



Data, Information and Process Integration
with Semantic Web Services

DIP

Data, Information and Process Integration with Semantic Web Services

FP6 – 507483

Deliverable

WP4: Service Usage

D4.20

DIP Orchestration prototype

Thomas Haselwanter
Sami Bhiri
Amit Goyal

September 3, 2006



EXECUTIVE SUMMARY

This deliverable documents the release of the WSMX orchestration component. The document displays the scope of this component and describes roughly its functionality. It explains how to run the provided demonstration illustrating the functionality of the orchestration component prototype. Finally, it details how to install and use it with respect to the DIP/WSMX architecture.

Disclaimer: The DIP Consortium is proprietary. There is no warranty for the accuracy or completeness of the information, text, graphics, links or other items contained within this material. This document represents the common view of the consortium and does not necessarily reflect the view of the individual partners.

DOCUMENT INFORMATION

IST Project Number	FP6 – 507483	Acronym	DIP
Full Title	Data, Information, and Process Integration with Semantic Web Services		
Project URL	http://dip.semanticweb.org/		
Document URL			
EU Project Officer	Brian Macklin		

Deliverable	Number	4.20	Title	DIP Orchestration prototype
Work Package	Number	4	Title	Service Usage




Date of Delivery	Contractual	M30	Actual	05-Jun-06
Status	version 001		final	<input type="checkbox"/>
Nature	prototype <input checked="" type="checkbox"/> report <input type="checkbox"/> dissemination <input type="checkbox"/>			
Dissemination Level	public <input checked="" type="checkbox"/> consortium <input type="checkbox"/>			









Authors (Partner)	Thomas Haselwanter (UIBK), Sami Bhiri (NUIG), Amit Goyal (NUIG)			
Resp. Author	Thomas Haselwanter		E-mail	thomas.haselwanter@deri.org
	Partner	UIBK	Phone	+43 512 507 6423

Abstract (for dissemination)	This deliverable documents the orchestration component release. It displays the component scope and roughly describes its functionality. The document explains how to run the provided demonstration of the prototype. It details how to install and use this componnet in the scope of the DIP/WSMX architecture.
Keywords	WSMO, WSMO service orchestration, ontologized abstract state machine

Version Log			
Issue Date	Rev No.	Author	Change
22-12-05	1	Thomas Haselwanter	First release of the orchestration component prototype: Abstract State Machine infrastructure
05-01-06	2	Sami Bhiri	Fact sheet and documentation of the orchestration component first release
30-05-06	3	Thomas Haselwanter	Orchestration Component prototype
15-06-06	4	Amit Goyal	Modification of the orchestration component interface
30-06-06	5	Sami Bhiri	Fact sheet, documentation and demonstration of the orchestration component
22-08-06	6	Sami Bhiri	Transcription of the fact sheet, documentation and demonstration guidelines into the DIP deliverable template

PROJECT CONSORTIUM INFORMATION

Partner	Acronym	Contact
National University of Galway	NUIG 	Prof. Dr. Christoph Bussler Digital Enterprise Research Institute (DERI) National University of Ireland, Galway Galway Ireland E-mail: chris.bussler@deri.ie Tel: +353 91 512460
Fundacion De La Innovacion.Bankinter	Bankinter 	Monica Martinez Montes Fundacion de la Innovation. BankInter, Paseo Castellana, 29 28046 Madrid, Spain Email: mmtnez@bankinter.es Tel: 916234238
Berlecon Research GmbH	Berelcon 	Dr. Thorsten Wichmann Berlecon Research GmbH, Oranienburger Str. 32, 10117 Berlin, Germany E-mail: tw@berlecon.de Tel: +49 30 2852960
British Telecommunications Plc.	BT 	Dr. John Davies BT Exact (Orion Floor 5 pp12) Adastral Park Martlesham Ipswich IP5 3RE, United Kingdom Email: john.nj.davies@bt.com Tel: +44 1473 609583
Swiss Federal Institute of Technology, Lausanne	EPFL 	Prof. Karl Aberer Distributed Information Systems Laboratory École Polytechnique Fédérale de Lausanne Bât. PSE-A 1015 Lausanne, Switzerland E-mail : Karl.Aberer@epfl.ch Tel: +41 21 693 4679
Essex County Council	Essex 	Mary Rowllatt, Essex County Council, PO Box 11, County Hall, Duke Street, Chelmsford, Essex, CM1 1LX, United Kingdom. E-mail: maryr@essexcc.gov.uk Tel: +44 (0)1245 436524
Forschungszentrum Informatik	FZI 	Andreas Abecker Forschungszentrum Informatik Haid-und-Neu Strasse 10-14 76131 Karlsruhe Germany E-mail: abecker@fzi.de Tel: +49 721 96540

Institut für Informatik, Leopold-Franzens Universität Innsbruck	UIBK 	Prof. Dieter Fensel Institute of computer science University of Innsbruck Technikerstr. 25 A-6020 Innsbruck, Austria Email: dieter.fensel@deri.org Tel: +43 512 5076485
ILOG SA	ILOG 	Christian de Sainte Marie 9 Rue de Verdun, 94253, Gentilly, France E-mail: csma@ilog.fr Tel: +33 1 49082981
inubit AG	inubit 	Torsten Schmale, inubit AG, Lützowstraße 105-106 D-10785 Berlin, Germany E-mail: ts@inubit.com Tel: +49 30726112 0
Intelligent Software Components, S.A.	iSOCO 	Dr. V. Richard Benjamins, Director R&D Intelligent Software Components, S.A. Pedro de Valdivia 10 28006 Madrid, Spain E-mail: rbenjamins@isoco.com Tel. +34 913 349 797
The Open University	OU 	Dr. John Domingue Knowledge Media Institute, The Open University, Walton Hall, Milton Keynes, MK7 6AA, UK E-mail: j.b.domingue@open.ac.uk Tel.: +44 1908 655014
SAP AG	SAP 	Dr. Elmar Dorner SAP Research, CEC Karlsruhe SAP AG Vincenz-Priessnitz-Str. 1 76131 Karlsruhe, Germany E-mail: elmar.dorner@sap.com Tel: +49 721 6902 31
Sirma AI Ltd.	Sirma 	Atanas Kiryakov, Ontotext Lab, - Sirma AI EAD, Office Express IT Centre, 3rd Floor 135 Tzarigradsko Chausse, Sofia 1784, Bulgaria E-mail: atanas.kiryakov@sirma.bg Tel.: +359 2 9768 303
Tiscali Österreich GmbH	Tiscali 	Dr Dieter Haacker Tiscali Österreich GmbH. Diefenbachgasse 35, A-1150 Vienna, Austria E-mail: Dieter.Haacker@at.tiscali.com Tel: +43 1 899 33 160



Unicorn Solution Ltd.	<p>Unicorn</p> 	<p>Jeff Eisenberg Unicorn Solutions Ltd, Malcha Technology Park 1 Jerusalem 96951, Israel E-mail: Jeff.Eisenberg@unicorn.com Tel.: +972 2 6491111</p>
Vrije Universiteit Brussel	<p>VUB</p> 	<p>Carlo Wouters, Starlab- VUB Vrije Universiteit Brussel Pleinlaan 2, G-10 1050 Brussel, Belgium E-mail: carlo.wouters@vub.ac.be Tel.: +32 (0) 2 629 3719</p>

TABLE OF CONTENTS

1	DOCUMENTATION OF THE ORCHESTRATION COMPONENT RELEASE	1
1.1	Purpose and functionality	1
1.2	Functionality demonstration	1
1.3	Installation and usage	2

1 DOCUMENTATION OF THE ORCHESTRATION COMPONENT RELEASE

In the following we describe the scope of the orchestration component and depict roughly its functionality. Then we detail how to run the demonstration that illustrates its practicality. Finally, we explain how to install and use it in the scope of the DIP/WSMX architecture.

1.1 Purpose and functionality

The scope of the WSMX orchestration component is to execute orchestration interfaces of composite services. From a conceptual point of view, a service orchestration describes how its functionality is implemented by orchestrating other services functionalities. In WSMO/WSMX context, a service orchestration interface is implemented as an Ontologized Abstract State Machine (OASM).

This component registers first a service orchestration interface after parsing its wsml file description. Then, given an input message, it navigates through the OASM invoking when indicated/necessary the component services functionalities.

1.2 Functionality demonstration

This component is intended to be run on the WSMX platform (see section 1.3 for installation). However, for illustration purpose all necessary libraries are included in the binary distribution.

We demonstrate the functionality of the prototype through an e-banking use case scenario (see the compressed document "Demonstration of the component functionalities"). The demonstration shows how the orchestration component registers the orchestration interface from the service description after parsing its wsml file description. It illustrates also the states change of the OASM (after eventually invoking other services functionalities). An illustrative movie demonstrates the main functionalities of this component. To play the movie; start the html file "demo".

To run manually the demo, you should: 1. launch the orchestration component, 2. register the Web service orchestration interface, 3. invoke the service orchestration and 4. eventually review the OASM states change. In the following we detail how to realize each of these steps.

1. To launch the orchestration component; run the batch file `wsmx_ebanking\start`, then point your browser to `http://localhost:8080`, choose the Server view tab from where choose the RadexOrchestration component. The component view tab displays 3 operations:
 - *registerOrchestration*: this operation allows to register the service orchestration given the wsml service description file (to paste it in the corresponding text box)
 - *getMachineHistory*: this operation allows to review the OASM states change.
 - *updateState*: this operation allows to invoke the registered orchestration interface. It requires, as a parameter, the sent message (to paste it in the corresponding text boxes)

2. To register a web service orchestration interface, invoke the registerOrchestration operation by providing the wsml description file. The wsml file describing the web service WSExecuteIfValueRises is provided under `wsmx_ebanking\resources\resourcemanager\webservices`.
3. To execute the orchestration interface, invoke the updatestate operation by providing the message to be sent. A message to send to the WSExecuteIfValueRises service is provided under `wsmx_ebanking\resources\resourcemanager\webservices`.
4. To review the states change of the orchestration OASM, invoke the gethistory operation.

1.3 Installation and usage

The licensing model of this component is LGPL. This component conforms with the DIP/WSMX Architecture and is intended to run on the WSMX platform. In order to execute it the following steps are necessary:

- Download a WSMX binary distribution from sourceforge or get the latest nightly build from <http://www.wsmx.org/downloads.html>
- Optional step: Start up a JavaSpaces compliant space implementation such as Outrigger or Blitz. If WSMX does not find a space during boot time it will substitute a virtual space that works as long as components only have local communication requirements.
- The microkernel within the executable `wsmx.core` can be run from the command line given that sufficient privileges are granted:

```
java -Djava.security.policy=/path/to/policy -jar wsmx.core
```

A sample policy file (`policy.all`) which grants unrestricted access is supplied with the release.
- To deploy the orchestration component, copy the packaged component archive to the systemcodebase. WSMX will discover it automatically and inject it into the running instance.

You may monitor and administer the orchestration component through either the GUI-based web console, the TUI-based SSH console, or the eclipse-base WSMT. Point your browser to `http://localhost:port`, where `port` is the port number which has been defined in the kernel configuration and is 8080 by default, if undefined and fallback for invalid ports. Point your SSH client to `localhost` and login with user `root` at the port defined in the configuration.

Even while it may be driven manually in this way, it is best utilized when driven by an execution semantic that executes the component in concert with other components. Execution semantics may be executed through the consoles as well.