional decomposition
- WS composition

WS
WS
WS
Forskning og utvikling på SOA
Trends

- Model-driven architecture (MDA)
  - Mer utstrakt bruk av modeller
  - Kombinasjon av ulike formalismer (DSLer) for å beskrive ulike aspekter av IT-arkitektur/system
  - Standardisering av service metamodel (UPMS)
- Konsolidering av SOA-plattformer
- Semantiske teknologier
  - Semantic Web services
- Rikere portalteknologi og tjenester ut mot sluttbrukere
  - Rikere brukergrensesnitt
  - Sluttbrukere vil ha anledning til å sette sammen egne tjenester
- Tjeneste-paradigme for å levere/tilby programvare
  - Ser på Weben som en plattform i seg selv
  - Software-as-a-Service (SaaS)
  - Software-as-a-Service Utility (SaaSU)
Forskningsaktiviteter

- **COIN (Enterprise Collaboration and Interoperability)**
  - EU-prosjekt med fokus på utvikling av open-source Enterprise Collaboration (EC) og Enterprise Interoperability (EI) services

- **SHAPE (Semantically-enabled Heterogeneous Service Architecture and Platforms Engineering)**
  - EU-prosjekt med fokus på utvikling av modell-baserte verktøy og metoder for tjenesteorienterte arkitekturer
  - [http://www.shape-project.eu/](http://www.shape-project.eu/)

- **NESSI (Networked European Software and Services Initiative)**
  - Europeisk teknologiplattform for SOA

- **SOA i praksis**
  - Norsk ressursnettverk om SOA
COIN main objectives

1. To design and develop a pervasive, adaptive service platform to host Baseline and Innovative COIN services for EI and EC and make them available under innovative on-demand, utility-oriented business models (i.e. the SaaS-U model) to European enterprises (and SMEs in particular) for running their business in a secure, reliable and efficient way.

2. To consolidate and stabilize the ICT results of both EC and EI FP6 research into some Baseline Services which constitute the service foundations for COIN.

3. To further enlarge, extend and improve the Baseline Services, by developing other more Innovative Services in the EC and EI fields, which could take into account the most recent and promising technology challenges (in the field of Web 2.0, semantic web, space computing) and put them at service of EC and EI purposes.

4. To represent a pathway to convergence for these two fundamental research streams: EI and EC, by integrating in the same project the most prominent stakeholders of the two research fields coming both from industry and from universities and research centers.

5. To demonstrate, experiment, trial and assess the project results into realistic industrial scenarios offered by our 6 test cases in Aeronautics (Aeronautic Cluster of Andalusia, Spain), Automotive (the Automotive Cluster of Slovenia), Aerospace (the Lazio Connect virtual enterprise network Italy), Pulp & Paper (the Poyry consultancy service providers), Healthcare (the VEN network in U.K.) and ICT (the Hungarian Association of ICT companies).
UPMS standardisering i SHAPE

- **UML Profile and Metamodel for Services (UPMS)**
  - support the activities of service modelling and design
  - fit into an overall model-driven development approach
    - platform independent
  - modelling requirements for service-oriented architectures
    - specification of systems of services
    - specification of individual service interfaces
    - specification of service implementations
    - specification of service contract
    - ...

---

UPMS Agents

Semantic Web Services

Service Variability

Flexible Business Models

Web Services

Grid

P2P

Semantic Web Services

Heterogeneous Platforms
UPMS metamodel overview
UPMS service metamodel
(abstract syntax)
UPMS service model example (concrete syntax)

```plaintext
<<interface>>
Purchasing
+ processPurchaseOrder (customerInfo: Customer, purchaseOrder: PurchaseOrder): Invoice

<<interface>>
Scheduling
+ requestProductionScheduling (customerInfo: Customer, purchaseOrder: PurchaseOrder)
+ sendShippingSchedule (schedule: Schedule)
```

```
<<participant>>
OrderProcessor

<<service>> purchasing: Purchasing

<<requisition>> invoicing: InvoicingService
```

```
<<service>> shipping: ShippingService

<<requisition>> scheduling: Scheduling
```

```
InvoiceProcessing

Invoicing

Invoice

Scheduling

Shipping

ScheduleProcessing
```
NESSI Open Framework (NEXOF)

The overall ambition of NESSI is to deliver NEXOF, a coherent and consistent open service framework leveraging research in the area of service-based systems to consolidate and trigger innovation in service-oriented economies. NEXOF consists of three core elements:

- **NESSI Open Reference Model** includes the conceptual model of the core elements that enable service-based ecosystems and their relationship as well as underlying rules, principles and policies which lead to interoperable implementations. Core elements include business dynamics, development environment and operational environment.

- **NESSI Open Reference Architecture** addressing definition and selection of innovative architectural styles and patterns based on the reference model. Aims to be a standardized open reference architecture for services and components, and also to some extent processes which corresponds to a significant advancement of today’s service-oriented architectures. This will include the definition of the infrastructure requirements.

- **NESSI Open Reference Implementation** taking the responsibility to deliver to the community at large with the implementation of the NEXOF concepts and approaches where the openness, built on open source and open standards.
SOA i praksis

SOA i praksis-nettverket er tverrfaglig sammensatt av brukermiljøer, kunnskapsmiljøer innen virksomhetsmodellering og -analyse, programvaremiljøer, SOA-miljøer, internasjonale IT-leverandører og representanter for nasjonal og internasjonal infrastruktur for både offentlige og private virksomheter.

Ressursnettverket har følgende målsettinger:

- Etablere status og identifiser SOA beste praksis basert på case-analyser.
- Etablere basis for å utnytte resultater fra pågående SOA-forskning, gjennom formidling av SOA problemstillinger, forskningsresultater og innovasjon (for EU og norske prosjekter).
- Definere nasjonale forsknings utfordringer innen "SOA i praksis" i forhold til temaet kommuniserende organisasjoner.
- Generere prosjekter– i form av søknader til VERDIKT samt EU og andre finansieringskilder.
Spørsmål?