Enterprise architecture frameworks with semantic models as a foundation for complex networked operations

SoaML – Service modeling
Agenda

(I) Enterprise Architecture, TOGAF, UPDM (Arne, Ulf, Dima)
- Zachman, TOGAF, MODAF/DODAF/NAF, MDA, UPDM - Arne
- Saarstahl SHAPE - Dima
- European ATM/SESAR - Ulf

(II) INFORMATION and ONTOLOGY MODELING (UML/ER, ODM/OWL with examples/tools) Arne (Ulf, Dima)
- Conceptual Modeling, Information Modeling, Ontologies - Ulf and Arne
- ODM with OWL for semantic modeling (WSMT) - Dima

(III) PROCESS MODELING (EPC/BPMN with examples/tools) (Dima)
- ARIS/EPC (Event-Driven Process Chains) - Dima
- BPMN (Business Process Modeling Notation) - Dima

(IV) SERVICE MODELING and Interoperability (SoaML with examples) (Arne)
- SoaML (Servic oriented architecture Modeling Language) - Arne
- Semantic annotations, SAWSDL, from existing system specifications to an ontology can support semantic interoperability - Arne
**Business Focused SOA Using Model Driven Architecture**

**MDA Terms**
- Business Concerns
- Business Model
  - Enterprise Services (e-SOA)
  - Roles, Collaborations & Interactions
  - Process & Information
- Logical System Model
  - Technology Services (t-SOA)
  - Components
  - Interfaces, Messages & Data
- Technology Specification
  - JMS, JEE, Web Services
  - WSDL, BPEL, XML Schema

**Refinement & Automation**

**Line-Of-Sight**
CIM – PIM - PSM

**SoaML-SHA**

- Core
  - Service Variability
  - PIM4
    - WS-A
  - PIM4
    - SWS
  - PIM4
    - Agents
  - P2P/Grid/Components

**WSDL, WSMO, OWLS, JACK, JADE, JXTA, OGSA, J2EE, CORBA**

**J2EE, NetWeaver, .Net, ...**

- **BPMN**
- **BPDM**
- **BMM**
- **EPC**
- **...**
BMM with MeansRealizations
Service oriented architecture Modeling Language (SoaML) - Specification for the UML Profile and Metamodel for Services (UPMS)

Revised Submission
OMG document: ad/2008-08-04

Submitters
Adaptive
Capgemini
EDS
Fujitsu
Fundacion European Software Institute
Hewlett-Packard
International Business Machines
MEGA International
Model Driven Solutions
Rhysome
Softeam

Supporters
BAE Systems
STI/University of Innsbruck
DFKI
Everware-CBDI
France Telecom R&D
General Services Administration
VisioPoint
MID GmbH
NKUA – University of Athens
Oslo Software
SINTEF
THALES Group
University of Augsburg
Wilton Consulting Group

Primary Contact:
Arne J. Berre, SINTEF
e-mail: Arne.J.Berre@sintef.no

Find the document here:

Revised submission per November 10th, 2008

See also: www.soaml.org
SoaML UML Profile for Service Interface and Participants
A ServicesArchitecture (or SOA) is a network of participant roles providing and consuming services to fulfill a purpose. The services architecture defines the requirements for the types of participants and service realizations that fulfill those roles.
Compound services
A ServicePoint is the offer of a service by one participant to others using well-defined terms, conditions and interfaces. A ServicePoint defines the connection point through which a Participant offers its capabilities and provides a service to clients. A ServicePoint is a mechanism by which a provider Participant makes available services that meet the needs of consumer requests as defined by ServiceInterfaces, Interfaces and ServiceContracts. A ServicePoint is represented by a UML Port on a Participant stereotyped as a «ServicePoint», 

```
+ requestProductionsScheduling (customerInfo: Customer, purchaseOrder: PurchaseOrder)
+ sendShippingsSchedule (schedule: Schedule)
```
Service Data

- Structural information exchanged between service consumers and service providers
- Attachments for opaque information
- Usage semantics make no assumptions with regard to global synchronization, control, or shared address spaces
Supply Chain of Saarstahl AG

Sales Department → Planing Department → Steelworks Völklingen → Semi-finished product inventory → Rolling mills Nauweiler → Rolling mills Burbach → Rolling mills Neunkirchen

Tech. Inspection
Participants

- **Order Agent**
  - Manages its working plan throughout production stages
  - Owns all necessary information

- **Sales Department**
  - Input of ordering service is forwarded to sales department
  - Creates instances of Order agents from this input
  - Can contact custumer

- **Technical Inspection**
  - Offers service to check if there is a feasible working plan for some order

- **Planning Department**
  - Provides planning instructions that cannot be automated or are not yet automated inside the Order agent
Participants ctd.

- **Steelwork**
  - Offers service to request melting capacity
- **Rolling Mills**
  - Offer several information providing services (e.g. rolling date)
  - Service for detailed scheduling inside rolling group
- **Semi-finished Product Inventory**
  - Manages inventories and semi-finished material
  - Keeps track of bindings: Order to material
PIM Modelling
Service Architecture ~ Service Contract

18.05.2009
Arne Jørgen Berre, SINTEF, Norway
Ulf Larsson, LFV, Sweden
Dima Panfilenko, DFKI IWi, Germany
PIM Modelling
Set of Service Contracts
PIM Modelling
Participant Architecture

[Diagram of PIM modelling showing relationships between components and services]

18.05.2009

Arne Jørgen Berre, SINTEF, Norway
Ulf Larsson, LFV, Sweden
Dima Panfilenko, DFKI IWi, Germany
PIM Modelling
Abstract Saarstahl Architecture
Information Modelling with UML/(SoaML) class models
A term in the vocabulary represents a **class** of things to be described.

Entities may be described as having a unique **identity**.

A relation between terms is described by an **association** between classes.

**Attributes** specify descriptive information having simple types.

A class may be **specialized** into sub-classifications.

This means “zero or more”

This indicates a compositional (as opposed to referential) association.

This means “one or more”

**Bill**

+ bills sent
  + bill
    + principal amount: Amount
    + interest amount: Amount
    + shipping charge: Amount
    + administrative costs amount: Amount
    + penalties amount: Amount
    + total amount: Amount
    + issue date: Date Time
    + due date: Date Time
    + bill number: Numeric
    + bill type: Code
    + terms: Code

+ bills received
  + bill

**Received Bill**

- (payer is self)

**Sent Bill**

- (payee is self)

**Bill Line Item**

- containing bill
  - (ordered) 1..*

**Party**

- + payee
- + payer

**Bill Line Item**

- detail type: Code
- unit price: Amount
- quantity: Amount
- extended price: Amount

**Billed Receivable**

- 1..*

This is a constraint that defines the sub-classification.

An un-shaded class is not detailed on this diagram.
Integrating the Information Model with SOA

The process model describes how business activities are (or are to be) carried out.

The information model details the vocabulary of the business entities and transactions used in the process model.

Workflow

Business transaction

Activities

Implicit memory of business information

State changes due to the activities
Everything as a service
CIM – PIM - PSM

CIM
Business Models

SoaML-SHA
Core
Service Variability

SoaML

PIM4
WS-A
PIM4
SWS
PIM4
Agents
P2P/Grid/
Components

WSDL, WSMO, OWLS, JACK, JADE, JXTA, OGSA, J2EE, CORBA

J2EE, NetWeaver, .Net, …

PIM
System Models

PIMs for different
Architectural Styles

PSM
Implementation Models

Realization Technologies
Clouds — everything as a service

- Infrastructure-as-a-Service
- Security-as-a-Service
- Integration-as-a-Service
- Database-as-a-Service
- Information-as-a-Service
- Process-as-a-Service
- Application-as-a-Service
- Management/Governance-as-a-Service
- Testing-as-a-Service

Source: David S. Linthicum
ENVISION use case: Oil spill modelling as chained services
Semantic Web and
Semantic Web Services

OWL-S
WSMO
SAWSDL

Semantic Days 2009, May 18th-20th, Stavanger, Norway
Semantic web service technologies

- OWL-S (was DAML-S, US)
- WSMO (Europe, DERI, STI, OASIS)
- WSDL-S (basis for SAWSDL)
- SAWSDL (W3C standard)
OWL-S Ontology

- OWL-S is an OWL ontology to describe Web services
- OWL-S leverages on OWL to
  - Support capability based discovery of Web services
  - Support automatic composition of Web Services
  - Support automatic invocation of Web services

"Complete do not compete"

- OWL-S does not aim to replace the Web services standards rather OWL S attempts to provide a semantic
OWL-S Upper Ontology

- Mapping to WSDL
  - communication protocol (RPC, HTTP, ...)
  - marshalling/serialization
  - transformation to and from XSD to OWL

- Control flow of the service
  - Black/Grey/Glass Box view
  - Protocol Specification
  - Abstract Messages

- Capability specification
- General features of the Service
  - Quality of Service
  - Classification in Service taxonomies

- Resource
  - provides (what it does)

- ServiceProfile
  - presents

- Service
  - supports (how to access it)
  - described by (how it works)

- ServiceGrounding
  - (how to access it)
The Web Service Modeling Ontology (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

Connectors between components with mediation facilities for handling heterogeneities

Semantic description of Web Services:
- Capability (functional)
- Interfaces (usage)
WSMO – Web Service Modeling Ontology

- WSMO working group includes the WSML working group, which aims at developing a language called Web Service Modeling Language (WSML) that formalizes the Web Service Modeling Ontology (WSMO).

- WSMO: an ontology called Web Service Modeling Ontology (WSMO) for describing various aspects related to Semantic Web Services. Taking the Web Service Modeling Framework (WSMF) as a starting point, we refine and extend this framework, and develop an ontology and a description language.

- WSML: aims developing a language called Web Service Modeling Language (WSML) that formalizes the Web Service Modeling Ontology (WSMO). Hereby, we have a two fold mission: a) developing a proper formalization language for semantic web services and b) providing a rule-based language for the semantic web.
WSMF

WSMF [consists of four different main elements for describing semantic Web Services:

- (1) ontologies that provide the terminology used by other elements,
- (2) goals that define the problems that should be solved by Web Services,
- (3) Web Services descriptions that define various aspects of a Web Service, and
- (4) mediators which bypass interpretability problems.
WSMO Web Service Description Model

Non-functional Properties
- complete item description
- quality aspects
- economic aspects
- for item management

Capability
- Advertising of Web Service
- Support for WS Discovery

DC + QoS + Version + financial

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

realization of functionality by aggregating
other Web Services
- functional
decomposition
- WS composition

Choreography --- Service Interfaces --- Orchestration

client-service
interaction interface
for consuming WS
- External Visible
Behavior
- Communication
Structure
- 'Grounding'
WSMO Working Groups

www.wsmo.org

Conceptual Model & Axiomatization for SWS

European Semantic Systems Initiative

Ontology & Rule Language for the Semantic Web

Formal Language for WSMO

Execution Environment for WSMO

WSML

WSMX

STI2 CMS WG

SEE TC
Semantically-Enabled Service-oriented Architecture

Stakeholders Layer

Problem Formulation Layer

Service Requesters Layer

User 1

Back-end System X

Business Service S1

User 2

System Administrator

Domain Expert

Software Engineer

Applications
(user tools, access portals, …)

Domain Ontologies

Developer Tools
(ontology management, monitoring, …)

Network
(internet, intranet, extranet)

Semantic Execution Environment (Machine A)

Discovery

Adaptation

Fault Handling

Monitoring

Orchestration

Mediation

Composition

Grounding

Execution Management Security

base

Formal Languages

Reasoning

Storage

Communication

Shared Message Space

SEE (Machine B)

SEE (Machine C)

SEE (Machine D)

Middleware Layer

Service Providers Layer
This specification defines a set of extension attributes for the Web Services Description Language and XML Schema definition language that allows description of additional semantics of WSDL components. The specification defines how such semantic annotation is accomplished using references to semantic models, e.g. ontologies.

3 constructs: modelReference, liftingSchemaMapping, loweringSchemaMapping
A Web Service Composition Scenario with Ontology Reasoning

A Web service request

PartNumber  
DueDate  
Quantity

CheckAvailabilityRequestService  
AvailabilityConfirmation

A Possible Web service composition that fulfills the above request

PartNumber subClassOf SKU  
DueDate  
Quantity

SKU2UPCLookup  
UPC  
DeliveryDate  
NumBundles

equivalentClass  
equivalentClass

CheckInventoryService  
AvailabilityConfirmation  
NumBundles
Model Driven Interoperability - for services
Architecture for semantic annotation and reconciliation

Reference Ontology

Design-time

Run-time

Internet

Reconciliation
EMPOWER

- an innovative framework for interoperability between enterprise systems
- a flexible and extensible architecture
- a system environment
Model Transformation Services support the runtime lifting and lowering transformations among messages and ontologies based on the Model Map.

Ontology Definition Meta-model is a family of MOF meta-models, mappings between those meta-models, and a set of profiles that enable ontology modeling through the use of UML-based tools.

Semantic Annotation Model editor is used to relate different PIM models and ontology. It is used to annotate the SoaML model with Ontology.

SoaML describes the services models. The Model Mapping in the MEMPOWER includes transformations from models to ontology and ontology to models.

Model Map stores mapping rules.
Overall SAS architecture
Ontology example
Address in the Ontology
“Address” in the source schema and UML
“Address” in the source and target transformation rules
“Address” transformations from source.xml and target.xmi
Further details during Semantic Days

- Tue: 1100-1130 Semantic interoperability for public administrations in Europe – Challenges and solutions (EIF)
- Tue: 1530-1600 Methods and tools for semi-automatic ontology engineering
- Tue: 1600-1630 QuOnto: Ontology-based data access and integration using relational technology
- Tue: 1700-1730 Semantic annotation for web services and their relevance to environmental models (ENVISION, SWING)
- Tue 1730-1800 Systems interoperability through use of semantic technologies (COIN)

- Wed: 0830-0900 Information management for interoperability in European air traffic control (SESAR)
- Wed: 1030-1100 Supporting intelligent and automated integrated operations with agent technologies in a services architecture (SHAPE)
- Wed: 1130-1200 IT architecture for supporting semantic interoperability through use of semantic annotations (EMPOWER)
- Wed: 1330-1400 Best practices in collaborative ontology engineering
- Wed: 1430-1500 Model-driven integration architecture for IO G2
- Wed: 1500-1530 Using semantic technology to represent standards for operations & maintenance

18.05.2009

Arne Jørgen Berre, SINTEF, Norway
Ulf Larsson, LFV, Sweden
Dima Panfilenko, DFKI IWi, Germany