



Project IST 026850 SUPER

Semantics Utilized for Process management within and between Enterprises

Deliverable 4.7

sBPEL to BPEL4SWS Lifting and Lowering

Leading Partner: USTUTT

Contributing Partner: USTUTT, Cefriel

Security Classification: Public (PU)

April, 2008

Version 0.7

Project	SUPER	SUPER-Project-No	026850
	sBPEL to BPEL4SWS Lifting and Lowering		
Document	Deliverable 4.7	Date	03.10.08

Project Details

IST Project Number	026850
Acronym	SUPER
Project Title	Semantics Utilised for Process management within and between EnteRprises
Project URL	http://www.ip-super.org
EU Project Officer	Werner Janusch

Authors (Partner)	Alessio Carenini (Cefriel), Jörg Nitzsche (USTUTT), Tammo van Lessen (USTUTT)		
Deliverable Owner (Partner)	Jörg Nitzsche (USTUTT)	E-mail	joerg.nitzsche@iaas.uni-stuttgart.de
		Phone	+49 (0) 711/7816486

Project	SUPER	SUPER-Project-No	026850
	SBPEL to BPEL4SWS Lifting and Lowering		
Document	Deliverable 4.7	Date	03.10.08

Versioning and Contribution History

Version	Description	Comments
0.1	Initial Input	JN
0.2	sBPEL to BPEL4SWS	JN
0.3	BPEL4SWS to SBPEL	AC
0.4	Peer Review	GH
0.5	Review Comments implemented except lifting	JN
0.6	Review Comments regarding lifting implemented	AC
0.7	Final Editing	JN

Project	SUPER	SUPER-Project-No	026850
	sBPEL to BPEL4SWS Lifting and Lowering		
Document	Deliverable 4.7	Date	03.10.08

Table of Contents

Project Details	II
Versioning and Contribution History	III
Table of Contents	IV
Table of Figures	V
Executive Summary	1
1 Introduction	2
2 Deliverable Alignment	2
2.1 Architecture Alignment	2
2.2 Methodology Alignment	3
2.3 Modelling Stack Alignment	3
2.4 Use Case Alignment	4
3 sBPEL vs. BPEL4SWS	5
4 BPEL(4SWS) to sBPEL	6
4.1 Purpose and Functionality	6
4.2 Transformation Description	6
4.2.1 Algorithm Cases: XML Node – WSML Concept	7
4.2.2 Algorithm Cases: XML Node – WSML Attribute Having a Built-in as Range	8
4.2.3 Algorithm Cases: XML Element – WSML Attribute Having a Concept as Range	8
4.2.4 Algorithm cases: XML Attribute – WSML Attribute Having a Concept as Range	8
5 sBPEL to BPEL(4SWS)	9
5.1 Purpose and Functionality	9
6 Usage	9
6.1 Availability and Contacts	9
6.2 Licensing	10
6.3 Requirements	10
6.3.1 Licensing of Third Party Libraries	10
6.4 Installation	11
6.4.1 Installation of the Lifting and Lowering Algorithms	11
6.4.2 Usage Example of the Lowering Algorithm	11
6.4.3 Usage Example of the Lifting Algorithm	13
6.4.4 Input/Output	14
7 References	21

Project	SUPER	SUPER-Project-No	026850
	sBPEL to BPEL4SWS Lifting and Lowering		
Document	Deliverable 4.7	Date	03.10.08

Table of Figures

Figure 1: SUPER Architecture Diagram	2
Figure 2: SUPER SBPM Lifecycle	3
Figure 1: SUPER Modelling Stack	4

IP- Project / Programme	SUPER	SUPER-Project-No	026850
	sBPEL to BPEL4SWS Lifting and Lowering	Work Package XX	
Document	Deliverable 4.7	Date	03.10.08

Executive Summary

This deliverable provides the translation from sBPEL to BPEL4SWS and vice versa. BPEL4SWS is an executable orchestration language. It is comprised of a set of specifications that in combination facilitate the orchestration of both, Web Services and Semantic Web Services. It uses an extension of BPEL that provides for an interaction model that is independent of WSDL and Semantic Web Service description frameworks like OWL-S and WSMO to specify the capabilities the process provides and the capabilities a process requires from its partners. Additionally it defines a grounding format to enable Web Service based communication with partner (Semantic) Web Services. sBPEL is an ontology that represents the knowledge captured in BPEL4SWS in an ontological form and thus makes BPEL accessible to ontological reasoning.

IP- Project / Programme	SUPER	SUPER-Project-No	026850
	sBPEL to BPEL4SWS Lifting and Lowering	Work Package XX	
Document	Deliverable 4.7	Date	03.10.08

1 Introduction

In SUPER processes are modelled using a stack of ontologies. For execution, a variant of the BPEL standard is used that facilitates the orchestration of both Web services and Semantic Web services. This extension is called BPEL4SWS and is defined as an extension to BPEL 2.0, thus it is based on XML. This deliverable bridges the gap between the modelling stack that is based on ontologies and the execution that's based on the industry standard BPEL by providing a translation between BPEL4SWS and its ontological representation sBPEL.

2 Deliverable Alignment

This deliverable focuses on the translation between BPEL4SWS specified in D1.10 and sBPEL defined in D1.1. The work is aligned with the other research areas in SUPER. In this section we present the alignment of the transformation with SUPER regarding the architecture, the methodology, the modelling stack and the use cases.

2.1 Architecture Alignment

The SUPER architecture showed in Figure 1 depicts SUPER Execution, SUPER Tooling, Semantic Service Bus (SSB), SUPER Services and SUPER Repositories. For more details please refer to Deliverable 7.2 "Semantic Web Services-based Business Process Architecture".

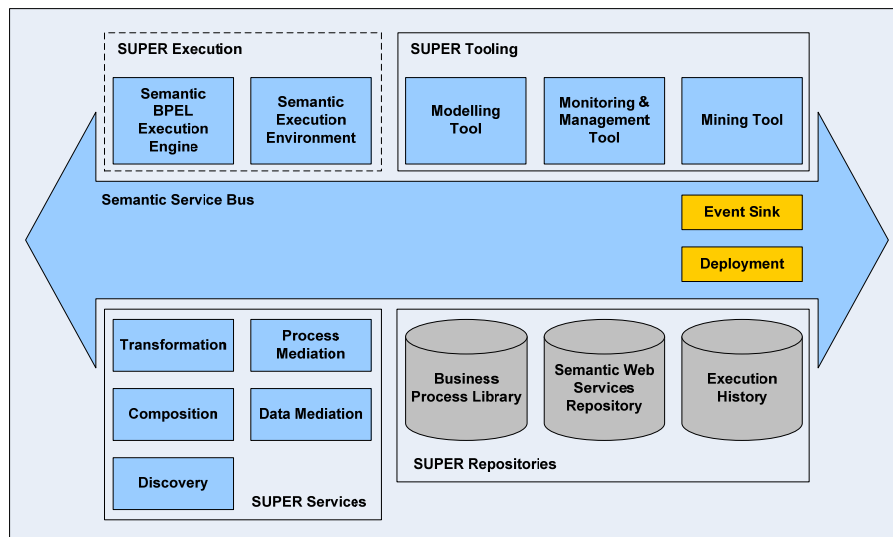


Figure 1: SUPER Architecture Diagram

This deliverable will provide a transformation service that enables both importing existing executable processes in to the modelling environment by lifting BPEL(4SWS) processes to the ontological level and translating ontological business processes expressed in sBPEL to the standard format of BPEL which is required to execute processes in the Semantic BPEL Execution Engine that is developed in WP6.

IP- Project / Programme	SUPER	SUPER-Project-No	026850
	sBPEL to BPEL4SWS Lifting and Lowering	Work Package XX	
Document	Deliverable 4.7	Date	03.10.08

2.2 Methodology Alignment

Figure 2 represents a symbolic depiction of the SUPER methodology for Semantic Business Process Management (SBPM). As depicted in the figure, the SBPM lifecycle is based on the Ontological Foundation and aims at supporting the Strategic Semantic Business Process Management. The Ontology Foundation is constituted by multiple ontologies, including the sBPEL ontology which is basically an ontological representation of BPEL4SWS. After the Configuration Phase, sBPEL processes are translated to BPEL4SWS and deployed to the Semantic BPEL Execution Engine. After the deployment, these process models are enacted during the execution phase.

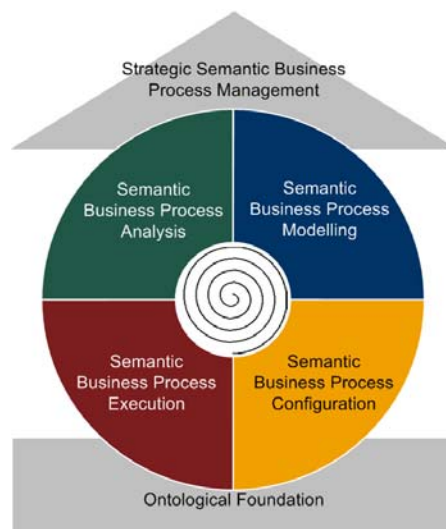


Figure 2: SUPER SBPM Lifecycle

2.3 Modelling Stack Alignment

Processes in SUPER will be represented using a set of ontologies. Figure 3 presents the SUPER ontology stack as described in D1.1. Process Modelling Ontology and Mapping to WSMO.

BPEL4SWS is a XML serialization of the sBPEL ontology. It is executed by the SBPELEE engine. In the scope of the project, a service requester does not model in BPEL4SWS directly, but in its ontological format. A bi-directional XML to WSML Transformation Service, or the ontological lifting and lowering accomplished in this deliverable, is utilized for executing mappings between the ontological level (sBPEL) defined in WSML and BPEL4SWS which is defined in XML Schema syntax.

IP- Project / Programme	SUPER	SUPER-Project-No	026850
	sBPEL to BPEL4SWS Lifting and Lowering	Work Package XX	
Document	Deliverable 4.7	Date	03.10.08

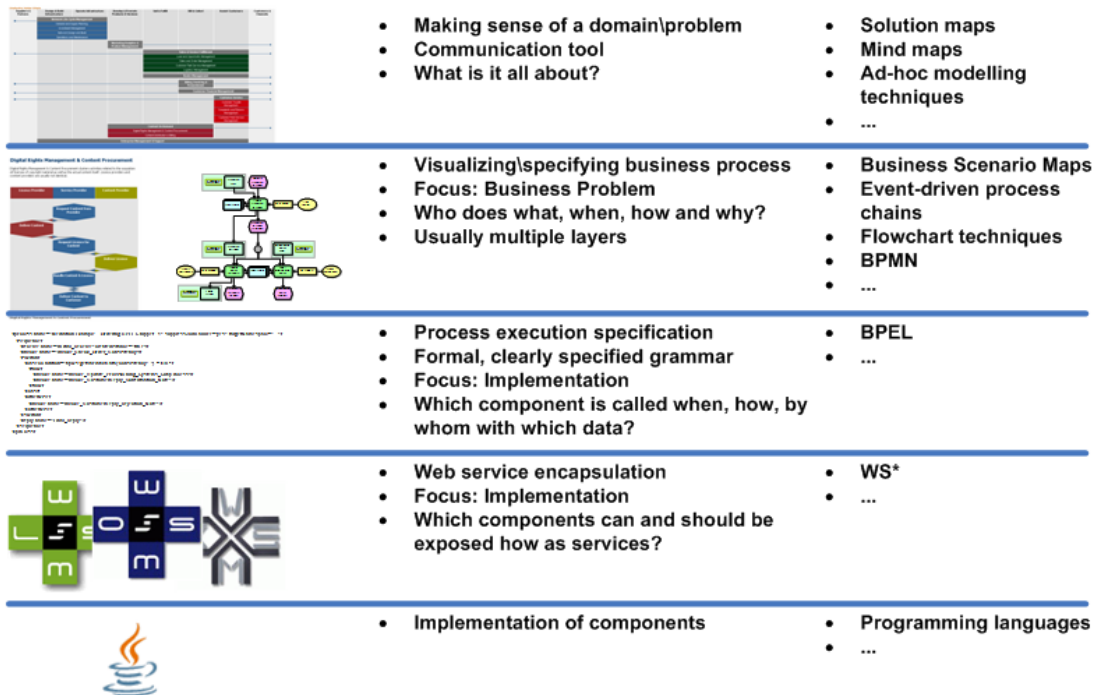


Figure 3: SUPER Modelling Stack

2.4 Use Case Alignment

BPEL is the de facto standard for describing Web Service flows. It is widely accepted and is already supported by major software vendors in their products. The Semantic BPEL Execution Engine, SBPELEE (WP 6), is based on an existing open-source BPEL engine, Apache ODE. SBPELEE executes BPEL4SWS models. The transformation from sBPEL to BPEL4SWS is one step when deploying process models that were created in the SUPER modelling suite, i.e. the transformation is required in all use cases in SUPER that include execution. The BPEL to sBPEL transformation facilitates importing BPEL processes into the SUPER modelling environment and thus fosters reuse of process models by providing access to a huge amount of existing BPEL processes. This transformation is helpful in all use cases in SUPER that include reuse of processes and process fragments.

IP- Project / Programme	SUPER	SUPER-Project-No	026850
	sBPEL to BPEL4SWS Lifting and Lowering	Work Package XX	
Document	Deliverable 4.7	Date	03.10.08

3 sBPEL vs. BPEL4SWS

sBPEL [1,2] is a WSML-flight ontology based on the XML schema defined by BPEL4SWS [3,4]. The general rule is to express each BPEL XML element as a class in the BPEL ontology and its corresponding attributes as attributes of the class. When possible, WSML built-in data types are reused as attributes (e.g. the strings “yes” and “no” are represented as values of type boolean rather than string). Cardinality constraints defined in the XML Schema definition are adopted as far as possible. This initial conceptualization is enriched by a formal description of information described in the specification in natural language form and not expressed in the XML Schema definition. Due to this semantic enrichment of the definition of the BPEL syntax, certain aspects of the specification are modelled differently to what their XML Schema definition suggests, resulting in an abstraction from the concrete specification of the language syntax. In the following, we give two examples where the structure of sBPEL significantly differs from the structure of BPEL4SWS which makes it impossible to use an automatic [5] lifting and lowering but requires manually defining and implementing a mapping between these two representations.

By following the generic approach of ontologizing the XML Schema definition of BPEL, unnecessary containers which do not represent any additional semantics are introduced. The FaultHandler element, for instance, is a container that only allows specifying multiple catch elements and one catchAll element to be used within a process or a scope. As the BPEL ontology only captures the semantics of the language and not syntax specific details, it abstracts from these, directly modelling the concepts Catch and CatchAll (which both inherit from FaultHandler) as attributes of the concept Process (Listing 1).

```
concept Process
  nonFunctionalProperties
    xmlns hasValue "http://docs.oasis-open.org/wsBPEL/2.0/Process/
      executable"
    dc:description hasValue "Concept of being a <process>-Element of an
      executable BPEL Process"
  endNonFunctionalProperties
  [...]
  hasCatch ofType Catch
  hasCatchAll ofType (0 1) CatchAll
  hasActivity ofType (1) Activity

concept FaultHandler
  nonFunctionalProperties
    dc:description hasValue "Concept of being a <faultHandler>-Element"
  endNonFunctionalProperties
  hasActivity ofType (1) Activity

concept Catch subConceptOf FaultHandler
  nonFunctionalProperties
    dc:description hasValue "Concept of being a <catch>-Element"
  endNonFunctionalProperties
  hasFaultName ofType (0 1) QName
  hasFaultVariable ofType (0 1) Variable
  hasFaultType ofType (0 1) DataType
```

IP- Project / Programme	SUPER	SUPER-Project-No	026850
	sBPEL to BPEL4SWS Lifting and Lowering	Work Package XX	
Document	Deliverable 4.7	Date	03.10.08

```
concept CatchAll subConceptOf FaultHandler
  nonFunctionalProperties
  dc:description hasValue "Concept of being a <catchAll>-Element"
endNonFunctionalProperties
```

Listing 1: Abstraction from XML Schema: Fault Handlers

In WSML there is no construct that allows expressing the order of elements. Since the BPEL element sequence defines a list of activities which are to be executed sequentially, and thus requires ordering of elements, the BPEL ontology caters to this issue, as is depicted in Listing 2. The BPEL activity sequence is represented as a concept with one attribute of type `OrderedActivity` which is a container with an attribute of type `Activity` and optionally another attribute of type `OrderedActivity`. This solution enables ordering elements and does not affect the implicit operational semantics of BPEL.

```
concept Sequence subConceptOf {StructuredActivity, standardAttributes}
  nonFunctionalProperties
  dc:description hasValue "Concept of being a <sequence>-Activity"
endNonFunctionalProperties
hasOrderedActivity ofType (1) OrderedActivity

concept OrderedActivity
  nonFunctionalProperties
  dc:description hasValue "concept of being an item of an ordered
  Activity list"
endNonFunctionalProperties
hasActivity ofType (1) Activity
hasOrderedActivity ofType (0 1) OrderedActivity
```

Listing 2: Ordered List of Activities

4 BPEL(4SWS) to sBPEL

4.1 Purpose and Functionality

The purpose of the lifting algorithm is to enable the transformation of processes described in BPEL4SWS to sBPEL instances (WSML syntax).

4.2 Transformation Description

BPEL4SWS, as stated in Section 2.3, is the XML Schema serialization of sBPEL. Comparing the XML and the ontological representation, we can see that both representations share the same terms, which are used as tag names in BPEL4SWS and as names of concepts and attributes in sBPEL. Moreover, BPEL4SWS contains explicit references to the ontological representation by using SA-WSDL attributes.

Klein's lifting algorithm [6] explores the possibility to use an existing ontology, resulting from the serialization of a set of concepts and relations using a formal language, as a way to give a meaning to tags in an XML document. The algorithm assumes the presence of a strong link between the XML document and the concepts in the ontology, the link being represented, for example, by the XML Schema being derived from an ontology. As the scenario represented by BPEL4SWS and sBPEL

IP- Project / Programme	SUPER	SUPER-Project-No	026850
	sBPEL to BPEL4SWS Lifting and Lowering	Work Package XX	
Document	Deliverable 4.7	Date	03.10.08

conforms to that assumption, it is possible to use the ideas expressed in such an algorithm in order to implement a generic, recursive algorithm that traverses the XML document and exploits the implicit link between XML node names and ontological entities in order to build an instance of the ontology itself.

The main limitations of the basic Klein's algorithm are:

- It does not completely exploit information in the ontology, as it does not check either the domain or the range of an attribute before creating the corresponding instances.
- It builds instances sequentially, so it can leave dangling triples in case of errors.

The extension of the algorithm implemented to lift BPEL4SWS to sBPEL, besides taking into account these limitations, makes use of the SA-WSDL annotations in BPEL4SWS as well as two heuristics used to cope with common patterns used when converting an XML Schema into an ontology, namely:

- Omission of sequence concepts: if the XML document has a tag used to contain a sequence, it is not converted in an ontological concept and its child nodes have to be considered directly linked to the parent node of the "sequence tag".
- Implicit nodes: when serializing ontological concepts in the XML Schema language, some links between concepts can be left as implicit (and thus not rendered) as a means to limit the number of tags in the document. In such a case, we are able to restore the missing link by using knowledge from the ontology.

The lifting algorithm operates recursively on the XML tree and performs a lookup for every XML node in the given ontology set to find which concept or attribute matches the XML tag name. When performing tag-ontological entity association discovery for an XML element, we also look whether there is a SA-WSDL annotation among its attributes or not. In the first case, the annotation is used to create the correct instance; otherwise we use the syntactical matching output.

Once the correct concept for the examined tag has been discovered, the correct instances are generated according to the possible types of an XML node and to the possible types of concepts in a WSML ontology, resulting in the following four cases:

- XML node - WSML concept, discussed in 4.3.1.
- XML node - WSML attribute having a built-in as range, discussed in 4.3.2.
- XML element - WSML attribute having a concept as range, reported in 4.3.3.
- XML attribute - WSML attribute having a concept as range, described in 4.3.4.

4.2.1 Algorithm Cases: XML Node – WSML Concept

When an XML node maps onto a concept in the ontology, a new concept instance is created. Before recurring on the child nodes, the concept instance is added to a list of active concepts, and is removed

IP- Project / Programme	SUPER	SUPER-Project-No	026850
	sBPEL to BPEL4SWS Lifting and Lowering	Work Package XX	
Document	Deliverable 4.7	Date	03.10.08

from the same list after the recursion has taken place. The purpose of the list is to offer context information while performing tag name-concept discovery.

4.2.2 Algorithm Cases: XML Node – WSML Attribute Having a Built-in as Range

When an XML corresponds to a WSML attribute whose range points to a WSML built-in in the ontology and its content is just text, a new attribute instance is created. When either the list of the active concepts is empty or the domain of the attribute does not match any of the active concepts, a new concept instance with an anonymous identifier is created before creating the attribute instance. Otherwise, it is used the concept instance in the most recent entry of the active concepts list that is listed also as the domain of the attribute.

4.2.3 Algorithm Cases: XML Element – WSML Attribute Having a Concept as Range

When an XML element maps onto a WSML attribute whose range points to a WSML concept in the ontology set, we have to verify the existence of a child node of the current node that corresponds to a concept and thus will trigger the creation of a new concept instance. If such a child node exists, it is processed as described in 4.3.1 and the newly created concept instance is used to create the attribute instance.

If a concept instance activator tag is not found, we can use a heuristic, assuming that the concept instance activator tag is implicit. When building the concept instance, we will act just as we had found the activator XML tag in generating the subject node identifier for the triple. In the following example, we can see a BPEL4SWS fragment (see Listing 3). The *process* tag links to the *Process* concept, while the *import* tag links to the *hasImport* attribute of *Process*. Once processed *import* as an ontology attribute, the tag is consumed, but the *hasImport* attribute requires a *Import* concept instance. Using the heuristic described before, the algorithm creates a new *Import* concept instance and continues processing the attributes of the XML element by adding that instance to the list of the active concepts.

```
<process name="ContentProvision">
  <import location="ContentProvision.wsdl"
           namespace="http://ip-super.org/processes/prereview/
ContentProvision.wsdl"
           importType="http://schemas.xmlsoap.org/wsdl/" />
```

Listing 3: BPEL4SWS Import Fragment

Independently from the presence of a concept instance trigger, the concept instance that is created is added to the list of the active concepts before recurring on the child nodes of the current node, and then is removed from the same list when the recursion ends.

4.2.4 Algorithm cases: XML Attribute – WSML Attribute Having a Concept as Range

When an XML attribute is linked to a WSML attribute whose range points to a WSML concept, we need to assign a textual value to a concept. According to the WSML model, this is clearly not possible,

IP- Project / Programme	SUPER	SUPER-Project-No	026850
	sBPEL to BPEL4SWS Lifting and Lowering	Work Package XX	
Document	Deliverable 4.7	Date	03.10.08

since it is only possible to assign textual values to an attribute. In this case we assume that by recursively (in breath-first order) navigating the attributes of the concepts, we will reach a concept that has a single attribute which points to a textual built-in. Once that built-in has been reached, we can proceed backwards by creating and linking instances of all the concepts and attributes of the path that links the starting attribute and the ending built-in.

5 sBPEL to BPEL(4SWS)

5.1 Purpose and Functionality

The purpose of the lowering algorithm is to enable the transformation of processes described in sBPEL to BPEL4SWS, a format that is compliant to BPEL 2.0.

6 Usage

6.1 Availability and Contacts

Version: 1.0, 31 March 2008.

Sources: <http://svn.sti2.at/super/WP4/D4.7/bpellilo>

Binaries and Source distribution:

<http://stronghold.sirma.bg/IP-SUPER/maven2/org/ipsuper/transformation/bpellilo/1.0-SNAPSHOT/>

Maven 2.x Artefacts:

Include the following pom fragment into your pom.xml:

```
<repositories>
  ...
  <repository>
    <id>Ontotext</id>
    <name>Ontotext Lab</name>
    <url>http://stronghold.sirma.bg/maven2</url>
    <layout>default</layout>
  </repository>
  ...
</repositories>
...
<dependencies>
  ...
  <dependency>
    <groupId>org.ipsuper.transformation.bpellilo</groupId>
    <artifactId>bpellilo</artifactId>
    <version>1.0-SNAPSHOT</version>
  </dependency>
  ...
</dependencies>
```

Contact person: Jörg Nitzsche, Tammo van Lessen, Alessio Carenini

IP- Project / Programme	SUPER	SUPER-Project-No	026850
	sBPEL to BPEL4SWS Lifting and Lowering	Work Package XX	
Document	Deliverable 4.7	Date	03.10.08

6.2 Licensing

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Lifting algorithm implementation Copyright © 2008, CEFRIEL S.C.R.L.

Lowering algorithm implementation Copyright © 2008, Institute of Architecture of Application Systems, University of Stuttgart.

6.3 Requirements

Nature: Java library

Interfaces (API, Web Services): a Java API.

Platform: JDK 5

Required Libraries:

- WSMO4J 0.6.1
- dom4j 1.6.1
- log4j 1.2.14
- JUnit 4.4 (for testing only)
- xmlunit 1.1 (for testing only)
- wsm12reasoner
- IRIS reasoner

6.3.1 Licensing of Third Party Libraries

Licensing of third party libraries and components required for the versioning library:

- [wsmo4j](#) - Copyright © Ontotext Lab, Sirma. It is an open-source library, available under [LGPL](#).
- [dom4j](#) - Copyright 2001-2005 © MetaStuff, Ltd. All Rights Reserved, available under a [BSD style license](#)
- [log4j](#) - Copyright 2007 © The Apache Software Foundation, available under the [Apache License 2.0](#)

IP- Project / Programme	SUPER	SUPER-Project-No	026850
	sBPEL to BPEL4SWS Lifting and Lowering	Work Package XX	
Document	Deliverable 4.7	Date	03.10.08

- wsml2reasoner – Copyright (c) 2005, University of Innsbruck, Austria. It is available under [LGPL](#).
- IRIS reasoner – Copyright (C) 2007 Digital Enterprise Research Institute (DERI). It is available under [LGPL](#).

6.4 Installation

6.4.1 Installation of the Lifting and Lowering Algorithms

The library file sbpellilo.jar in the installation package needs to be in the classpath of the application that wishes to use it, together with all mentioned required third party libraries.

6.4.2 Usage Example of the Lowering Algorithm

The following is an excerpt from a simple demo program that uses the lowering algorithm:

```
package de.unistuttgart.iaas.ipsuper.sbpello;

import java.io.File;
import java.io.FileReader;
import java.io.IOException;
import java.io.StringWriter;
import java.util.HashMap;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

import org.apache.commons.io.FileUtils;
import org.apache.log4j.BasicConfigurator;
import org.custommonkey.xmlunit.XMLAssert;
import org.junit.Before;
import org.junit.Ignore;
import org.junit.Test;
import org.omg.ontology.Concept;
import org.omg.ontology.Ontology;
import org.w3c.dom.Element;
import org.wsmo.common.IRI;
import org.wsmo.common.TopEntity;
import org.wsmo.common.exception.InvalidModelException;
import org.wsmo.factory.DataFactory;
import org.wsmo.factory.Factory;
import org.wsmo.factory.LogicalExpressionFactory;
import org.wsmo.factory.WsmoFactory;
import org.wsmo.wsml.Parser;
import org.wsmo.wsml.ParserException;

import com.sun.org.apache.xml.internal.serialize.OutputFormat;
import com.sun.org.apache.xml.internal.serialize.XMLSerializer;

import de.unistuttgart.iaas.ipsuper.sbpello.SBPello;

/**
 * A test case.
 *
 * @author Tammo van Lessen (University of Stuttgart)
 * @author Jörg Nitzsche (University of Stuttgart)
 */
```

IP- Project / Programme	SUPER	SUPER-Project-No	026850
	sBPEL to BPEL4SWS Lifting and Lowering	Work Package XX	
Document	Deliverable 4.7	Date	03.10.08

```

*/
public class BPELLoweringTest {

    private WsmoFactory factory;
    private LogicalExpressionFactory leFactory;
    private DataFactory dataFactory;
    private Parser parser;
    private SBPELlO lowerer;

    @Before public void setup() {
        HashMap <String, Object> props = new HashMap <String, Object>();

        // use default implementation for factory
        factory = Factory.createWsmoFactory(null);
        leFactory = Factory.createLogicalExpressionFactory(null);
        dataFactory = Factory.createDataFactory(null);

        props.put(Factory.WSMO_FACTORY, factory);
        props.put(Factory.LE_FACTORY, leFactory);
        props.put(Factory.DATA_FACTORY, dataFactory);

        parser = Factory.createParser(props);

        lowerer = new SBPELlO();
    }

    private TopEntity[] parseOntology(File f) throws IOException,
        InvalidModelException {
        FileReader reader = null;
        try {
            reader = new FileReader(f);
            return parser.parse(reader);
        }
        catch (ParserException ex) {
            throw new RuntimeException("Invalid WSMML token encountered at
                line " + ex.getErrorLine() + " position " +
                ex.getErrorPos(), ex);
        } finally {
            if (reader != null) {
                reader.close();
            }
        }
    }

    @Test
    public void testLoweringFlowWithLinks() throws Exception {
        String testfile = "/20070917_BPELsample_FlowWithLinks";
        TopEntity[] sbpel = parseOntology(new File(getClass().
            getResource(testfile + ".wsml").toURI()));
        String test = serialize(lowerer.lowerSBPEL((Ontology) sbpel[0]));
        String control = FileUtils.readFileToString(new File(getClass().
            getResource(testfile + ".bpel").toURI()), null);
        System.out.println(test);
        XMLAssert.assertXMLEqual(control, test);
    }

    private String serialize(Element el) throws IOException {
        StringWriter sw = new StringWriter();
        OutputFormat outformat = new OutputFormat();
        outformat.setIndenting(true);
    }

```

IP- Project / Programme	SUPER	SUPER-Project-No	026850
	sBPEL to BPEL4SWS Lifting and Lowering	Work Package XX	
Document	Deliverable 4.7	Date	03.10.08

```

XMLSerializer serializer = new XMLSerializer(sw, outformat);
serializer.serialize(e1);
return sw.toString();
}
}

```

6.4.3 Usage Example of the Lifting Algorithm

The following is an excerpt from a simple demo program that uses the lifting algorithm.

```

package it.cefriel.ipsuper.lift;

import java.io.File;
import java.io.IOException;
import java.io.InputStream;
import java.io.StringWriter;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;

import javax.xml.parsers.DocumentBuilderFactory;

import org.apache.commons.io.FileUtils;
import org.junit.Assert;
import org.junit.Ignore;
import org.junit.Test;
import org.omg.ontology.Ontology;
import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.wsmo.common.TopEntity;
import org.wsmo.factory.Factory;
import org.wsmo.wsml.Serializer;

/**
 * A test case.
 *
 * @author Alessio Carenini (Cefriel)
 */
public class BPELLiftingTest {

    @Test @Ignore
    public void testBPELLifting () throws Exception {

        List<String> ontologies = new ArrayList<String>();
        ontologies.add(getClass().getResource("/20080227_BPEL2_0.wsml").
            toString());
        ontologies.add(getClass().getResource("/20080227_sBPEL.wsml").
            toString());
        ontologies.add(getClass().getResource
            ("/WSDLExtension4BPEL20070126.wsml").toString());

        Ontology lifted = new SBPELLi().lift(parseXmlFile(getClass()
            .getResourceAsStream("/20070917_BPELsample_Blockoriented.bpel"));
        String output = ontologyToString(lifted);
        //TODO: normalize control data to have exactly the same serialization
        // as the output
        String control = FileUtils.readFileToString(new File(getClass().
            getResource("/20070917_BPELsample_Blockoriented.wsml").toURI()), null);
        Assert.assertEquals(control, output);
    }
}

```

IP- Project / Programme	SUPER	SUPER-Project-No	026850
	sBPEL to BPEL4SWS Lifting and Lowering	Work Package XX	
Document	Deliverable 4.7	Date	03.10.08

```

    }

    private Element parseXmlFile(InputStream is) {
        try {
            DocumentBuilderFactory factory = DocumentBuilderFactory.
                newInstance();
            Document doc = factory.newDocumentBuilder().parse(is);
            return doc.getDocumentElement();
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }

    private String ontologyToString(Ontology o) {
        Serializer serializer=Factory.createSerializer(new
            HashMap<String, Object>(0));
        try {
            StringWriter writer = new StringWriter();
            serializer.serialize(new TopEntity[] {o}, writer);
            return writer.toString();
        }
        catch (IOException ioe) {
            throw new RuntimeException(ioe);
        }
    }
}

```

6.4.4 Input/Output

Input of the Lowering / Output of the Lifting:

```

wsmlVariant _"http://www.wsmo.org/wsml/wsml-syntax/wsml-flight"
namespace { _"http://www.ip-
super.org/ontologies/sBPEL/serviceProvision_FlowWithLinks#",
    bpel _"http://www.ip-super.org/ontologies/BPEL2_0/20080227#",
    sbpel _"http://www.ip-super.org/ontologies/sBPEL/20080227#",
    wsdlx _"http://www.ip-
super.org/ontologies/WSDLExtension4BPEL/20070126#",
    dc _"http://purl.org/dc/elements/1.1#" }

```

ontology sbpelProcess

```

    nonFunctionalProperties
        dc#title hasValue "sample sBPEL Process"
        dc#description hasValue "sBPEL Process of service provision"
        dc#publisher hasValue "SUPER European Integrated Project"
        dc#subject hasValue {"sBPEL", "business process", "workflow"}
        dc#language hasValue "en-UK"
        dc#date hasValue "$Date: 2007/04/04$"
    endNonFunctionalProperties

```

```

    importsOntology
        { _"http://www.ip-super.org/ontologies/BPEL2_0/20080227#",
          _"http://www.ip-super.org/ontologies/sBPEL/20080227#",
          _"http://www.ip-
super.org/ontologies/wsdlExtension4BPEL/20070126#" }

```

concept reqGetLicense

IP- Project / Programme	SUPER	SUPER-Project-No	026850
	sBPEL to BPEL4SWS Lifting and Lowering	Work Package XX	
Document	Deliverable 4.7	Date	03.10.08

```

    requestedContent ofType (1) _string
    byUser ofType (1) _string

concept resGetLicense
  agreement ofType (1) _string

concept reqGetURL
  requestedContent ofType (1) _string
  byUser ofType (1) _string

concept resGetURL
  url ofType (1) _iri

instance procContentProvision memberOf sbpel#SemanticProcess
  hasName hasValue "ContentProvision"
  hasTargetNamespace hasValue "http://www.ip-
super.org/ontologies/prereview"
  hasPartnerLink hasValue serviceProviderPL
  hasConversations hasValue {convLicenseServer, convPackager}
  hasActivity hasValue processFlow
  hasVariable hasValue {varContentRequest, varContentLicenseReq,
varContentLicenseRes, varContentURLReq, varContentURLRes, varResult}

instance varContentRequest memberOf bpel#Variable
  hasName hasValue "contentRequest"
  hasType hasValue varContentType

instance varContentType memberOf bpel#WSDLMessageType
  hasDefinition hasValue "http://ip-
super.org/processes/ContentProvision.wsdl#contentRequestMessage"

instance varContentLicenseReq memberOf sbpel#SemanticVariable
  hasName hasValue "contentLicenseReq"
  hasType hasValue varContentLicenseReqType
  hasSemanticType hasValue reqGetLicense

instance varContentLicenseReqType memberOf bpel#XSDElement
  hasDefinition hasValue "http://ip-
super.org/processes/ContentProvision.wsdl#contentRequestMessage"

instance varContentURLReq memberOf sbpel#SemanticVariable
  hasName hasValue "contentURLReq"
  hasType hasValue varContentURLReqType
  hasSemanticType hasValue reqGetURL

instance varContentURLReqType memberOf bpel#XSDType
  hasDefinition hasValue "http://ip-
super.org/processes/ContentProvision.wsdl#contentRequestMessage"

instance varContentLicenseRes memberOf sbpel#SemanticVariable
  hasName hasValue "contentLicenseRes"
  hasType hasValue varContentLicenseResType
  hasSemanticType hasValue resGetLicense

instance varContentLicenseResType memberOf bpel#XSDElement
  hasDefinition hasValue "http://ip-
super.org/processes/ContentProvision.wsdl#tContentLicense"

instance varContentURLRes memberOf sbpel#SemanticVariable

```

IP- Project / Programme	SUPER	SUPER-Project-No	026850
	sBPEL to BPEL4SWS Lifting and Lowering	Work Package XX	
Document	Deliverable 4.7	Date	03.10.08

```

hasName hasValue "contentURLRes"
hasType hasValue varContentURLResType
hasSemanticType hasValue resGetURL

instance varContentURLResType memberOf bpel#XSDType
hasDefinition hasValue "http://ip-
super.org/processes/ContentProvision.wsdl#tContentURL"

instance varResult memberOf bpel#Variable
hasName hasValue "result"
hasType hasValue varResultType

instance varResultType memberOf bpel#WSDLMessageType
hasDefinition hasValue "http://ip-
super.org/processes/ContentProvision.wsdl#ContentResponseMessage"

instance serviceProviderPL memberOf _"http://www.ip-
super.org/ontologies/sBPEL/20070308#PartnerLink"
hasName hasValue "ServiceProvider"
hasPartnerLinkType hasValue serviceProviderPLT
hasMyRole hasValue provider

instance serviceProviderPLT memberOf wsdlx#PartnerLinkType
hasName hasValue "serviceProviderPLT"
hasRole hasValue provider

instance provider memberOf wsdlx#Role
hasName hasValue "provider"
hasPortType hasValue "http://ip-
super.org/processes/ContentProvision.wsdl#ServiceProviderPortType"

instance convLicenseServer memberOf sbpel#Conversation
hasName hasValue "LicenseServer"
describesInterface hasValue licenseServerInterface

instance licenseServerInterface memberOf sbpel#OutgoingInterface
hasGoalDescription hasValue
_"http://irs.open.ac.uk/superPrereview#goalGetLicense"

instance convPackager memberOf sbpel#Conversation
hasName hasValue "Packager"
describesInterface hasValue packagerInterface

instance packagerInterface memberOf sbpel#OutgoingInterface
hasGoalDescription hasValue goalGetURL

instance processFlow memberOf bpel#Flow
hasActivity hasValue {prepareRequest, recContentRequest,
actGoalGenerateLicense, actGoalGenerateURL, actAggregateResult,
repGetContent}
hasLink hasValue {receiveToPrepareRequest, assignToGenerateURL,
assignToGenerateLicense, generateLicenseToAssign, generateURLToAssign,
assignToReply}

instance recContentRequest memberOf bpel#Receive
hasName hasValue "receiveContentRequest"
doesCreateInstance hasValue true
hasVariable hasValue varContentRequest
hasPartnerLink hasValue serviceProviderPL

```

IP- Project / Programme	SUPER	SUPER-Project-No	026850
	sBPEL to BPEL4SWS Lifting and Lowering	Work Package XX	
Document	Deliverable 4.7	Date	03.10.08

```

hasPortType hasValue "http://ip-
super.org/processes/ContentProvision.wsdl#ServiceProviderPortType"
hasOperation hasValue "requestContent"
isSource hasValue recContentRequestLinks

instance recContentRequestLinks memberOf bpel#OrderedLink
hasLink hasValue receiveToPrepareRequest

instance receiveToPrepareRequest memberOf bpel#Link
hasName hasValue "receiveToPrepareRequest"

instance assignToGenerateLicense memberOf bpel#Link
hasName hasValue "assignToGenerateLicense"

instance assignToGenerateURL memberOf bpel#Link
hasName hasValue "assignToGenerateURL"

instance generateLicenseToAssign memberOf bpel#Link
hasName hasValue "generateLicenseToAssign"

instance generateURLToAssign memberOf bpel#Link
hasName hasValue "generateURLToAssign"

instance assignToReply memberOf bpel#Link
hasName hasValue "assignToReply"

instance repGetContent memberOf bpel#Reply
hasName hasValue "reply"
hasVariable hasValue varResult
hasPartnerLink hasValue serviceProviderPL
hasPortType hasValue "http://ip-
super.org/processes/ContentProvision.wsdl#ServiceProviderPortType"
hasOperation hasValue "requestContent"
isTarget hasValue assignToReply

instance actGoalGenerateLicense memberOf bpel#ExtensionActivity
hasActivity hasValue invokeGoalGenerateLicense

instance invokeGoalGenerateLicense memberOf sbpel#SendReceive
hasName hasValue "invokeGoalGenerateLicense"
belongsToConversation hasValue convPackager
hasInputVariable hasValue varContentLicenseReq
hasOutputVariable hasValue varContentLicenseRes
isTarget hasValue assignToGenerateLicense
isSource hasValue invokeGoalGenerateLicenseLinks

instance invokeGoalGenerateLicenseLinks memberOf bpel#OrderedLink
hasLink hasValue generateLicenseToAssign

instance actGoalGenerateURL memberOf bpel#ExtensionActivity
hasActivity hasValue invokeGoalGenerateURL

instance invokeGoalGenerateURL memberOf sbpel#SendReceive
hasName hasValue "invokeGoalGenerateURL"
belongsToConversation hasValue convLicenseServer
hasInputVariable hasValue varContentURLReq
hasOutputVariable hasValue varContentURLRes
isTarget hasValue assignToGenerateURL
isSource hasValue invokeGoalGenerateURLLinks

```

IP- Project / Programme	SUPER	SUPER-Project-No	026850
	sBPEL to BPEL4SWS Lifting and Lowering	Work Package XX	
Document	Deliverable 4.7	Date	03.10.08

```

instance invokeGoalGenerateURLLinks memberOf bpel#OrderedLink
  hasLink hasValue generateURLToAssign

instance actAggregateResult memberOf bpel#Assign
  hasName hasValue "aggregateResult"
  hasAssignOperation hasValue {copyURL, copyLicense}
  isTarget hasValue {generateURLToAssign, generateLicenseToAssign}
  isSource hasValue aggregateResultLinks

instance aggregateResultLinks memberOf bpel#OrderedLink
  hasLink hasValue assignToReply

instance copyURL memberOf bpel#Copy
  hasFromSpecification hasValue fromContentURL
  hasToSpecification hasValue toSpecification

instance copyLicense memberOf bpel#Copy
  hasFromSpecification hasValue fromContentLicense
  hasToSpecification hasValue toSpecification

instance fromContentURL memberOf bpel#CopyExpression
  hasExpression hasValue "$contentURLRes"

instance fromContentLicense memberOf bpel#CopyExpression
  hasExpression hasValue "$contentLicenseRes"

instance toVariableResultURL memberOf bpel#CopyVariablePart
  hasVariable hasValue varResult
  hasPart hasValue "http://ip-
super.org/processes/ContentProvision.wsdl#wsdl11.messagePart (ContentRespons
eMessage,URL) "

instance toVariableResultLicense memberOf bpel#CopyVariablePart
  hasVariable hasValue varResult
  hasPart hasValue "http://ip-
super.org/processes/ContentProvision.wsdl#wsdl11.messagePart (ContentRespons
eMessage,license) "

instance prepareRequest memberOf bpel#Assign
  hasName hasValue "prepareRequest"
  hasAssignOperation hasValue {copyURLReq, copyLicenseReq}
  isSource hasValue prepareRequestLinks
  isTarget hasValue receiveToPrepareRequest

instance prepareRequestLinks memberOf bpel#OrderedLink
  hasLink hasValue assignToGenerateLicense
  hasOrderedLink hasValue bla

instance bla memberOf bpel#OrderedLink
  hasLink hasValue assignToGenerateURL

instance copyURLReq memberOf bpel#Copy
  hasFromSpecification hasValue fromContentReq
  hasToSpecification hasValue toContentURLReq

instance copyLicenseReq memberOf bpel#Copy
  hasFromSpecification hasValue fromContentReq
  hasToSpecification hasValue toContentLicenseReq

instance fromContentReq memberOf bpel#CopyVariablePart

```

IP- Project / Programme	SUPER	SUPER-Project-No	026850
	sBPEL to BPPEL4SWS Lifting and Lowering	Work Package XX	
Document	Deliverable 4.7	Date	03.10.08

hasVariable hasValue varContentRequest

instance toContentURLReq memberOf bpel#CopyVariablePart
hasVariable hasValue varContentURLReq

instance toContentLicenseReq memberOf bpel#CopyVariablePart
hasVariable hasValue varContentLicenseReq

Output of the Lowering / Input of the Lifting:

```
<bpws:process
  xmlns:bpws="http://docs.oasis-open.org/wsbpel/2.0/process/executable"
  name="ContentProvision"
  targetNamespace="http://www.ip-super.org/ontologies/prereview"
  xmlns:sa="http://www.w3.org/2002/ws/sawsdl/spec/sawsdl#"
  sa:modelReference="http://www.ip-super.org/ontologies/sBPEL/
    serviceProvision_FlowWithLinks#procContentProvision">
  <bpws:partnerLinks>
    <bpws:partnerLink name="ServiceProvider"
      partnerLinkType="serviceProviderPLT"
      myRole="provider"/>
  </bpws:partnerLinks>
  <bpws:variables>
    <bpws:variable name="contentRequest" messageType="http://ip-
super.org/processes/ContentProvision.wsdl#contentRequestMessage"/>
    <bpws:variable name="contentLicenseReq" element="http://ip-
super.org/processes/ContentProvision.wsdl#contentRequestMessage"/>
    <bpws:variable name="contentLicenseRes" element="http://ip-
super.org/processes/ContentProvision.wsdl#tContentLicense"/>
    <bpws:variable name="contentURLReq" type="http://ip-
super.org/processes/ContentProvision.wsdl#contentRequestMessage"/>
    <bpws:variable name="contentURLRes" type="http://ip-
super.org/processes/ContentProvision.wsdl#tContentURL"/>
    <bpws:variable name="result" messageType="http://ip-
super.org/processes/ContentProvision.wsdl#ContentResponseMessage"/>
  </bpws:variables>
  <b4s:conversations xmlns:b4s="http://ip-super.org/process/BPEL4SWS">
    <b4s:conversation
      name="LicenseServer"
      requestedFunctionality="http://irs.open.ac.uk/
        superPrereview#goalGetLicense"
      sa:modelReference="http://www.ip-super.org/ontologies/
        sBPEL/serviceProvision_FlowWithLinks#
        convLicenseServer"/>
    <b4s:conversation
      name="Packager"
      requestedFunctionality="http://www.ip-super.org/ontologies/
        sBPEL/serviceProvision_FlowWithLinks
        #goalGetURL"
      sa:modelReference="http://www.ip-super.org/ontologies/
        sBPEL/serviceProvision_FlowWithLinks#
        convPackager"/>
  </b4s:conversations>
  <bpws:flow
    sa:modelReference="http://www.ip-super.org/ontologies/
      sBPEL/serviceProvision_FlowWithLinks#processFlow">
  <bpws:links>
    <bpws:link name="receiveToPrepareRequest"/>
    <bpws:link name="assignToGenerateURL"/>
    <bpws:link name="assignToGenerateLicense"/>
  </bpws:links>
</bpws:process>
```

IP- Project / Programme	SUPER	SUPER-Project-No	026850
	sBPEL to BPEL4SWS Lifting and Lowering	Work Package XX	
Document	Deliverable 4.7	Date	03.10.08

```

    <bpws:link name="generateLicenseToAssign"/>
    <bpws:link name="generateURLToAssign"/>
    <bpws:link name="assignToReply"/>
  </bpws:links>
  <bpws:assign
    name="prepareRequest"
    sa:modelReference="http://www.ip-super.org/ontologies/
      sBPEL/serviceProvision_FlowWithLinks#
      prepareRequest">
    <bpws:targets>
      <bpws:target linkName="receiveToPrepareRequest"/>
    </bpws:targets>
    <bpws:sources>
      <bpws:source linkName="assignToGenerateLicense"/>
    </bpws:sources>
    <bpws:copy>
      <bpws:from variable="contentRequest"/>
      <bpws:to variable="contentURLReq"/>
    </bpws:copy>
    <bpws:copy>
      <bpws:from variable="contentRequest"/>
      <bpws:to variable="contentLicenseReq"/>
    </bpws:copy>
  </bpws:assign>
  <bpws:receive
    partnerLink="ServiceProvider"
    portType="http://ip-super.org/processes/ContentProvision.
      wsdl#ServiceProviderPortType" operation="requestContent"
    variable="contentRequest"
    createInstance="yes"
    name="receiveContentRequest"
    sa:modelReference="http://www.ip-super.org/ontologies/sBPEL/
      serviceProvision_FlowWithLinks#
      recContentRequest">
    <bpws:sources>
      <bpws:source linkName="receiveToPrepareRequest"/>
    </bpws:sources>
  </bpws:receive>
  <bpws:extensionActivity>
    <b4s:interactionActivity
      xmlns:b4s=http://ip-super.org/process/BPEL4SWS
      conversation="Packager"
      inputVariable="contentLicenseReq"
      outputVariable="contentLicenseRes"
      name="invokeGoalGenerateLicense"
      sa:modelReference="http://www.ip-super.org/ontologies/sBPEL/
        serviceProvision_FlowWithLinks#
        invokeGoalGenerateLicense">
      <bpws:targets>
        <bpws:target linkName="assignToGenerateLicense"/>
      </bpws:targets>
      <bpws:sources>
        <bpws:source linkName="generateLicenseToAssign"/>
      </bpws:sources>
    </b4s:interactionActivity>
  </bpws:extensionActivity>
  <bpws:extensionActivity>
    <b4s:interactionActivity
      xmlns:b4s=http://ip-super.org/process/BPEL4SWS

```

IP- Project / Programme	SUPER	SUPER-Project-No	026850
	sBPEL to BPEL4SWS Lifting and Lowering	Work Package XX	
Document	Deliverable 4.7	Date	03.10.08

```

        conversation="LicenseServer"
        inputVariable="contentURLReq"
        outputVariable="contentURLRes"
        name="invokeGoalGenerateURL"
        sa:modelReference="http://www.ip-super.org/ontologies/sBPEL/
            serviceProvision_FlowWithLinks#
            invokeGoalGenerateURL">
    <bpws:targets>
        <bpws:target linkName="assignToGenerateURL"/>
    </bpws:targets>
    <bpws:sources>
        <bpws:source linkName="generateURLToAssign"/>
    </bpws:sources>
    </b4s:interactionActivity>
</bpws:extensionActivity>
<bpws:assign
    name="aggregateResult"
    sa:modelReference="http://www.ip-super.org/ontologies/sBPEL/
        serviceProvision_FlowWithLinks#
        actAggregateResult">
    <bpws:targets>
        <bpws:target linkName="generateURLToAssign"/>
        <bpws:target linkName="generateLicenseToAssign"/>
    </bpws:targets>
    <bpws:sources>
        <bpws:source linkName="assignToReply"/>
    </bpws:sources>
    <bpws:copy>
        <bpws:from>$contentURLRes</bpws:from>
        <bpws:to/>
    </bpws:copy>
    <bpws:copy>
        <bpws:from>$contentLicenseRes</bpws:from>
        <bpws:to/>
    </bpws:copy>
</bpws:assign>
<bpws:reply
    partnerLink="ServiceProvider"
    portType="http://ip-super.org/processes/ContentProvision.
        wsdl#ServiceProviderPortType"
    operation="requestContent"
    variable="result"
    name="reply"
    sa:modelReference="http://www.ip-super.org/ontologies/sBPEL/
        serviceProvision_FlowWithLinks#repGetContent">
    <bpws:targets>
        <bpws:target linkName="assignToReply"/>
    </bpws:targets>
</bpws:reply>
</bpws:flow>
</bpws:process>

```

7 References

- [1] Roxana Belecheanu, Liliana Cabral, John Domingue, Walid Gaaloul, Martin Hepp, Agata Filipowska, Monika Kaczmarek, Tomasz Kaczmarek, Jörg Nitzsche, Barry Norton, Carlos

IP- Project / Programme	SUPER	SUPER-Project-No	026850
	sBPEL to BPEL4SWS Lifting and Lowering	Work Package XX	
Document	Deliverable 4.7	Date	03.10.08

Pedrinaci, Dumitru Roman, Michael Stollberg, Sebastian Stein. Business Process Ontology Framework. SUPER Project Deliverable D1.1, September 2006.

- [2] Jörg Nitzsche, Daniel Wutke, Tammo van Lessen. An Ontology for Executable Business Processes. [Workshop on Semantic Business Process and Product Lifecycle Management \(SBPM 2007\)](#), in conjunction with [ESWC 2007](#). Innsbruck, Austria, June 7, 2007.
- [3] Jörg Nitzsche, Tammo van Lessen. BPEL4SWS – final version. SUPER project deliverable D1.10, March 2008.
- [4] Jörg Nitzsche, Tammo van Lessen, Dimka Karastoyanova, Frank Leymann.: BPEL for Semantic Web Services. Proceedings of the 3rd International Workshop on Agents and Web Services in Distributed Environments (AWeSome'07), November 2007.
- [5] E. Miller and CM Sperberg-McQueen. On mapping from colloquial XML to RDF using XSLT. *Proc. of W3C Extreme Markup Languages*, 2004.
- [6] Michel Klein. Interpreting XML documents via an RDF Schema ontology. *dexa*, 00:889, 2002.
- [7] S. Battle. Gloze: XML to RDF and back again. *2006 Jena User Conference*, 2006.
- [8] M. Ferdinand, C. Zircins, and D. Trastour. Lifting XML Schema to OWL. *Proceedings of 4th International Conference on Web Engineering (ICWE4)*, pages 26–30.
- [9] J. Kopeck, T. Vitvar, C. Bournez, and J. Farrell. SAWSDL: Semantic Annotations for WSDL and XML Schema. *IEEE Internet Computing*, 11(6):60–67, 2007.