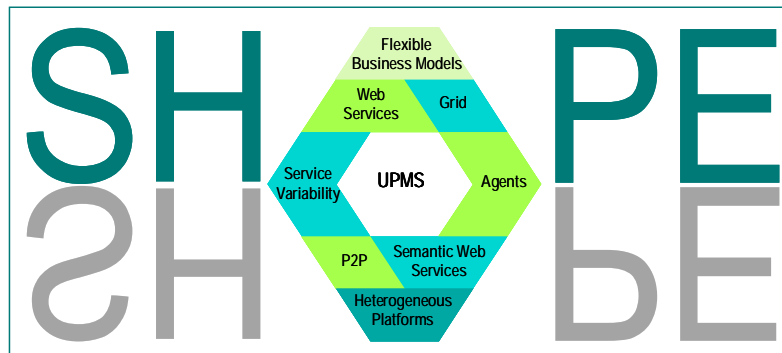


Service and Software Architectures, Infrastructures and Engineering
Small or Medium-scale Focused Research Project
Semantically-enabled Heterogeneous Service Architecture and Platforms Engineering
Acronym
SHAPE
Project No
216408



Deliverable D4.1

MDE Toolset and Architecture Specification

Work Package # 4

Leading partner: SOFTEAM

Editor: Andrey Sadovykh

Authors: A. Sadovykh, Ph. Desfray, Ch. Hahn, G. Benguria

Dissemination level: Public

Date:

Version: 1.0 - final version

Versioning and contribution history

Version	Description	Contributors
0.1	Initial Version	A. Sadovykh
0.2	Drivers section contributions	A. Sadovykh
0.3	Sequence diagrams added to illustrate case study tool chains	A. Sadovykh, Ph. Desfray, Christian Hahn, Gorka Benguria
0.4	“Tool chains” were refined, Integration Architecture was added.	A. Sadovykh
0.5	“Addressing Main Drivers” section was added	A. Sadovykh
0.6	Comments from Reviewers were addressed	A. Sadovykh
1.0	Final Editing	A. Sadovykh

Executive Summary

This report presents Deliverable 4.1 "MDE Toolset and Architecture Specification" of the SHAPE project, including the results of the tasks T4.1 as defined in the Description of Work.

This document covers the SHAPE's toolset and architecture specification. It addresses the conceptual architecture of the tool suite describing the components and integration means.

The architecture of the SHAPE Integrated Toolsuite is driven by:

- [1] SHAPE Document of Work;
- [2] D1.1 – Case study scenario descriptions and requirements specifications;
- [3] D2.1 – Model-driven methodology and architecture specification
- [4] D3.1 – Consolidation of common basis for metamodels and languages for SHA.

The transformation related architecture topics are covered by [5] D5.1 – Model Transformation and Deployment Architecture Description.

Table of Contents

EXECUTIVE SUMMARY	3
TABLE OF CONTENTS.....	4
LIST OF FIGURES.....	5
1 INTRODUCTION	6
1.1 SCOPE.....	6
1.2 INTENDED AUDIENCE	6
1.3 DOCUMENT STRUCTURE	6
2 MAIN DRIVERS FOR ARCHITECTURE	7
2.1 CASE STUDY DRIVERS.....	7
2.2 DRIVERS BY STANDARDS.....	7
2.3 METHODOLOGY DRIVERS	8
3 CONCEPTUAL ARCHITECTURE	9
3.1 ARCHITECTURE OVERVIEW	9
3.2 MDE TOOL COMPONENTS BY CONSORTIUM.....	10
3.2.1 Objecteering CASE Tool by SOFTEAM	10
3.2.2 SoaML / ShaML Editor by ESI.....	12
3.2.3 MOFScript by SINTEF	13
3.2.4 Web-Service Modelling Toolkit by STI.....	13
3.2.5 PIM4Agents by DFKI	15
3.2.6 CIMFlex by DFKI/IWI	16
3.2.7 Composition Studio by SINTEF	17
3.3 EXTERNAL COMPONENTS AVAILABLE TO CONSORTIUM.....	18
3.3.1 ARIS	18
3.3.2 OAW.....	19
3.3.3 ATL.....	20
3.3.4 EPF	20
3.3.5 KBR by MOMOCS	20
3.4 MDE TOOL CHAINS	21
3.4.1 StatoilHydro	22
3.4.2 Saarstahl	23
3.5 FUNCTIONALITIES TO BE DEVELOPED.....	24
4 TOOL SUITE INTEGRATION APPROACH.....	25
4.1 AVAILABLE TOOL INTEGRATION SOLUTIONS	26
4.1.1 XML.....	26
4.1.2 Eclipse Plugins.....	26
4.1.3 Eclipse OSGi	27
4.1.4 ModelBus by ModelWare/ModelPlex.....	27
4.2 RECOMMENDATIONS FOR INTEGRATION	27
4.3 INTEGRATION ARCHITECTURE	28
4.4 ADDRESSING MAIN DRIVERS	29
5 ANNEXES.....	30
5.1 ACRONYMS	30
5.2 REFERENCES	31
5.3 SHAPE TOOL SUITE REQUIREMENTS FROM [2]	31
5.3.1 Case Study Requirements by StatoilHydro.....	31
5.3.2 Case Study Requirements by Saarstahl	32

List of Figures

Figure 1: SHAPE Models and Transformations	9
Figure 2: Component Architecture Overview	10
Figure 3: The transformation process in CIMFlex	17
Figure 4: Tool Chain Example for StatoilHydro Case Studies	22
Figure 5: Tool Chain Example for Saarstahl Case Studies	23
Figure 6: Integration Architecture	28

1 Introduction

1.1 Scope

The document intends to analyse the main drivers for the MDE tool components and identify the missing functionalities. The SHAPE consortium owns many relevant MDE tool components. The document does not pretend to describe the architectures of these components, but rather gives their overview and provides recommendations of their integration into the SHAPE Toolsuite.

1.2 Intended Audience

Although the document is mainly intended for the Work Package 4 participants, it can be useful for other work packages as a reference for the SHAPE Toolsuite integration architecture.

1.3 Document Structure

The document is structured as follows:

- Section 1 – introduces the document;
- Section 2 – provides a reminder of the main drivers for the SHAPE tool suite development;
- Section 3 – gives an overview of the architecture component;
- Section 4 – discusses the integration approaches and proposes recommendations;
- Section 5 – presents some additional information in annexes.

2 Main Drivers for Architecture

2.1 Case Study Drivers

This section describes the main functional requirements for the SHAPE's tool suite mentioned in [2].

Several case studies mention necessity of the Business Modelling. The Business modelling also includes using semantic technologies for information models and workflows. The complex multi criteria scheduling problems require the multi-agent system (MAS) architectures to be modelled and deployed. In addition some cases involve adaptable orchestrations of Web Services.

The following table summarises the case study drivers.

Main Drivers	Understanding
MAS	MAS architecture is required for solving multi criteria scheduling problems
Semantics	Information models are to be semantically annotated
Orchestrations	The adaptable orchestrations are required.
Legacy Systems	Modelling from business perspective is necessary. Integration using middleware technologies should be supported
Business Processes	Support in adaptation and orchestration

Table 1 Case Study Drivers

All these aspects should be considered while developing the SHAPE Architecture.

2.2 Drivers by Standards

This section describes the requirements driven by standards to be supported.

The Document of Work mentions the following standard languages to be supported by the SHAPE tool suite:

BPMN – Business Process Modelling Notation¹ by OMG

BMM – Business Motivation Metamodel² by OMG

BPDM – Business Process Definition Metamodel³ by OMG

EPC – Event Driven Process Chain by SAP implemented in ARIS

SOAML – SOA Modelling Language⁴ by OMG with contributions by the SHAPE consortium

UML – Unified Modelling Language⁵ by OMG

OWL – Web Ontology Language⁶ by W3C

BPEL – Business Process Execution Language⁷ by W3C and OASIS

WSDL – Web Service Definition Language⁸ by W3C

¹ <http://www.bpmn.org/>

² http://www.omg.org/technology/documents/bms_spec_catalog.htm

³ <http://www.omg.org/cgi-bin/doc?bei/03-01-06>

⁴ <http://www.omg.org/docs/ad/08-08-04.pdf>

⁵ <http://www.uml.org/>

⁶ <http://www.w3.org/TR/owl-features/>

⁷ <http://www.oasis-open.org/committees/wsbpel/>

XSD – XML Schema Definition⁹ by W3C

The definitions of the use cases [2] also imposes the following standard languages to be supported

WITSML – Wellsite Information Transfer Standard Markup Language¹⁰ by Energistics

PRODML – Production Markup Language¹¹ by Energistics

2.3 Methodology Drivers

This section describes the main requirements that are driven by the methodology [3].

The methodology supposes that several types of models and different versions will be elaborated during the lifecycle of system definition and development. The workflow of the lifecycle will involve different participants with different concerns. In this way, the SHAPE MDE process will require a collaborative environment. This environment should support the management of the models: their versioning, change log and access rights.

In addition integration with an automated human-enactment / project workflow engines may be advised for supporting resource planning, scheduling and participants notification.

⁸ <http://www.w3.org/TR/wsd/>

⁹ <http://www.w3.org/XML/Schema>

¹⁰ <http://www.witsml.org>

¹¹ <http://www.prodml.org/>

3 Conceptual Architecture

In this section we are presenting the concepts behind the SHAPE tool suite architecture.

3.1 Architecture Overview

According to the Document of Work [1] the following figure shows the MDA layers, standards involved and the place of the ShaML – the main scope of the SHAPE project.

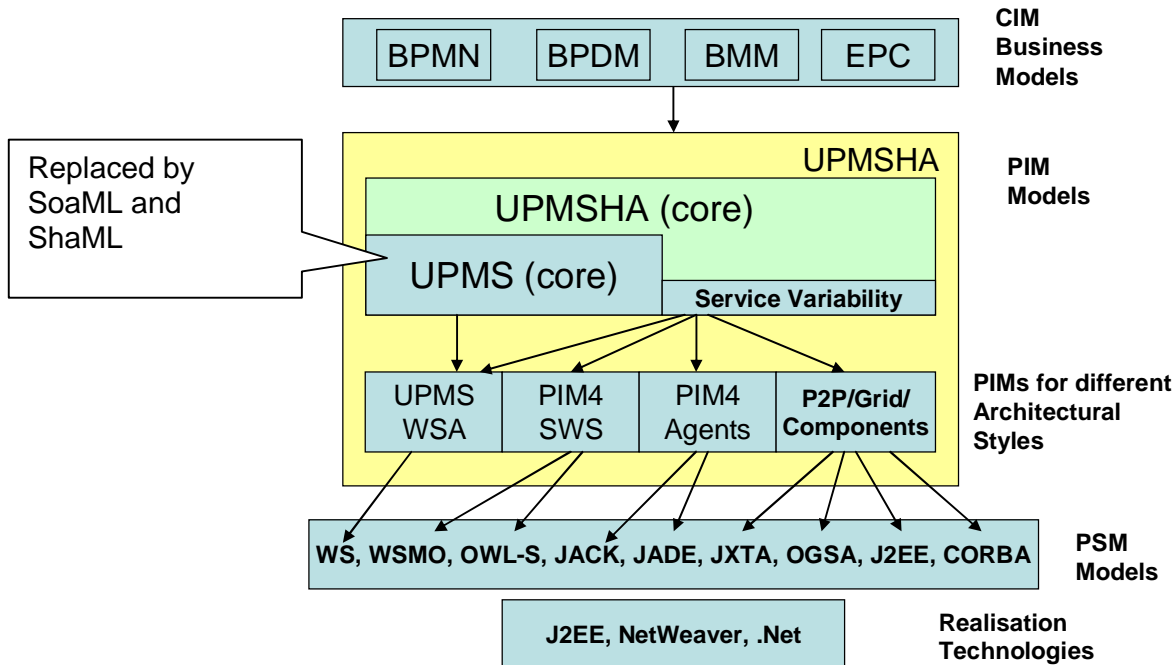


Figure 1: SHAPE Models and Transformations

The tool components included to the SHAPE Tool Suite should provide functionalities in all indicated domains and also provide bridges between modelling approaches. These bridges should be implemented by the model transformation technologies. The utmost goal is to elaborate a tool suite that would cover all MDA layers: CIM Business Models, PIM logical architecture, PSM for several required platforms and code generation for the most of the realisation technologies.

The MDE tool components identified can be categorised in the way depicted on Figure 2.

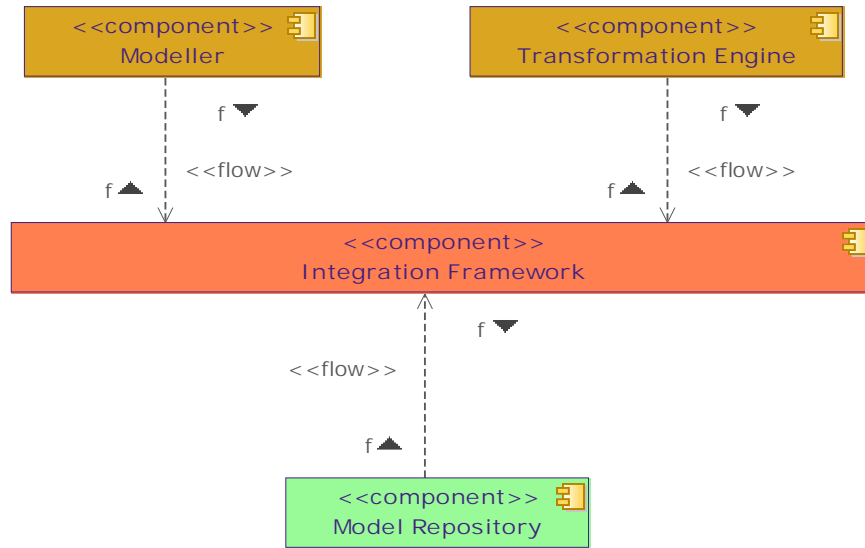


Figure 2: Component Architecture Overview

Modeller is an abstract model editor component, which regroups model editors for all metamodels that must be supported (BMM, BPMN, UML, ShaML and etc.).

Transformation Engine is a group of tools dedicated to converting the models. The transformation related topics are addressed in details in [5].

Model Repository is a service component required for model sharing in a collaborative team work.

Integration Framework is the tool set integration environment.

In the next section we will discuss the available candidate components that will constitute the SHAPE Tool Suite.

3.2 MDE Tool Components by Consortium

This section provides an overview of the Modeller and Transformation Engine components available in the consortium.

3.2.1 Objecteering CASE Tool by SOFTEAM

Tool Name	Objecteering Case Tool www.objecteering.com
Tool Version	6.1
Company Name	SOFTEAM
Tool Type	Modeller / Transformation Tool / Verification Tool
Tool Description	Objecteering 6 provides complete coverage of model driven development. Incorporating the latest UML 2.0 modeling and MDA developments for model driven engineering, Objecteering 6 manages model consistency and traceability throughout the entire development cycle, from requirements, analysis and design through code generation, tests and application deployment. Complete coverage of model-driven development, from requirements,

	<p>analysis and design through code and tests</p> <p>Seamless transition from one phase to the next, with full traceability between requirements, model elements and code</p> <p>Dynamic model completeness and consistency checking</p> <p>Powerful MDA automation, resulting in improved model transformation and code generation</p> <p>Integrated repository shared throughout all development phases</p> <p>Automated documentation generation and web publishing</p> <p>Integration and synchronization with the Eclipse development environment</p> <p>Full teamwork management capacity with model merging facilities and CM tools integration</p> <p>Integration with third party tools such as Configuration Management, Requirements Management, Application Servers and IDEs</p> <p>UML2 Profiles</p>
Tool Keywords	UML2, BPMN, Enterprise Architecture, UML2 Profiles, Round Trip Engineering: C++, Java, C#, Traceability, Requirements Management
Expected Usage	SOFTEAM will provide an industrial implementation of SOAML as an UML2 Profile in the Objecteering CASE Tool. SOFTEAM will also provide extension and support in order to cover the expected CIM/PIM/PSM layers within the scope of SHAPE.
Tool User Benefits	Industrial Product, User Support
Extension/Integration mechanisms available	Eclipse plugin, open Java API, Export/Import EMF UML2. Possibility of integration of custom export/import features, by preference EMF Ecore
Tool Licensing	<p>Pay License: 2 free licenses per partner for the whole duration of SHAPE. Additional free licenses should be discussed separately. Please contact andrey.sadovykh@softeam.fr</p> <p>Limited version - Objecteering Fee Edition - is available.</p>

Layer	Available Functionalities	Intended Improvements
CIM Business Models:	<p>Enterprise Architecture Module:</p> <p>BPMN business process modelling</p> <p>Companies Organization modeling</p> <p>Company's IT infrastructure modelling</p> <p>Scope Manager Module:</p> <p>Dictionary</p> <p>Mission</p> <p>Goals and Requirements</p>	<p>Coupling with SoaML profile.</p> <p>Transformation approaches.</p> <p>Traceability approaches.</p>

	(BMM) Business Rules	
PIM Models:	Company's IT infrastructure modelling UML2	SoaML Profile
PSM Models:	Custom UML2 Profiles for: CORBA, J2EE, Java, C++, .Net (C#)	Web Services modelling
Transformation Technologies	Plain Java, "J" (Objecteering script language)	Coupling with an open source transformation engine
Realisation Technologies:		J2EE with Apache Axis and BPEL

3.2.2 SoaML / ShaML Editor by ESI

Tool Name	SoaML editor
Tool Version	1.0
Company Name	ESI, SINTEF
Tool Type	Modeller
Tool Description	The ESI will contribute to the project to obtain a SoaML editor. The roadmap for the editor will have two parallel lines. One will have as objective the support of the metamodel edition. In a first stage this will make use of emf (treeview) facilities, in a second stage it will make use of gmf, and in the third stage it will integrate both views emf(treeview) and gmf. The second line will work in the customisation of a state of the practices uml modelling tool with a UML profile for the SoaML metamodel.
Tool Keywords	UML2, EMF, GMF, UML Profile
Expected Usage	ESI will provide an experimental implementation of SoaML as metamodel and as UML2 Profile in the eclipse environment and in a state of the practice UML tool (preferability eclipse based and open source).
Tool User Benefits	SoaML metamodel editor. SoaML java library will facilitate the development of transformations
Extension/Integration mechanisms available	Eclipse plugin, SoaML Java API, Export/Import EMF UML2, Export/Import EMF SoaML.
Tool Licensing	Open Source

Layer	Available Functionalities	Intended Improvements
-------	---------------------------	-----------------------

PIM Models:	N/A	SoaML metamodel tree-view editor, SoaML Profile coupling
Transformation Technologies	N/A	Transformations from/to SoaML Will be developed in the WP5

3.2.3 MOFScript by SINTEF

Tool Name	MOFScript
Tool Version	1.3
Company Name	SINTEF
Tool Type	Model-to-Text Transformation Tool
Tool Description	The MOFScript subproject aims at developing tools and frameworks for supporting model to text transformations. http://www.eclipse.org/gmt/mofscript/
Tool Keywords	Model transformation, model to text
Expected Usage	Developing model transformation rules
Tool User Benefits	Support generation of implementation code or documentation from models. It should provide a metamodel-agnostic framework that allows usage of any kind of metamodel and its instances for text generation.
Extension/Integration mechanisms available	Eclipse plug-in
Tool Licensing	Eclipse Public License (EPL)

3.2.4 Web-Service Modelling Toolkit by STI

Tool Name	The Web Service Modeling Toolkit
Tool Version	1.4.3
Company Name	Semantic Technology Institute (STI) Innsbruck
Tool Type	Modeller / Transformation Tool / Verification Tool
Tool Description	<p>The Web Service Modeling Toolkit (WSMT) is an Integrated Development Environment for Semantic Web services that supports an engineer through the full life cycle of their semantic descriptions of Web services in WSML, from creation, through validation and testing, to deployment on a Semantic Execution Environment. The WSMT is implemented as a collection of plug-ins for the Eclipse framework and currently focuses on three main areas of functionality:</p> <p>Creation and Management of WSML Artefacts: The Web Service Modeling Ontology (WSMO) is made up of four top level elements, namely Ontologies, Web services, Mediators and Goals. The WSMT provides functionality for editing, browsing, validating and testing these elements through the Web Service Modeling Language (WSML) – a formalization of the WSMO conceptual model.</p> <p>Creation and Management of Mediation Mappings: One of the key</p>

	<p>challenges in semantics is the interoperability of ontologies. In the Semantic Web Service field this becomes even more important when the service requester and service provider use different ontologies to describe the same domain. The WSMT provides the Mapping perspective within which mediation mappings between two or more ontologies can be created at design time, such that they can later be executed at runtime.</p> <p>Interfacing with Semantic Execution Environments: Crucially once all the artifacts related to a Semantic Web Service have been created these artifacts need to be deployed to the execution environment within which they will be used. The SEE Perspective provides functionality for interfacing with Semantic Execution Environments like the Web Service Execution Environment (WSMX) and IRSIII.</p>
Tool Keywords	Semantic Web Services, Ontologies, Semantics
Expected Usage	<p>In the SHAPE project the tools in the WSMT will be the main entry point for those wishing to use SoaML with semantics and with Semantic Web Services. The following developments will be made in the course of the project to enable this integration:</p> <p>Integration of Model Transformation from WP5 to enable SoaML models to be translated to WSML descriptions</p> <p>Extension of existing tools to enable models generated from SoaML to be further specified</p> <p>Extension of Model Transformation functionality to enable additions made by the engineer to be preserved when models are re-transformed from SOAML to WSMO.</p> <p>Implementation of automated testing facilities to enable automated unit testing of WSML Ontologies, Web Services and Goals as they are generated from the SOAML models.</p> <p>Implementation of methodology-support tools for Semantic Web Services that support the methodologies from work package 2 and guide the engineer in the creation of their Semantic Web Service descriptions from requirements gathering through to deployment.</p>
Tool User Benefits	Provides support for creating semantic description for Web services. It offers editing capabilities for users with different levels of expertise, from beginners (graph based visualization and editing) to expert users (direct access to WSML descriptions).
Extension/Integration mechanisms available	Eclipse plugin, open Java API, custom import/export functionalities (OWL, RDFS)
Tool Licensing	Open Source (Dual licensed under GPL and LGPL – GPL being phased out where possible)

Layer	Available Functionalities	Intended Improvements
PIMs for different Architectural Styles:	N/A	SOAML Profile
PSM Models:	Editors and visualizers for WSMO Ontologies, Web Services, Goals and Mediators	<p>Coupling with SOAML profile: Transformation approaches for import/export from/to SOAML.</p> <p>Validation tools for ensuring</p>

		WSMO models retransformed from SoaML are still logically sound.
Transformation Technologies	WSMO to OWL, RDFS and OCML	SoaML to WSMO
Realisation Technologies:	Web Services, OWL-S	N/A

3.2.5 PIM4Agents by DFKI

Tool Name	PIM4Agents
Tool Version	1.0
Company Name	DFKI
Tool Type	Modeller / Transformation Tool
Tool Description	<p>PIM4Agents provides complete coverage of model driven development of agent-based systems.</p> <p>Complete coverage of model-driven agent-based development, from design to code.</p> <p>Seamless transition from service-oriented architectures through models transformation form PIM4SOA to PIM4Agents.</p> <p>Dynamic model completeness and consistency checking through formal semantics.</p> <p>Powerful MDD automation, resulting in improved model transformation and code generation for two agent-based platforms (i.e. JACK and JADE).</p> <p>Integration and synchronization with the Eclipse development environment.</p>
Tool Keywords	MAS, EMF, JACK, JADE, GMF
Expected Usage	DFKI will provide a model transformation from SoaML to DSML4MAS
Tool User Benefits	Open Source Product, User Support
Extension/Integration mechanisms available	Eclipse plugin, Export/Import EMF Ecore.
Tool Licensing	Open source, contact: Christian.Hahn@dfki.de

Layer	Available Functionalities	Intended Improvements
PIMs for different Architectural Styles:	PIM4Agents metamodel	Integration of Semantic Web Services. Goal modelling.
PSM Models:	JACK, JADE	N/A
Transformation Technologies	ATL transformation from PIM4Agents to JACK, JADE	Transformation from SoaML to PIM4Agents.
Realisation Technologies:	JACK, JADE	N/A

3.2.6 CIMFlex by DFKI/IWI

Tool Name	CIMFlex
Tool Version	1.0
Company Name	DFKI/IWI
Tool Type	Modeller
Tool Description	The CIMFlex acts on the computation level. It is an open replacement for EPC modelling from ARIS. It also allows organisational modelling, business process modelling with BPMN
Tool Keywords	CIM, EPC, BPMN, Organisation, Rules
Expected Usage	DFKI will provide a model transformation from SoaML to DSML4MAS
Tool User Benefits	Open Source Product, User Support
Extension/Integration mechanisms available	Eclipse plugin, Export/Import EMF Ecore.
Tool Licensing	Proprietary. Free for consortium. contact: Dima.Panfilenko@iwi.dfki.de

Layer	Available Functionalities	Intended Improvements
CIM	N/A	CIM Flex development: EPC support, BPMN Support, Organization models, Rules Support
Transformation Technologies	N/A	Transformation from CIMFlex to SoaML

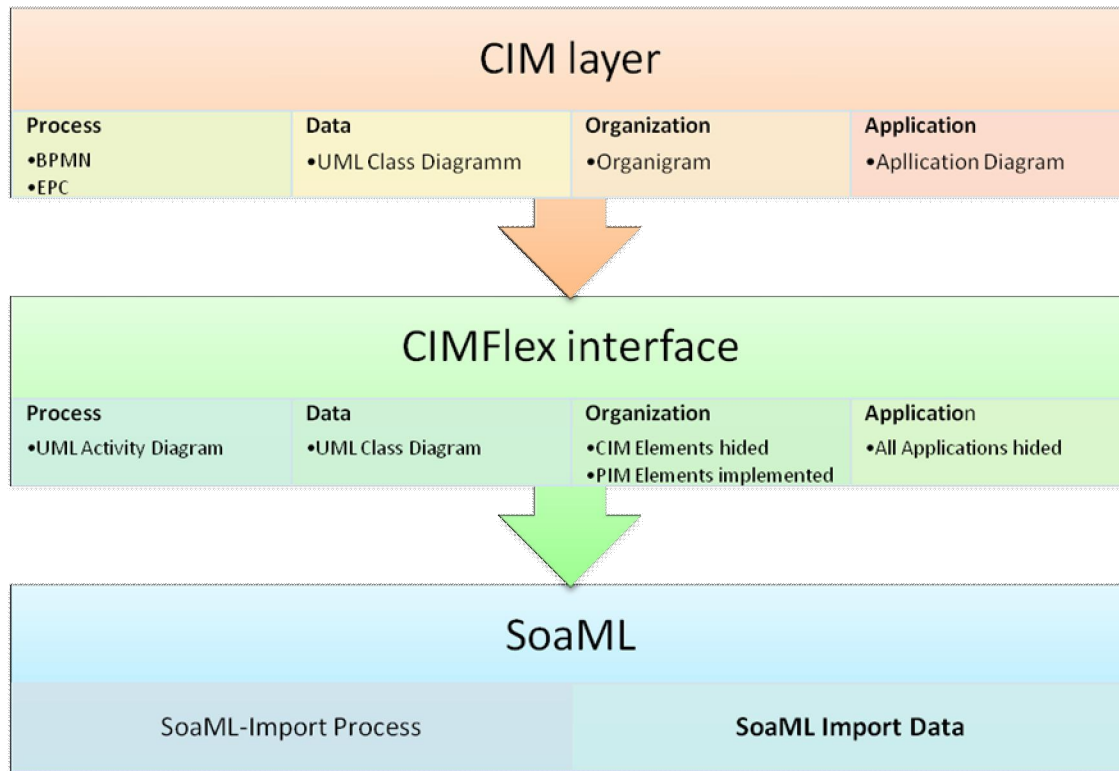


Figure 3: The transformation process in CIMFlex

3.2.7 Composition Studio by SINTEF

Tool Name	Composition Studio
Tool Version	1.0
Company Name	SINTEF
Tool Type	Modeller
Tool Description	
Tool Keywords	Web Services
Expected Usage	The Composition Studio will provide special facilities for covering the Web Service composition aspects.
Tool User Benefits	Open Source Product, User Support
Extension/Integration mechanisms available	Eclipse plugin, Export/Import EMF Ecore.
Tool Licensing	Open source. Contact: Simon.Han@sintef.no

Layer	Available Functionalities	Intended Improvements
PIM	Web Service composition modelling.	Coupling with SoaML
Transformation Technologies	MOFScript is used for	Transformation from

	transformation into WSML	Composition Studio from/to SoaML will be developed
--	--------------------------	--

3.3 External Components Available to Consortium

3.3.1 ARIS

Tool Name	ARIS
Tool Version	7.0.0.2
Company Name	IDS Scheer
Tool Type	Integrated Modeling Tool for Business Models and Services
Tool Description	<p>The ARIS Toolsuite is a integrated development environment. ARIS (Architecture of Integrated Information Systems) is also a method for analyzing processes and taking a holistic view of process design, management, work flow, and application processing.</p> <p>As one of the Enterprise Modeling methods, "Architecture of Integrated Information Systems" (ARIS) provides four different aspects of applications:</p> <p>The ARIS concept is the architecture for describing business processes.</p> <p>The ARIS concept provides modeling methods, the meta structures of which are comprised in information models.</p> <p>The ARIS concept is the foundation for the ARIS Toolset software system for the support of modeling.</p> <p>The ARIS house of Business Engineering (HOBE) represents a concept for comprehensive computer-aided Business Process Management.</p>
Tool Keywords	EPC
Expected Usage	<p>Possible usage for the standards:</p> <p>EPC</p> <p>BPMN</p> <p>BPEL</p> <p>UML</p> <p>WSDL</p> <p>and for the CIM, PIM and PSM-Layers.</p>
Tool User Benefits	<p>Integrated view of all perspectives:</p> <p>organisation</p> <p>function</p> <p>data</p> <p>product</p>

	<p>process</p> <p>rules</p> <p>roles</p> <p>filters</p> <p>modeltransformation</p> <p>managing hierarchies between models</p>
Extension/Integration mechanisms available	<p>ARIS Strategy Platform</p> <p>ARIS Implementation Platform</p> <p>ARIS Controlling Platform</p> <p>EMF exchange-format</p> <p>Link: http://www.ids-scheer.co.uk/en/Software/70484.html</p>
Tool Licensing	Commercial. Evaluation License is to be confirmed.

3.3.2 OAW

Tool Name	openArchitectureWare
Tool Version	4.2
Company Name	openArchitectureWare.org
Tool Type	MDA/MDD generator framework
Tool Description	<p>openArchitectureWare (oAW) is a modular MDA/MDD generator framework. It supports parsing of arbitrary models, and a language family to check and transform models as well as generate code based on them. Supporting editors are based on the Eclipse platform. OAW has strong support for EMF (Eclipse Modelling Framework) based models but can work with other models, too (e.g. UML2, XML or simple JavaBeans) At the core there is a workflow engine allowing the definition of generator/transformation workflows. A number of prebuilt workflow components can be used for reading and instantiating models, checking them for constraint violations, transforming them into other models and then finally, for generating code.</p> <p>http://www.openarchitectureware.org</p>
Tool Keywords	model transformation, modeling frontends
Expected Usage	<p>Model-to-text transformation</p> <p>Model-to-model transformation</p> <p>Text-to-model transformation</p>
Tool User Benefits	<p>Modeling Frontends</p> <p>Full EMF Integration</p> <p>Configurable transformation process</p> <p>All current UML tools supported</p> <p>GMF adapter</p> <p>Eclipse Integration (syntax checking, code assist, debugger, ...)</p>

Extension/Integration mechanisms available	Eclipse plug-in
Tool Licensing	Eclipse Public License (EPL)

3.3.3 ATL

Tool Name	ATLAS (ATL, AMMA, AMW, AM3)
Company Name	INRIA
Tool Type	Transformation Engine
Tool Description	A set of MDD components: ATL (ATLAS Transformation Language) AM3 (ATLAS MegaModel Management) AMW (ATLAS Model Weaver) AMMA Toolkit (ATLAS Model Management Architecture) http://www.eclipse.org/gmt/
Tool Keywords	EMF, Transformation, M2M, M2T
Expected Usage	Model-to-Model and Mode-to-Text transformations
Tool User Benefits	Documentation, Examples, Libraries of Transformations
Extension/Integration mechanisms available	Eclipse plug-in
Tool Licensing	Eclipse Public License (EPL)

3.3.4 EPF

Tool Name	Eclipse Process Framework
Tool Type	SPEM Modeller, Project Planner
Tool Description	Eclipse Process Framework proposes a Software Process Engineering Metamodel (SPEM) enabled modelling tool.
Tool Keywords	SPEM, Eclipse, EPF
Expected Usage	Modeling of the SHAPE methodology. Process Enactment, Automation.
Extension/Integration mechanisms available	Eclipse plug-in
Tool Licensing	Eclipse Public License (EPL)

3.3.5 KBR by MOMOCS¹²

Tool Name	Knowledge Base Repository (KBR)
-----------	---------------------------------

¹² <http://www.momocs.org/>

Tool Type	Model Repository
Tool Description	Semantically enabled model repository. It allows storing various types of artefacts, attaching transformation traceability information, semantically annotation and search.
Tool Keywords	KBR
Expected Usage	Distributed sharing of modelling artefacts.
Extension/Integration mechanisms available	Eclipse plug-in
Tool Licensing	Eclipse Public License (EPL)

3.4 MDE Tool Chains

In this section a potential tool chains for the SHAPE case studies are presented. These tool chains are specified taking into account the information provided in [2].

3.4.1 StatoilHydro

Figure 4 presents a potential tool chain for StatoilHydro case study. CIM models developed with Objecteering are transformed into ShaML models. This ShaML models are enriched in Objecteering in order to add necessary details for meaningful transformation into a WSML model. Then the WSML model is enriched with a dedicated editors from WSMT and deployed into Web Service Execution Environment (WSMX).

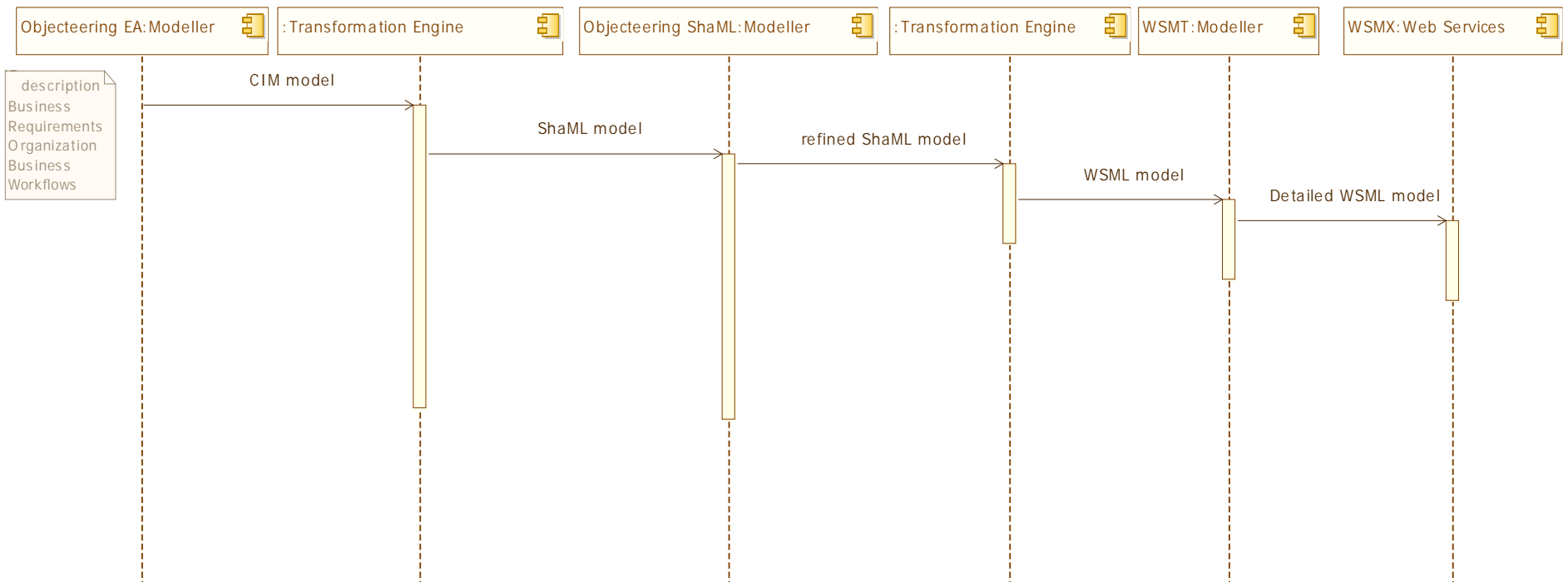


Figure 4: Tool Chain Example for StatoilHydro Case Studies

3.4.2 Saarstahl

In the Saarstahl use case (Figure 5), business models are transformed to PSM models that are finally executed. Therefore, the entire tool chain developed within the SHAPE project is applied. Business models designed using the CIMFlex tool suite are transformed to ShaML, refined using the Objecteering editor, and transformed to more platform-specific platforms like PIM4Agents or Web services. Tools like ATL and MOFScript are used for the purpose of transforming the related models.

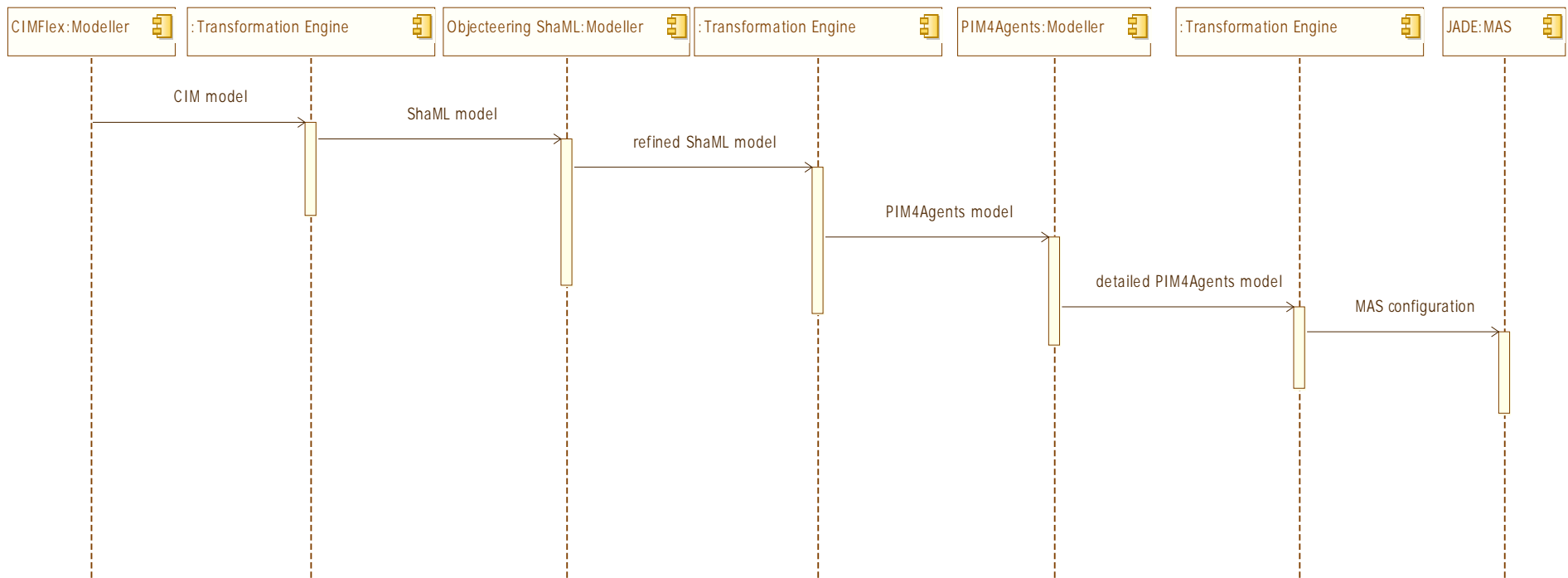


Figure 5: Tool Chain Example for Saarstahl Case Studies

3.5 Functionalities to be Developed

The functionalities to be developed can be divided into two categories: modelling functionalities and dedicated transformations.

As it was outlined the available components cover most of the required modelling standards. The missing items are the SoaML and ShaML. In addition the integration of the different modelling languages from different layers (CIM, PIM, PSM, code) should be developed using transformation techniques covered in details in [5].

Finally, the most critical part will be the application of the all set of languages to the case studies, analysis of the results and adaptation of the approaches.

Table 2 summarise the available functional units and the development tasks.

Layer	Available Components	To be Developed
CIM	Objecteering EA and Scope, ARIS	CIMFlex tool, Transformation to ShaML
PIM	Objecteering EA and UML, DSML4MAS PIM4Agents	SoaML and ShaML UML2 Profile in Objecteering and other modellers. SoaML and ShaML Metamodel Editor, Transformations to PIM4Agents and WSML, PSMs and Realisations languages
PSM	Objecteering C++, Java, CORBA and J2EE. Editors and visualizers for WSMO Ontologies, Web Services, Goals and Mediators	Support of Web Services and BPEL as UML2 Profiles in Objecteering. Transformations to Realisations languages
Realisation	JACK, JADE, Web Services	Objecteering Web Services

Table 2 Development List

Considering the transformations required it is identified that the business models on the CIM level should influence the PIM level ShaML models. In addition the transformation from/to PIM4Agents and WSML are needed to integrate these modelling approaches into the workbench. Finally some direct ShaML to Code transformation may be necessary for JACK, JADE and Web Services platforms.

4 Tool Suite Integration Approach

The main goal of the tool suite integration is to enable heterogeneous MDE tools to interoperate in order to fulfil the activities defined by the SHAPE's methodology.

The integration of the software systems should be considered from different view points as suggested in the literature:

- Data integration – harmonisation of the information flows. The systems should talk the same language, which includes definitions of the same semantics, information constructs and serialization formats.
- Control integration – the means to access functionalities should be agreed and standardized. This includes definition of shared interfaces and transport protocols.
- User Interface – to ensure the common perception of the integrated system by users, it is also required to integrate user interfaces. The means to achieve this include: GUI development conventions, look&feel conventions, reporting conventions to be shared between the software systems.

While all aspects are important for building the truly integrated tool suite, the third aspect is particularly difficult to respect in the SHAPE context. Indeed, the tools to be integrated have their own GUI policies and limited options for GUI refining. The project has limited resources in manpower and time, while delivering working prototypes is essential for the use-case validation and thus the project success. Therefore, the levels of the integration should be defined as follows in order of priority and feasibility:

- Conceptual or semantical – the tools adopt the same semantical model meaning that the tools share the same notions. For example, the term “service” used in different tools has the same standardized meaning. The tools declare their compatibility to the ShaML metamodel to the determined extent.
- Data exchange – the tools define the data format for exchanging the information. In the SHAPE's case, the tools may interoperate using the file exchange in the XMI format. The manual execution of the modelling services is necessary. The users export and import the data from/to the tools and run MDE functionalities.
- Modelling services – the tools implement standard interfaces for invoking modelling functionalities. The consumer tools use this interfaces for particular modelling services. For example, the model repository can implement a service for storing, versioning, queering retrieving of the model artefacts. The MDE tools may use this service for making available the models to the team in a central storage.
- Automation – the execution of the chains of modelling service invocations in order to fulfil a methodological activity. The tools can use some predefined orchestrations of the modelling services. For example, the code generation activity may require execution of PSM model querying, model retrieving, and verification, calling transformation engine for model-to-text processing, delivering the code to the particular developers, informing the developers about the code available and proceeding to the next methodological phase.
- User Interface – from our opinion, the most difficult level is harmonisation of the tools look&feel.

Depending the priorities and available resources in the tool development teams, the tool suite will be iteratively integrated to achieve the most possible upper level.

4.1 Available Tool Integration Solutions

Developing a tool integration solution is not in the scope of the SHAPE project. Therefore exiting COTS should be carefully chosen and adopted. In this section we present a set of technologies the most relevant in our opinion.

4.1.1 XMI

XML Metadata Interchange (XMI)¹³ is the specification by OMG for model serialization, which the most relevant for data integration in the SHAPE's context. Although the specification imposes a standard, there exist different versions (2.1.1 is the latest) and implementation flavours depending on the modeller tools. Till recently the tools supposed to provide XMI export/import could not interoperate due to the differences in the XMI format. Currently Eclipse Modelling Framework (EMF¹⁴) XMI is the most supported flavour. Although the versioning problems remain, the migration from one version to another is less complicated since the EMF Java API is mostly backward compatible.

The above mentioned considerations are relevant for exchanging metamodels and their instance models. For the UML2 Profile exchange the situation is even more complicated. Tools implement different versions of UML (2.1.2 is the latest) and in different manners since the UML metamodel is often ambiguous in details. Eclipse provides an EMF based UML2 metamodel¹⁵. Currently it is the most supported metamodel (IBM RSM, Topcased, Papyrus, Sparx Enterprise Architect, Objecteering by SOFTEAM) though versioning problems remain but are manageable (2.2.0 is the latest version).

Concerning the profile support, although EMF supports profiles, only EMF-native modellers (Topcased, Papyrus) support exchange of profiles. Other tools (e.g. Enterprise Architect, Objecteering, Magic Draw) implement their own extensions mechanisms that are not interchangeable. OMG MDA Tool Component¹⁶ initiative lead by SOFTEAM intends to provide the common basis for tool components interchange, but the work is far to be finished and widely adopted. In this way, the XMI export/import of profiled UML models in most of the commercial modellers is not operational, since the profile references may not be read properly by other modellers.

Anyway, it is highly recommended to choose either the metamodel-based, or the UML2 profile-based approach, fix the versions of the EMF and UML2 metamodel (in case of profiles). As it was proven by other EU projects the limited interoperability is still largely possible and achievable.

4.1.2 Eclipse Plugins

With regards to the control and interface integration, Eclipse proposes a great plugin framework. Plugins can share the resources and access each other's functionalities at runtime via JVM. In this way the Eclipse based MDE tools can share functionalities. In addition, Eclipse imposes a common look&feel through a well-defined GUI concepts: workspace, perspectives, properties, preferences, consoles and etc.

However, several inconvenience issues should be noted:

- The tools should be Java-based or have a Java-based integration layer. For example Microsoft or other Windows based tools such as Enterprise Architect would be difficult to integrate with Eclipse based tools.
- The resource sharing can be achieved only locally and in the same Java runtime. The modelling services cannot be accessed remotely. In particular, this means that network communication layer with a central model repository needs to be implemented for sharing the models, while middleware principles allow it to be seamless.

Nevertheless, Eclipse plugins are very convenient and should be seriously considered for integration of the SHAPE tool suite.

¹³ <http://www.omg.org/technology/documents/formal/xmi.htm>

¹⁴ <http://www.eclipse.org/modeling/emf/>

¹⁵ <http://www.eclipse.org/modeling/mdt/?project=uml2>

¹⁶ <http://doc.omg.org/ad/2006-6-9> and <http://www.omg.org/docs/ad/07-06-12.pdf>

4.1.3 Eclipse OSGi

Open Service Gateway initiative (OSGi¹⁷) is a Java based middleware for integration of the components. Eclipse natively proposes an OSGi implementation called Equinox¹⁸. It allows Eclipse the bundle integration, which in particular permits remote service invocation.

Basically the Equinox is used to web-enable Eclipse Rich Client Platform (RCPs) and can be compared with J2EE middleware.

The Equinox has the following issues important in the SHAPE context:

- The tools should be Java-based or have a Java based integration layer. For example Microsoft tools and Enterprise Architect would be difficult to integrate with Eclipse based tools over Equinox.
- Equinox doesn't have specific facilities for marshalling EMF models. Thus packaging/unpackaging should be developed for each tool.
- Since it relies on the Java serialization, the in/out parameters transport schema is not possible.
- Equinox doesn't have comprehensive bundle registry describing MDE services.
- There are no orchestration engines for Equinox, and thus the automation is only possible by Java coding.

4.1.4 ModelBus by ModelWare/ModelPlex

LIP6, FHG FOKUS and SOFTEAM prototyped ModelBus¹⁹ in the frame of the ModelWare/ModelPlex²⁰ projects. ModelBus is an open source web-service middleware for integration of modelling services. It allows remote service invocation and provides special services for "model" parameters. The services are comprehensively described including the possibility for "model" parameters to be typed by "metamodels". The Registry allows locating the necessary modelling service. Both Java and dotNet tools can be integrated by means of dedicated Adapters. Automation of service execution can be implemented using an Orchestration Tool. Many ModelWare/ModelPlex tools have been already integrated with ModelBus including:

- Modellers – RSA, Objecteering, Topcased, Enterprise Architect;
- Transformation Engines – MOFScrip, ATL;
- Model Repositories – Adaptive, FHG FOKUS;
- Verification Tools – FHG FOKUS OCL checker.

Although the results are promising, the following important issues should also be mentioned:

- ModelBus is a research project and rather limited support is available yet due to limited resources;
- The backwards compatibility is still rather limited meaning that tools claimed integrated are probably incompatible since using different versions of the ModelBus runtime.

4.2 Recommendations for Integration

After the analysis of the available means and basing on our experience, the following can be drawn.

¹⁷ <http://www.osgi.org/>

¹⁸ <http://www.eclipse.org/equinox/>

¹⁹ <http://www.modelbus.org/>

²⁰ <http://www.modelplex-ist.org/>

The integration using Eclipse plugins framework is the most convenient and time-efficient approach. Although promising and flexible, the middleware technologies need a complex infrastructure to be managed and maintained. In the context of the research project unnecessary complication should be avoided. Therefore, the integration of non-Java tools or the tools to be remotely accessed should be analysed case by case.

As it is mentioned before, it is highly recommended to choose either the metamodel-based, or the UML2 profile-based approach, fix the versions of the EMF and UML2 metamodel (in case of profiles).

4.3 Integration Architecture

Figure 6 depicts the most recent view of the integration architecture of the SHAPE Toolsuite.

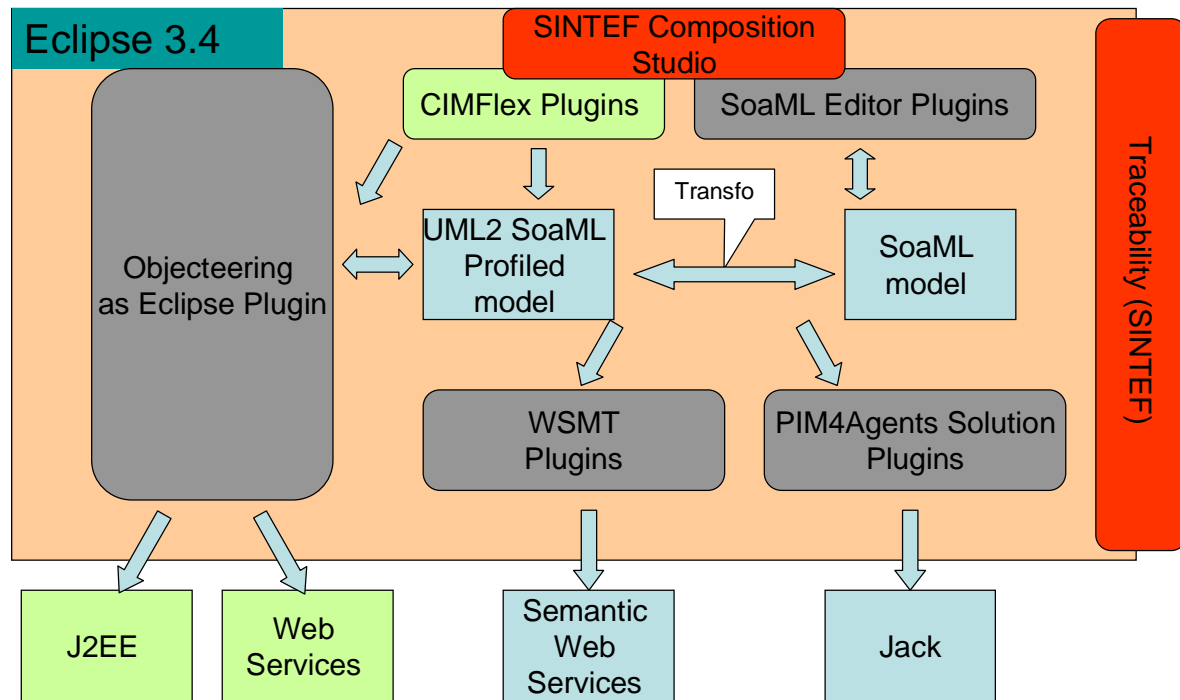


Figure 6: Integration Architecture

It was decided that Eclipse 3.4 will host the SHAPE Toolsuite. Platform included EMF, Eclipse UML2Tools, GEF and GMF as well. The tools are deployed as Eclipse equinox plugins. The exchange between all tools is fulfilled through the SoaML profiled UML2 models. Although, all tool providers decided to integrate with the SHAPE Toolsuit by means of the SoaML UML2 Profile, the SoaML metamodel and its instances are still present in the architecture for more flexibility in future. It is planned to develop an automatic transformation from SoaML profiled UML2 models to SoaML metamodel instances and vice versa.

The version 1 of the SHAPE Toolsuit will contain:

- Objecteering CASE Tool covering CIM and PIM modelling including creation of SoaML models;
- CIMFlex for CIM modelling. It is planned that CIMFlex will export CIM models to Objecteering ;
- PIM4Agents Solution for modelling MAS Architectures. The SoaML profiled UML2models will be transformed in PIM4Agent models;
- WSMT for modelling semantic Web Services. The SoaML profiled UML2models will be transformed in WSML models.

WSML models produced by WSMT may be deployed into a Semantic Web Services platform, while PIM4Agents models may serve for creation of MAS systems with Jack and Jade. The J2EE and plain Web Services platforms are implicitly covered by Objecteering.

For the future versions SINTEF Composition Studio will be integrated for better covering the Web Service Composition aspects. Moreover a new Traceability Tool by SINTEF will help to ensure trace logging and consistency for all involved models on the all MDA abstraction levels. In addition to SoaML the future SHAPE Toolsuit will also address the extension – ShaML, which is being developed in the frame of the SHAPE project.

4.4 Addressing Main Drivers

In this section we summarise the means that addresses the main drivers for SHAPE Toolsuite development.

Main Drivers	Means for addressing
MAS	PIM4Agents Solution covers the need for modelling the MAS systems. The ShaML extension will help in integrating SOA and MAS architectures.
Semantics	Ontologies are addressed by WSMT. The SOA architectures will be enriched with semantically information using ShaML. The transformation to WSML will help in automated creation of Semantic Web Services.
Orchestrations	Objecteering BPMN-to-BPEL transformation covers the need for supporting Web Service orchestrations.
Legacy Systems	The middleware technologies used - Web Services and MAS – are dedicated to deal with legacy issue for SOA.
Business Processes	Both EPC and BPMN are supported by SHAPE Toolsuite with CIMFlex and Objecteering correspondingly.
Standards	All standards mentioned in section 2.2 are addressed by the SHAPE Toolsuite components.
Collaborative work	Later versions of the SHAPE Toolsuite will integrate Traceability Tool and Model Repository in order to cover need for support of the collaborative work expressed in the SHAPE Methodology (section 2.3).

5 Annexes

5.1 Acronyms

Acronym	Description
ATL	ATAS Transformation Language
BMM	Business Motivation Metamodel by OMG
BPDM	Business Process Definition Metamodel by OMG
BPEL	Business Process Execution Language by W3C and OASIS
BPMN	Business Process Modelling Notation by OMG
DSL	Domain Specific Language
EMF	Eclipse Modelling Framework
EPC	Event Driven Process Chain by SAP implemented in ARIS
GMF	Graphical Modeling Framework by Eclipse
MAS	Multi Agent Systems
MDA	Model Driven Architecture, trademark by OMG
MDD	Model Driven Development
MDE	Model Driven Engineering
OWL	Web Ontology Language by W3C
ShaML	(formerly UPMSHA) UML Profile for Modelling for Semantically-enabled service Architectures by the SHAPE consortium
SoaML	(formerly UPMS) SOA Modelling Language by OMG with contributions by the SHAPE consortium
UML	Unified Modelling Language by OMG
UPMS	(currently SoaML) UML Profile and Metamodel for Services
UPMSHA	(currently ShaML) UML Profile and Metamodel for Semantically-enabled Heterogeneous Service Architecture
WSDL	Web Service Definition Language by W3C
WSML	Web Service Modeling Language
WSMO	Web Service Modeling Ontology
WSMT	Web Service Modeling Toolkit
WSMX	Web Service Execution Environment
XSD	XML Schema Definition by W3C

Table 3 – Acronyms

5.2 References

- [1] SHAPE Document of Work
- [2] D1.1 – Case study scenario descriptions and requirements specifications
- [3] D2.1 – Model-driven methodology and architecture specification
- [4] D3.1 – Consolidation of common basis for metamodels and languages for SHA
- [5] D5.1 – Model Transformation and Deployment Architecture Description

5.3 SHAPE Tool Suite Requirements from [2]

5.3.1 Case Study Requirements by StatoilHydro

Context	Requirements	StatoilHydro Specific Requirements
Modeling	Direct editor support for the SoaML metamodel and the SoaML UML profile	Editor support allows to refine the generated design based on the CIM description
	BPMN should be supported by the Tool Suite	BPMN can be used to describe the workflow between the different software components as a starting point for process optimization
	Support for adapting orchestration without changing the underlying services	Designing adaptive service is one major precondition when it comes to designing adaptive orchestrations
	Support for modeling the problem domain components as agents	Agent technology is applied to the problem of scheduling product cargos
	Support for identifying and defining the algorithms for scheduling optimization	StatoilHydro requires the possibility to model their utilized algorithms for production planning
	Model management system	Required for managing different versions of models and metamodels
	Collaboration environment for the different modelers and users	StatoilHydro has offices all over Norway, and the rest of the world. Collaboration environment support that offices can easily interact and collaborate
	Tool Suite should help to address extension of existing standards with richer semantic content (WITSML/PRODML)	Semantic content needs to be provided to create adaptive workflows
Transformation	Transformation from PIM to PSM should be supported	Necessary to transfer business models to SOA
	Transformations from PIM to PSM should be supported	Necessary to transfer SOAs to multiagent systems
	Transformation should support executable code generation	Necessary for concrete planning and assignment of ships to jetties
	Transformations have to consider already existing legacy system	Interfaces to the existing SPORT RAF cargo database need to be provided

Table 4 Case Study Requirements by StatoilHydro

5.3.2 Case Study Requirements by Saarstahl

Context	Requirements	Saarstahl Specific Requirements
Modeling	Direct editor support for the UPMS metamodel and the SOA-Pro UML profile	Necessary in the case when business models need to be manually refined after applying CIM to PIM transformation
	BPMN should be supported by the Tool Suite	Necessary for modeling business requirements and business processes
	Support for adapting orchestration without changing the underlying services	Existing legacy systems need to be integrated as services
	Support for modeling the problem domain components as agents	Saarstahl already has experience with using agent technology for productions planning and wants to integrate it
	Support for identifying and defining the algorithms for scheduling optimization	Saarstahl requires the possibility to model their utilized algorithms for production planning
	Model management system	Required for managing different versions of models and metamodels
Transformation	Transformation from PIM to PSM should be supported	Necessary to transfer business models to SOA
	Transformations from CIM to PIM should be supported	Necessary to transfer SOAs to multiagent systems
	Transformation should support executable code generation	Necessary for concrete capacity planning
	Transformations have to consider already existing legacy system	Interfaces for the connection to existing databases, DispoXT, etc. are required

Table 5 Case Study Requirements by Saarstahl