



Data, Information and Process Integration
with Semantic Web Services

DIP

Data, Information and Process Integration with Semantic Web Services

FP6 - 507483

Deliverable

WP 6: Architecture and Interoperability

D6.9

Revision of API

Matthew Moran, Michal Zaremba, Mick Kerrigan
Emilia Cimpian, Adrian Mocan, Brahmananda Sapkota
James Scicluna, Thomas Haselwanter, Maciej Zaremba
John Domingue, Liliana Cabral, Joachim Quantz

December 20, 2004



SUMMARY

This document provides details of the revised API for components in the DIP architecture. The API is specified in Java with the source code available at the open-source software project, WSMX¹, hosted at SourceForge. The documentation tool, JavaDoc, was used to generate the content of this deliverable as it provides the most common way of presenting Java API for dissemination.

The HTML pages, generated using the JavaDoc tool, are available in the zip-file, *d6.9_HTML_API.zip*, at the URL - <https://bscw.dip.deri.ie/bscw/bscw.cgi/0/54228>. This is the most appropriate means of disseminating the API as it provides the facility to hyperlink between Java classes and interfaces, which is the most intuitive means for reading or using a Java API.

The contribution of this deliverable to the four golden bullets of the DIP project is described in the following table:

Golden Bullet	Contribution of this Deliverable
Open-source architecture for Semantic Web services	The API is an integral part of the WSMX open-source architecture
Real Use Case Implementations	The API includes the description of the interfaces that each of the DIP use-cases will use to achieve their aims
Practical, exploitable tools and methods	The API is coded using industry standard Java v.5
Impact on International Standards	The API of the Semantic Execution environment described in this deliverable is a key contribution to the standardization work started in the OASIS Semantic Execution Environment Technical Committee (OASIS SEE TC).

Additionally, the deliverable is relevant to all technical work packages of DIP as it defines the API that each functional component should implement so that it can operate in the DIP reference implementation provided by WSMX. This document is of principle interest to developers working on functional components for any of the DIP technical work packages.

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.

¹ <http://www.sourceforge.net/projects/wsmx>

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	Kai Tullius		

Deliverable	Number	6.9	Title	Revision of API
Work package	Number	6	Title	Architecture and Interoperability

Date of delivery	Contractual	M 24	Actual	31 December 2005
Status	Version. 0.01		final	
Nature	Prototype <input type="checkbox"/> Report <input checked="" type="checkbox"/> Dissemination <input type="checkbox"/>			
Dissemination Level	Public <input checked="" type="checkbox"/> Consortium <input type="checkbox"/>			

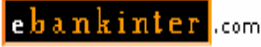




Authors (Partner)	Matthew Moran (NUIG), Michal Zaremba (NUIG), Mick Kerrigan (NUIG), Emilia Cimpian (NUIG), Adrian Mocan (NUIG), Brahmananda Sapkota (NUIG), Maciej Zaremba (NUIG), Thomas Haselwanter (UIBK), James Scicluna (UIBK), John Domingue (OU), Liliana Cabral (OU), Joachim Quantz (inubit)			
Responsible Author	Author 1	Matthew Moran	Email	Matthew.moran@deri.org
	Partner	NUIG	Phone	+353 91 495017

Abstract (for dissemination)	<p>This document provides details of the revised API for components in the DIP architecture. The API is specified in Java with the source code available at the open-source software project, WSMX, hosted at SourceForge.</p> <p>The deliverable consists of this covering document and a zip file containing HTML pages generated from the comments embedded in the Java coding of the API. The zip file is available at: https://bscw.dip.deri.ie/bscw/bscw.cgi/0/54228</p>	
Keywords	API, Architecture, Interface	

Version Log			
Issue Date	Rev No.	Author	Change
30-Nov-05	001	Matthew Moran	First version of the deliverable.
20-Dec-05	002	Matthew Moran	Response to review comments. Removed PDF form of API from this document to separate set of HTML pages. The tool that was used to create the PDF form of the API had several bugs which resulted in an incomplete API description.

Reviewer			
Marc Richardson	Reviewer 1		Email marc.richardson@bt.com
	Partner	BT	Phone
Vassil Momtchev	Reviewer 2		Email vassil.momtchev@ontotext.com
	Partner	Sirma	Phone

Project Consortium Information

Partner	Acronym	Contact
National University of Ireland Galway	NUIG  <i>National University of Ireland, Galway</i> <i>Ollscoil na hÉireann, Gaillimh</i>	Dr. Sigurd Harand Digital Enterprise Research Institute (DERI) National University of Ireland, Galway Galway Ireland Email: sigurd.haramd@deri.org Tel: +353 91 495112
Fundacion De La Innovacion.Bankinter	Bankinter 	Monica Martinez Montes Fundacion de la Innovacion. BankInter Paseo Castellana, 29 28046 Madrid, Spain Email: mmtnez@bankinter.es Tel: 916234238
Berlecon Research GmbH	Berlecon 	Dr. Thorsten Wichmann Berlecon Research GmbH Oranienburger Str. 32 10117 Berlin, Germany Email: 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 Email : Karl.Aberer@epfl.ch Tel: +41 21 693 4679
Essex County Council	Essex 	Mary Rowlett, Essex County Council PO Box 11, County Hall, Duke Street Chelmsford, Essex, CM1 1LX United Kingdom. Email: 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 Email: abecker@fzi.de Tel: +49 721 9654 0

Partner	Acronym	Contact
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  Changing the rules of business	Christian de Sainte Marie 9 Rue de Verdun, 94253 Gentilly, France Email: csma@ilog.fr Tel: +33 1 49082981
inubit AG	Inubit  the integration experts	Torsten Schmale inubit AG Lützowstraße 105-106 D-10785 Berlin Germany Email: 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 Email: rbenjamins@isoco.com Tel. +34 913 349 797
NIWA WEB Solutions	NIWA 	Alexander Wahler NIWA WEB Solutions Niederacher & Wahler OEG Kirchengasse 13/1a A-1070 Wien Email: wahler@niwa.at Tel:+43(0)1 3195843-11
The Open University	OU  The Open University	Dr. John Domingue Knowledge Media Institute The Open University, Walton Hall Milton Keynes, MK7 6AA United Kingdom Email: 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 Email: elmar.dorner@sap.com Tel: +49 721 6902 31




<p>Sirma AI Ltd.</p>	<p>Sirma</p>  <p>Ontotext Knowledge and Language Engineering Lab of Sirma</p>	<p>Atanas Kiryakov, Ontotext Lab, - Sirma AI EAD Office Express IT Centre, 3rd Floor 135 Tzarigradsko Chausse Sofia 1784, Bulgaria Email: atanas.kiryakov@sirma.bg Tel.: +359 2 9768 303</p>
<p>Unicorn Solution Ltd.</p>	<p>Unicorn</p> 	<p>Jeff Eisenberg Unicorn Solutions Ltd, Malcha Technology Park 1 Jerusalem 96951 Israel Email: Jeff.Eisenberg@unicorn.com Tel.: +972 2 6491111</p>
<p>Vrije Universiteit Brussel</p>	<p>VUB</p>  <p>Vrije Universiteit Brussel</p>	<p>Carlo Wouters Starlab- VUB Vrije Universiteit Brussel Pleinlaan 2, G-10 1050 Brussel ,Belgium Email: carlo.wouters@vub.ac.be Tel.: +32 (0) 2 629 3719</p>

TABLE OF CONTENTS

SUMMARY	I
TABLE OF CONTENTS	VII
1 INTRODUCTION	VIII
2 CONTENT	IX
REFERENCES	IX

LIST OF TABLES

Table 1: Java Package Description	viii
---	------

1 INTRODUCTION

This revision of the DIP API clarifies the definition of the interfaces of the various components that, together, form the DIP architecture. There are two broad viewpoints that are covered:

1. API to agents outside the DIP architecture wishing to interact with it.
2. APIs for each internal component of the DIP architecture defining how other elements of the architecture can interact with them.

The viewpoint of an agent outside the DIP architecture that wishes to use the DIP architecture to achieve a particular task is defined by the EntryPoint interface. This interface provides a number of entry points to the DIP architecture. In the WSMX reference implementation for DIP, each entry point is mapped to a description of an *execution semantics*. Each *execution semantics* is used to define how some or all of the components in the WSMX architecture are used to achieve the task represented by the entry point interface. This includes a description of which components must be used, the order in which the components should be invoked and the control and data flow between the components. It is similar to the description of a workflow. The advantage in having multiple possible execution semantics, for the DIP architecture, is that it allows the dynamic behaviour of the architecture to be specified independently of the implementation of the functional components. In this context, dynamic behaviour means that the architecture can support a number of different behaviours based on the same set of functional components. The functional components of the DIP architecture are described in [1].

Table 1: Java Package Descriptions

Java Package	Description
org.wsmo.execution.common	The interfaces that provide system level access to the DIP architecture.
org.wsmo.execution.common.component	The interfaces to each of the components available in the DIP architecture.
org.wsmo.execution.common.component.resourcemanager	The interfaces that allow the WSMO description of Web Services, Goals, Mediators and Ontologies to be published directly in the DIP architecture.
org.wsmo.execution.common.component.exception	Exceptions that are used by the other interfaces in the DIP API.
org.wsmo.execution.common.nonwsmo	Helper classes that are outside the WSMO object model (WSMO4j) but are still necessary for the API.
org.wsmo.common.* and org.wsmo.service.*	The Java Packages denoted by the prefixes, org.wsmo.common and org.wsmo.service are taken from the open-source WSMO4j project, available at http://wsmo4j.sourceforge.net/ . These packages are not explicitly described in this deliverable.

The second viewpoint is that the API provides the Java description of the interfaces that each functional component in the DIP architecture must implement. This allows the public view of the behaviour of each component to be agreed between the partners involved in DIP technical work packages without them necessarily needing to know how the functionality will be implemented. The major benefit here is that the components that work together to form the architecture can be decoupled from each other. There are still dependencies but these are at the behaviour level between the component interfaces rather than hard-coded in the implementation details of the components themselves. In this way, the DIP architecture follows the style of service oriented architecture (SOA).

Java packages are used to provide the structure for the API. A summary of the Java packages used for the DIP API is given in the table 1.

2 CONTENT

The content of this deliverable is in two parts. The first part is this document which provides the mandatory details for a DIP deliverable as laid down in the consortium agreement. The second part of the deliverable is a set of HTML pages describing the API in detail. The rationale for this split is explained in the next paragraph.

The standard documentation tool, JavaDoc, has been used to extract the comments and descriptions embedded directly in the Java code. The output of JavaDoc is a set of related HTML pages that provide an intuitive means of browsing through the API. This is the standard way to document a Java API and is available at the following persistent URL: <https://bscw.dip.deri.ie/bscw/bscw.cgi/0/54228> .

REFERENCES

- [1] Zaremba, Michal, Moran M., Mocan, A., Cimpian, E., Haselwanter, T. and Zaremba, Maciej: (2005). DIP Deliverable D6.8 Revised Architecture.