Rapid Prototyping for Service-Oriented Architectures

2nd Workshop on Web Services Interoperability (WSI 2006)
Bordeaux, France, March 21, 2006

Julien Vayssière¹, Gorka Benguria², Brian Elvesæter³, Klaus Fischer⁴, Ingo Zinnikus⁴

¹ SAP Research, Brisbane, Australia, julien.vayssiere@sap.com
² European Software Institute (ESI), Bizkaia, Spain, gorka.benguria@esi.es
³ SINTEF ICT, Oslo, Norway, brian.elvesater@sintef.no
⁴ DFKI GmbH, Saarbrücken, Germany, {klaus.fischer, ingo.zinnikus}@dfki.de
Outline

• Background and motivation
• Framework overview
  – Model-driven development framework
  – Web services enactment framework
  – Agent-based execution platform
• Prototyping case study
• Conclusion
Background

• Service-Oriented Architecture
  – architectural style
  – gaining momentum
  – mainstream in enterprise computing

• Four tenets of service-orientation (Box 2004)
  – explicit boundaries
  – autonomy of services
  – declarative interfaces and data formats
  – policy-based service description

• Web services architecture
  – technology most often used for implementing SOAs
  – standards-based stack of specifications
  – enable interoperable interactions between Web-based applications
Motivation

• Prototyping SOAs
  – working implementation of an SOA that can be used for validating the initial design choices

• Different compared to traditional application development
  – need to take into account existing services
  – developed by organisations over which we have no control
  – introduces constraints into the prototyping exercise

• Current state of the art tools
  – assumes that we are starting with a blank page
  – merely extends the approach of regular software prototyping to the scale of SOAs
  – they make the implicit assumption that services will behave as expected

• This is why we designed an approach that
  – from the start, takes into account the fact that parts of the SOA needs to be considered as a given; and
  – should be treated with a healthy dose of caution.
Framework overview

- The ATHENA baseline methodology for SOA provides guidelines for developing platform independent models for SOA (PIM4SOA).
- Provides a set of modelling tools and services for mapping between PIM4SOA and platform specific models (Web services and BDI agents).

- The Web service extensions to the JACK autonomous agents platform allow SOAs to use agents for brokering, mediation and negotiation between Web services.
- BDI teams provide a flexible and composable alternative to traditional approaches to Web service composition.

- Johnson and Lyndon provide enactment of all the roles found in an SOA (consumer, provider, intermediary) and flexible communication between Web services through an intuitive user interface.
- The WSDL Analyzer tool detected syntactical mismatches between service descriptions and provides a basis for runtime mediation of Web service messages.
Model-driven development framework

- **Follows the OMG Model Driven Architecture (MDA)**
  - Defines a Platform Independent Model (PIM) for SOA (PIM4SOA)
  - Platform Specific Models (PSMs) for describing Web services (XML Schemas and WSDL), Jack BDI agents and BPEL (Business Process Execution Language)

- **PIM4SOA is a visual PIM which specifies services in a technology independent manner**
  - Integrated view of the SOA in which different components can be deployed on different execution platforms.
  - The PIM4SOA model helps us to align relevant aspects of enterprise and technical IT models
  - Allows us to raise the abstraction level at which we can talk about and reason on the architecture we design.
PIM4SOA → platform specific models
### PIM4SOA addresses four system aspects

<table>
<thead>
<tr>
<th>Metamodel for (software) services</th>
<th>Metamodel for (automated software) processes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Services are an abstraction and an encapsulation of the functionality provided by an autonomous entity. Service architectures are composed of functions provided by a system or a set of systems to achieve a shared goal.</td>
<td>Processes describe sequencing of work in terms of actions, control flows, information flows, interactions, protocols, etc.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Metamodel for information</th>
<th>Metamodel for quality of service (QoS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information is related to the messages or structures exchanged, processed and stored by software systems or software components.</td>
<td>Extra-functional qualities that can be applied to services, information and processes.</td>
</tr>
<tr>
<td>Structural constructs for class modelling in UML 2.0 (OMG 2003)</td>
<td>UML Profile for Modeling Quality of Service and Fault Tolerance Characteristics and Mechanisms (OMG 2004)</td>
</tr>
<tr>
<td>UML Profile for Enterprise Distributed Object Computing (OMG 2002)</td>
<td></td>
</tr>
</tbody>
</table>
Web services architecture metamodel

- **Registry**
  - + UDDI

- **Service Interface Description**
  - + WSDL 1.1
  - + WSDL 2.0

- **Endpoint Description**
  - + WS-MetadataExchange
  - + WS-Policy
  - + WS-PolicyAttachment

- **Reliability**
  - + WS-ReliableMessaging

- **Coordination**
  - + WS-Coordination

- **Messaging**
  - + SOAP
  - + WS-Addressing

- **Transport**
  - + HTTP

- **Resource Access and Management**
  - + WS-Enumeration
  - + WS-Resource
  - + WS-ResourceLifetime
  - + WS-ResourceProperty
  - + WS-Transfer

- **Eventing**
  - + WS-BaseNotification
  - + WS-BrokeredNotification
  - + WS-Eventing
  - + WS-Topics

- **Coordination**
  - + WS-Coordination

- **Eventing**
  - + WS-BaseNotification
  - + WS-BrokeredNotification
  - + WS-Eventing
  - + WS-Topics

- **Resource Access and Management**
  - + WS-Enumeration
  - + WS-Resource
  - + WS-ResourceLifetime
  - + WS-ResourceProperty
  - + WS-Transfer

- **Security**
  - + WS-Security

- **Composition**
  - + ACE-GIS Composition Extensions
  - + WS-BPEL
  - + WS-CDL

- **XML**
  - + XML Core / XSD
  - + XML Encryption
  - + XML Signature
  - + XPATH

- **Mes saging**
  - + SOAP
  - + WS-Addressing

- **XML**
  - + XML Core / XSD
  - + XML Encryption
  - + XML Signature
  - + XPATH

- **Security**
  - + WS-Security

- **Composition**
  - + ACE-GIS Composition Extensions
  - + WS-BPEL
  - + WS-CDL

- **eContract**
  - + ATHENA eContract Extensions
PIM4SOA → platform specific models

PIM4SOA source metamodel

Mapping model

Web services target metamodel

Bordeaux, France, March 21, 2006
Model transformations

**EMF**: Eclipse Modeling Framework  
**RSM**: Rational Software Modeler UML modelling tool  
**PIM4SOA Plugin**: RSM plugin – UML Profile for PIM4SOA  
**MTF**: Model Transformation Framework  
**Eclipse WST**: WSDL graphical editor  
**WID**: WebSphere Integration Developer

- **Bordeaux, France, March 21, 2006**
- **EMF**
- **RSM**
- **PIM4SOA Plugin**
- **EMF**: BPEL Metamodel (.ecore)  
- **RSM**: eProc Model (.emx)  
- **PIM4SOA Plugin**: UML Profile for PIM4SOA (.epx)  
- **BPEL Metamodel**
- **UML 2.0 Metamodel**
- **eProc Metamodel**
- **Eclipse WST**: WSDL Metamodel (.ecore)  
- **WID**: BPEL Document (.bpel)  
- **EMF**: eProc Document (.ecore)  
- **MTF**: WSDL Document (.wsdl)
Rapid prototyping framework for SOA

- The ATHENA baseline methodology for SOA provides guidelines for developing platform independent models for SOA (PIM4SOA).
- Provides a set of modelling tools and services for mapping between PIM4SOA and platform specific models (Web services and BDI agents).

Benefits:
- Johnson and Lyndon provide enactment of all the roles found in an SOA (consumer, provider, intermediary) and flexible communication between Web services through an intuitive user interface.
- The WSDL Analyzer tool detected syntactical mismatches between service descriptions and provides a basis for runtime mediation of Web service messages.
- The Web service extensions to the JACK autonomous agents platform allow SOAs to use agents for brokering, mediation and negotiation between Web services.
- BDI teams provide a flexible and composable alternative to traditional approaches to Web service composition.
The WSDL Analyzer is a tool for detecting similarities between Web service descriptions.

- The tool can be used to find a list of similar services and produces a mapping between messages, thereby enabling brokering and mediation of services.
- The idea of the tree-edit distance is that a similarity between two XML structures can be measured by stepwise transforming a tree representation of the first structure into the other.

A possible scenario for using the WSDL Analyzer is that the user already knows a service which provides the correct format.

- The WSDL of this service can be used as requirement for a similarity search.
- The WSDL Analyzer allows browsing the original WSDL and the candidate files.
Web services enactment framework (1)

- Johnson is a runtime enactment tool
  - It enables users to enact most of the roles typically found in an SOA.
  - It allows sending real SOAP messages between Web services without having to write a single line of code.
  - It features a Web-based user interface designed to closely resemble Web-based email applications.
  - SOAP messages and Web Services endpoints are used in place of email messages and email addresses.
  - The user can see incoming SOAP messages in the Inbox and create outgoing SOAP messages in the Outbox that will be sent to external Web services.
  - A powerful user-interface generator relieves the user from having to deal with XML documents by generating forms for displaying and editing any XML-based data type.
• Lyndon is the design-time counterpart of the Johnson tool
  – It analyses WSDL files and automatically configures Johnson for playing either the role of consumer or provider of the service described.
  – Lyndon parses a WSDL file and determines which endpoints need to be created, and which processing chains need to be assigned to them.
  – Determining which processing modules to include in the processing chain takes into account information extracted from the WSDL file as well as options set by the user.
  – The user may, for example, specify whether Johnson should be configured as a service consumer or a service provider, or whether messages sent to or from the service should be logged
  – Some configuration information can be extracted from the WSDL file, such as the need for implementing the WS-Addressing specification, which is specified as part of the description of the bindings of a Web service.
Rapid prototyping framework for SOA

• The ATHENA baseline methodology for SOA provides guidelines for developing platform independent models for SOA (PIM4SOA).
• Provides a set of modelling tools and services for mapping between PIM4SOA and platform specific models (Web services and BDI agents).

Agents
• The Web service extensions to the JACK autonomous agents platform allow SOAs to use agents for brokering, mediation and negotiation between Web services
• BDI teams provide a flexible and composable alternative to traditional approaches to Web service composition

Services
• Johnson and Lyndon provide enactment of all the roles found in an SOA (consumer, provider, intermediary) and flexible communication between Web services through an intuitive user interface
• The WSDL Analyzer tool detected syntactical mismatches between service descriptions and provides a basis for runtime mediation of Web service messages

Modelling
• The WSDL Analyzer tool detected syntactical mismatches between service descriptions and provides a basis for runtime mediation of Web service messages

Bordeaux, France, March 21, 2006
At its core the JACK agent framework with plan library and knowledge base.

Following the MDA approach, a modeller specifies at design time a set of plans (PSM level) that constitute the workflow library of the agents.

Web service calls are integrated as steps into plans.

Workflows are modelled graphically and most of the common workflow patterns are supported.
The following approach was followed for the validation of the rapid prototyping framework:

- Used the MDD framework (1) to derive the WSDL files and BDI models from the e-procurement PIM4SOA model.
- Enact the services identified for the e-procurement scenario using the WSDL Analyser (2) and the Lyndon (3) tools.
  - Used the WSDL Analyser to locate existing services similar to those required in the e-procurement scenario.
  - Used the Lyndon tool to configure the Johnson platform to simulate required, new services.
- Configure Johnson (4) to act as a service proxy.
  - This allowed us to change the final service endpoints without affecting the process execution.
- Finally the PSM model for Jack (5) was implemented and tested with the enacted services.
R1. Request for Quotation
R2. Quotation
R3. Order
R4. Order Confirmation

R1. Request for Quotation
R2. Quotation
R3. Order
R4. Order Confirmation

- Retailer-Manufacturer
  - 1. RFQ
  - 2. Quote
  - 3. Order

- Manufacturer-Supplier
  - 1. RFQ
  - 2. Quote
  - 3. Order
  - 4. Order Confirmation

- Retailer-Manufacturer
  - 4. Order Confirmation

Interior Decoration Project

MANUFACTURER

RETAILER

PROVIDER
PIM4SOA: Order process
PIM4SOA: Furniture procurement collaboration

- Three roles
  - “Retailer”
  - “Manufacturer”
  - “Supplier”
- Two usage of collaboration
  - “Goods Supply”
  - “Materials Supply”
- Relationships between role and collaboration use
  - “RoleBinding”
PIM4SOA: Goods supply collaboration

```
get quotation : Obtain Quotation

Place Order : Manage Order

shipping : Delivery

arrange payment : Invoicing
```

Bordeaux, France, March 21, 2006
PIM4SOA: Services interfaces

IManufacturerService

- order (input: order)
- orderResponse (input: orderResponse)
- quotation (input: quotation)
- requestForQuotation (input: requestForQuotation)
PIM4SOA: Order document
Conclusion

• This paper presented a rapid prototyping framework for SOAs built around a model-driven development (MDD) methodology which is used for
  – transforming high-level specifications of an SOA into executable artefacts
  – Web Services
  – autonomous agents
• The framework can handle
  – a mix of new and existing services; and
  – provides facilities for simulating, logging, analysing and debugging
• The framework was validated in an industrial electronic procurement scenario from the furniture manufacturing industry
  – input from business expert had been collected
  – creating the high-level PIM4SOA model
  – deriving the Web service description and incorporating existing Web services took less than a day for a person already familiar with all the tools involved
• The model-based approach helps us in
  – keeping all the pieces of the SOA aligned with high-level business objectives throughout rounds of prototyping