Web Services Composition: Mashups Driven Orchestration Definition

Sébastien Mosser
Franck Chauvel  Mireille Blay–Fornarino  Michel Riveill
mosser@polytech.unice.fr

CNRS, I3S Laboratory, Modalis Team
University of Nice – Sophia Antipolis
http://rainbow.i3s.unice.fr

IAWTIC’08
December, 10th 2008
Wien, Austria
1. Introduction: Orchestration vs Mashups

2. Seduite: Defined as Mashups, Executed as Orchestrations

3. A Model Driven Engineering Approach

4. Model Transformation: From Mashup to Orchestration

5. Implementation & Validation

6. Conclusions
1. Introduction: Orchestration vs Mashups
2. Seduite: Defined as Mashups, Executed as Orchestrations
3. A Model Driven Engineering Approach
4. Model Transformation: From Mashup to Orchestration
5. Implementation & Validation
6. Conclusions
Initial Context: Web Services Oriented Architecture

- Web Services:
  - Elementary brick
  - Syntactical contract
  - Platform Independent

- Orchestrationss:
  - Composite application
  - Exposed as a Service
  - Industrial usage (BPEL)

- WSOA Objectives:
  - Developers build services
  - Business analysts define Orchestrations
Using BPEL to implement Orchestration
Using a designer to draw Orchestration...
Data composition: the mashup approach

- A Web 2.0 principle:
  - The Web provides informations (RSS, Flickr, Picasa, ...)
  - Let's combine those informations
  - Value added information production!

- Multiples usage:
  - User friendly
  - “End user programming for the Web”
  - e–Sciences users like biology, ...

⇒ Let’s fill the gap between WSOA & Mashups!
Example: Combine RSS news with Flickr Pictures
Agenda

1. Introduction: Orchestration vs Mashups

2. Seduite: Defined as Mashups, Executed as Orchestrations

3. A Model Driven Engineering Approach

4. Model Transformation: From Mashup to Orchestration

5. Implementation & Validation

6. Conclusions
Global overview:

- An information broadcasting system build as a WSOA
- Domain specific: Academic institutions
- Used as a validation platform by FAROS

Founding principles:

- Sources of information:
  - Timetable, Bus schedule, Restaurant menu, ...
  - Implemented as Web Services
- Business Processes:
  - Information retrieval process, public broadcast, ...
  - Implemented as BPEL Orchestrations
Retrieving School information through Seduite
Profiling information retrieval process
Natural design methodology

- Previous figures define data–flow
- Information Data Flow $\equiv$ Mashup!

But, in “real–life”

- Business process must be deployed inside the BPEL engine
- The designer must develop the process by hand
- And express a control–flow instead of a data–flow

⇒ Is there a way to express a mashup at the design level and obtain an orchestration at the platform one?
Agenda

1. Introduction: Orchestration vs Mashups
2. Seduite: Defined as Mashups, Executed as Orchestrations
3. A Model Driven Engineering Approach
4. Model Transformation: From Mashup to Orchestration
5. Implementation & Validation
6. Conclusions
Capturing business domain inside meta–models

- A meta–model defines a set of domain–dedicated concepts
- And a set of relationship between these entities
- We define two meta–models:
  - Mashups data–flow & WSOA

Model Transformation: from Mashup to WSOA

- Express the correspondence between two meta–models
- Implemented using existing and efficient tools (Kermeta, …)
- cf next part of the presentation
Main concept: a Mashup is a Flow

- A Flow is a set of Nodes
- ⇒ Connected through a DataPath.
- A Node is a Source of informations
- ⇒ or a Process applied to informations

Data-driven definition

- A Node exposes output data through Slots
- A Process defines input Slots
- A Slot is connected to a Parameter
  ⇒ Constant, FlowParameter
Target meta–model: Simplified WSOA

A simplified WSOA meta–model

- A System is defined as a set of Services
  - Do not forget: an Orchestration is a Service
- A Service exposes several Operations
- An Operation takes a set of input Parameters
- And can return a single one.

- We do not address the behavioral content of Operation
- The goal of this work is to reach a legacy architecture
  - Not to automagically generate an application from scratch
Target meta-model: Simplified WSOA
1. Introduction: Orchestration vs Mashups
2. Seduite: Defined as Mashups, Executed as Orchestrations
3. A Model Driven Engineering Approach
4. Model Transformation: From Mashup to Orchestration
5. Implementation & Validation
6. Conclusions
Global Overview of the transformation process

User point of view: Design a mashup

- System user graphically design a mashup (ie a data–flow)
  - Connecting boxes using pipes
  - And then *publishes* the mashup

System point of view: Generate an orchestration

- Optimize the source data–flow to follow WSOA guidelines:
  - $T_0$: Unfold mashups invocations
  - $T_1$: Group invocations
  - $T_2$: Identify business operations

- Generate a control–flow from a data–flow
$T_0$: Unfolding mashups invocation
$T_1$: Grouping redundant invocation
Optimizing redundant invocations

- Get Timetable and then Filter data:
  - Timetable does not take any parameter
    ⇒ Use a single invocation
  - Filter process uses an array of input parameters
    ⇒ Compute the union of expressed parameters

- Retrieve Weather information
  - Existing legacy & black box service
    ⇒ Do not change anything

- Concatenation Sequence:
  - $\oplus$ process is associative
    ⇒ Use a single node with good parameters
 UserInfo' \equiv T_1(T_0(UserInfo))
\( T_2 \): Identifying business sub–processes

- Identifying recurrent paths at a global point of view
- Filtering timetable appears inside School and User mashups
- It makes sense to expose this process as a service operation!
  \[ \Rightarrow \text{Let’s generate the expected architecture!} \]
Result: Transformation output

- Legacy services are identified and addressed:
  - Weather, TimeTable, News & UserProfile
- Business processes are defined by inverting the data flow:
  - The sub-process is exposed as an orchestration FilteredTimeTable
  - Information providing mashups define an InformationProvider orchestration.

⇒ The architectural part of the WSOA is automatically generated from users-driven mashups.
Agenda

1. Introduction: Orchestration vs Mashups
2. Seduite: Defined as Mashups, Executed as Orchestrations
3. A Model Driven Engineering Approach
4. Model Transformation: From Mashup to Orchestration
5. Implementation & Validation
6. Conclusions
Implementation: State of the art MDE technologies

**Meta–models definition**

- Eclipse Modeling Framework
- Bundled with Eclipse IDE
- More restrictive than UML (but sufficient !!)

**Transformation Implementation**

- Kermeta “*breathe life into metamodels*”
- An aspect weaver & toolbox for meta–models
- The transformation is woven as a Visitor design pattern
Faros project: “Reliable Services Oriented Architecture”

Goals of the FAROS National research project:
- Expressing contracts at a business domain level
- Reach execution platform as target

Seduite as a validation application
- Information management is a business domain
  - Faros meta–model is more complicated than the mahsups one
- The legacy infrastructure is a WSOA
  - Deployed inside two academic institutions

⇒ The work presented here is a subpart of the transformation set defined inside the FAROS project.
<table>
<thead>
<tr>
<th>Agenda</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1</strong> Introduction: Orchestration vs Mashups</td>
</tr>
<tr>
<td><strong>2</strong> Seduite: Defined as Mashups, Executed as Orchestrations</td>
</tr>
<tr>
<td><strong>3</strong> A Model Driven Engineering Approach</td>
</tr>
<tr>
<td><strong>4</strong> Model Transformation: From Mashup to Orchestration</td>
</tr>
<tr>
<td><strong>5</strong> Implementation &amp; Validation</td>
</tr>
<tr>
<td><strong>6</strong> Conclusions</td>
</tr>
</tbody>
</table>
Related Work & Perspectives

- Data–flow expressiveness:
  - Real data–flows languages handle loops, XOR, ...
  ⇒ Enrich the mashup meta–model semantic

- Optimization inference techniques:
  - Reify *good properties* of known nodes inside mashups
  ⇒ Use a real inference engine (Prolog) to perform the optimization

- Grid Computing Workflow:
  - Large–scale and complicated data–flows
  ⇒ Graph optimization techniques can be used
Contribution summary

- The SOA paradigm does not provide the good abstraction level
- Mashups can be used to define user-friendly data-flows
- Following a model driven engineering methodology:
  - We define two meta-models (Mashups & WSOA)
  - We implement optimization techniques
  - We generate a reusable WSOA from a set of mashups
- A more global transformation is defined inside FAROS:
  - Orange Labs, EDF, IRISA, LIFL, I3S, Alicante

http://www.lifl.fr/faros
Any questions?

http://anubis.polytech.unice.fr/jSeduite