Deliverable D42.3
Factsheet #1
"Generic Service Development Platform"

Document Owner:	Singular Logic
Contributors:	
Dissemination:	Public
Contributing to:	WP4.2
Date:	29/3/2013
Revision:	1.0

List of Abbreviations

IDE	Integrated Development Environment
JDK	Java Development Kit
SVN	Subversion (versioning control system)
BSM	Business-Specific Model
TIM	Technology-Independent Model
TSM	Technology-Specific Model
ACL	Access Control List
SAWSDL	Semantic Annotations for WSDL
WSDL	Web Services Description Language
SOAP	Simple Object Access Protocol
M2T	Model To Text

Table of Contents

1 Availability and Contacts

Version	1.0
Availability	Source is available from the MSEE svn repository
Accompanying specification and design document	Deliverable D42.3: Generic Service Development Platform First Prototype
Source control	svn://repo.nimbus-ware.com/MSEE/SP4/WP42/D42.3/trunk/gsdev-platform/
Contact person	Jerry Dimitriou – ep6@singularlogic.eu

2 Architecture and Functionalities

2.1 Introduction

The MSEE Generic Service Development Platform is an integrated development environment, allowing IT users to develop software applications and services for MSEE. The chosen use case for the first release of the platform involves:

- importing Technology Independent Models (UML Class Diagrams & BPMN 2.0 models)
- selection of the desired target technologies and components
- model transformation and/or enhancement to produce a Technology-Specific Model which can then be used to compose the target application or service

Based on this composition, the Generic Service Development platform will automatically generate editable source code, which will be then mapped into an executable application / service. This application/service will be then registered into the Service Delivery Platform, making it available for use in the appropriate MSEE-supported ecosystems through the Innovation Ecosystem Platform of SP2.

2.2 Functionalities & use cases

The first prototype of the Generic Service Development Platform is able to support the following use cases:

- Obtain TIM models (UML class diagram, BPMN 2.0 diagram) from Model Repository
- Enrich TIM models as required for adding TSM-level metadata and save them to Model Repository
- Build a service's domain model by transforming TSM-level UML class diagram to Java sources
- Enrich a TSM-level WSDL with required annotations in SAWSDL and register a deployed web service with the Service Delivery Platform
- Locate a SOAP service registered with the Delivery Platform and prepare Java client code to invoke the service
- Enrich a TIM-level BPMN model with required information to invoke a SOAP service already registered in the Service Delivery Platform in an automated task

2.3 Architecture

The Generic Service Development Platform is an IDE based on Eclipse Juno SR2 platform, including a number of additional plugins, developed, customized and configured as a whole to facilitate the use cases described previously. The Generic Service Development Platform is a rich GUI application that is executed by each technical user/service developer and has runtime integration with Model Repository Server and Service Delivery Platform.

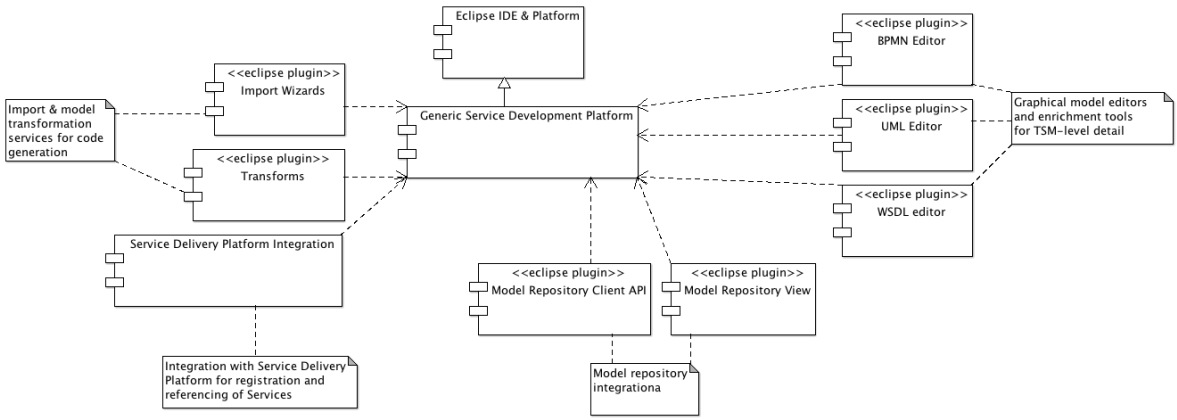

Figure 1 Generic Service Development Components

Each component of the Generic Service Development Platform is further described in section 3 Availability and ContactsTechnical Information.

3 Technical Information

3.1 Generic Service Development Platform

The Generic Service Development Platform is built on the Eclipse Juno SR2 platform (version 4.2.2). Its functionalities are implemented as extensions and plugins of the Eclipse IDE, taking advantage of existing Eclipse functionality whenever available, such as Java projects support, as well as extending Eclipse with model-driven development support tools, to facilitate and implement development tasks in the service lifecycle. In the following sections we provide technical information about the components introduced in Figure 1. Moreover, all technical information regarding supported operating systems, Eclipse platform and development environment, apply to all components of the Generic Service Development Platform.

3.1.1 Operating systems

The Generic Service Development Platform is available on all operating systems, on which Eclipse is available: Windows, Mac OS X, Linux. It has been tested on Mac OS X and Windows.

3.1.2 Eclipse platform

The Generic Service Development platform and associated plugins are built on Eclipse Juno SR2 (v4.2.2) environment.

3.1.3 Development environment

All plugins have been developed using Eclipse Juno SR2 and Plugin Development Environment. The source code is distributed as Eclipse projects.

3.2 Model Repository Client API and View Plugins

The Model Repository Client API and View plugins have been extensively described in a separate factsheet ([2]).

3.3 Import Wizards

Wizards have been created to facilitate import of elements from the Model Repository and the Generic Service Delivery Platform. Imported elements may include TIM models (UML Class Diagram and BPMN 2.0 diagrams) or semantically annotated web service definitions registered in the Delivery Platform.

3.4 Graphical Editors

The Generic Service Development Platform integrates and enhances graphical editors for enhancing TIM models with TSM-level details. Integrated editors include UML class diagram editor, BPMN 2.0 process editor as well as WSDL editor.

3.5 Service Delivery Platform integration

The Service Delivery Platform integration component is a headless plugin which provides service registration and discovery functionalities when required by other plugins (e.g. Import Wizards).

3.6 Transformations

The Transformations component is responsible for transforming TSM-enriched models to source code to be customized packaged and deployed using standard Java development tools. The first prototype implementation incorporates transformations:

- From UML class diagram to Java Beans sources
- From WSDL to Java web service client for invocation

The Transformations component takes advantage of Eclipse M2T (Model-to-Text) tools.

3.7 Technical details

Nature	Eclipse plug-ins
Programming Language	Java
Development Tools	Eclipse Juno SR2
Additional libraries	Papyrus UML plugin Activiti BPMN plugin

4 Licensing

4.1 License

To be defined.

4.2 Third party licenses

Third party software	License
Papyrus UML plugin	Eclipse Public License v1.0
Activiti BPMN plugin	Apache License v2.0
Eclipse Platform	Eclipse Public License v1.0

5 Technical Manual

5.1 Installation

The Generic Service Development Platform is distributed as a complete binary Eclipse distribution with everything required to execute the IDE, given that a Java runtime environment is already installed on the workstation.

5.2 Retrieving sources from the SVN

Source code for the Generic Service Development Platform is available at MSEE project SVN repository:
svn://repo.nimbus-ware.com/MSEE/SP4/WP42/D42.3/trunk/gsdev-platform/

6 User Manual

The Generic Service Development extends Eclipse functionality via plugins, building on Eclipse IDE existing support for standard Java projects and enhancing it with additional capabilities to support the development of services based on TIM models. In this section, we provide a walkthrough of some of the key features integrated in the Generic Service Development Platform.

Importing TIM models to a Java project is a matter of invoking the Model Import wizard. This wizard operates in 3 steps, depicted in Figure 2, Figure 3 and **Error! Reference source not found.**Figure 4:

- Setup connection to model repository
- Browse repository and select models for import
- Select destination folder where models will be imported.

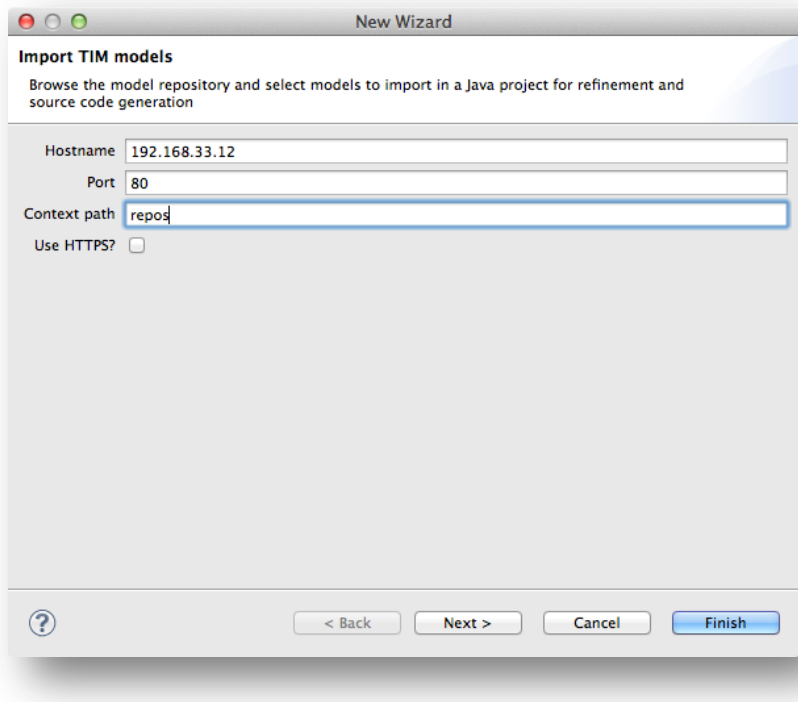

Figure 2 Import models wizard - step 1 of 3, connection to model repository

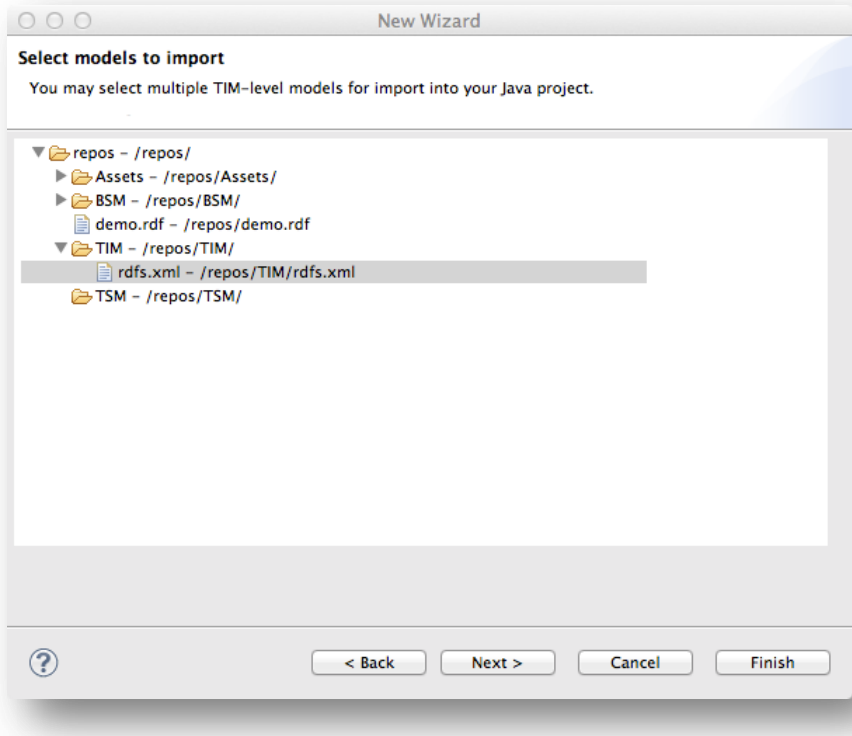

Figure 3 Import models wizard, step 2 of 3 - select models for import

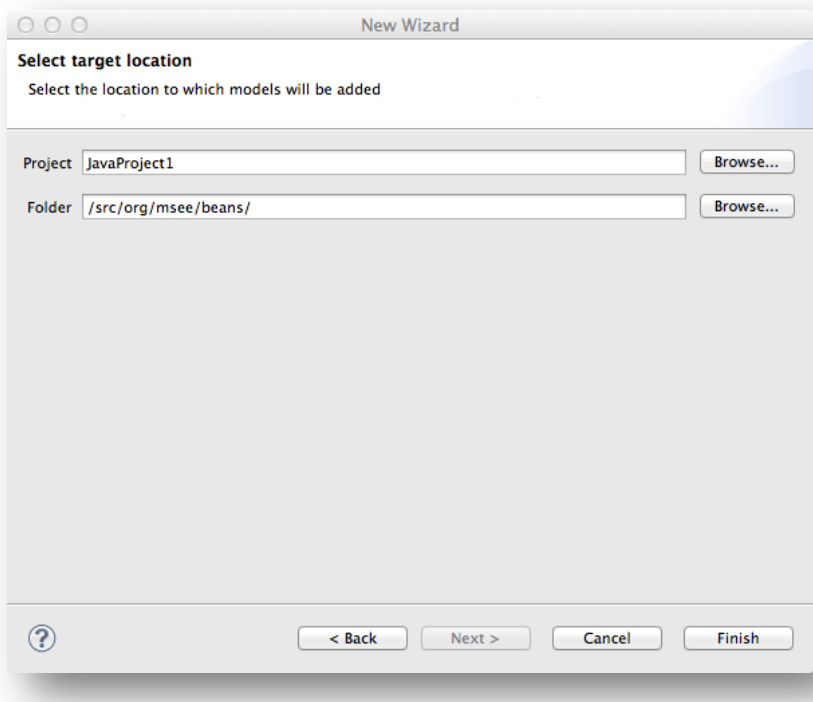

Figure 4 Import models wizard - step 3 of 3, select import destination

Model repository connection configuration can be also saved to a project's properties, so the initial repository connection wizard page (Figure 2) is already filled-in for the user.

In order to generate Java beans sources from an XMI-serialized UML class diagram, Generic Service Development Platform provides the Java beans source generation wizard, which is presented in the following screenshots. The action to start source code generation is available

as an action in an XMI file's context menu as well as a "New Java beans source code generation from UML Class Diagram" wizard via New → Other… option. In the former case, the first page of the wizard is already filled-in with the details of the currently selected XMI file.

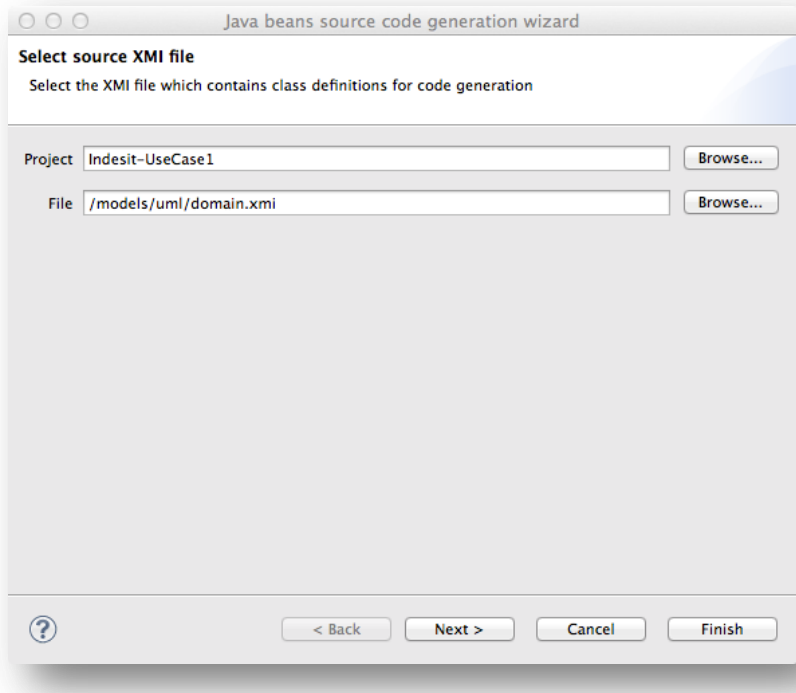

Figure 5 Java beans source code generation wizard: Select source XMI file

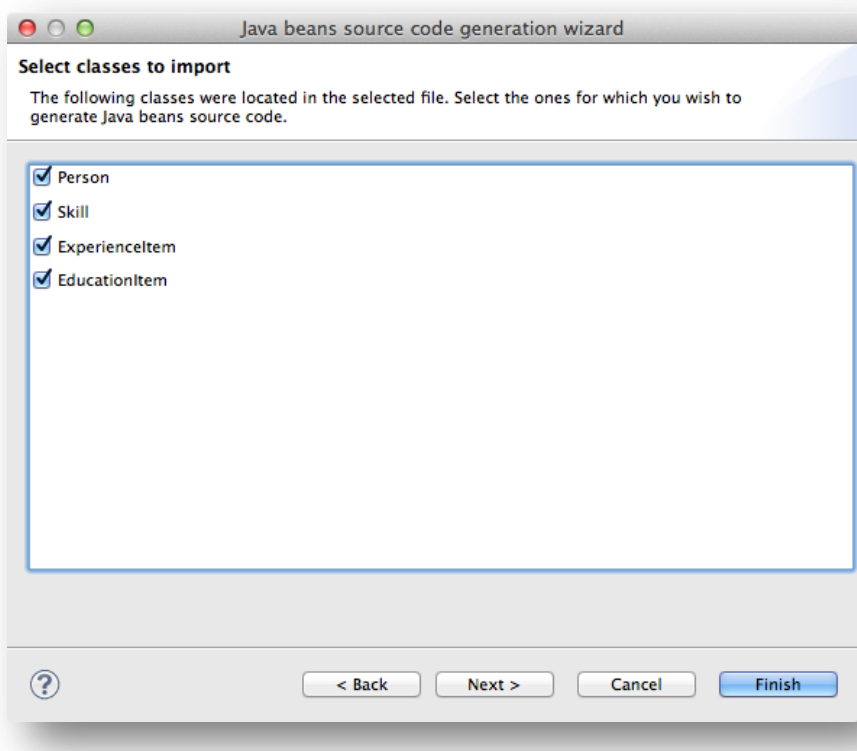

Figure 6 Java beans source code generation wizard: Select classes for which source code will be generated

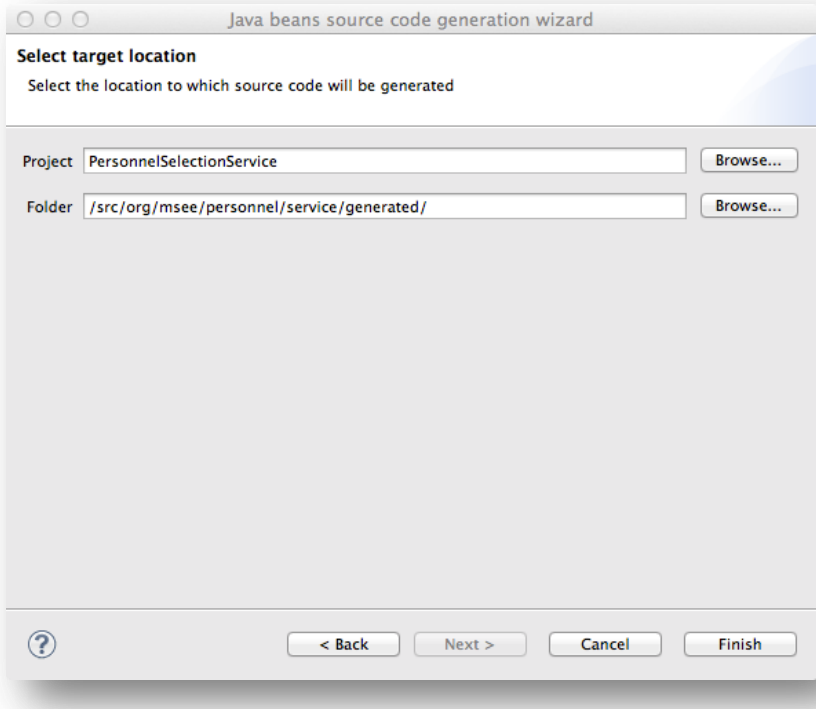

Figure 7 Java beans source code generation: Select destination for generated source code

Once the wizard finishes, the generated source files are located at the target location and reflect the structure obtained via XMI.

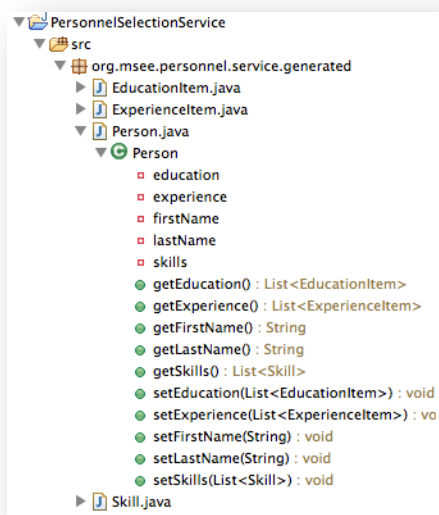

Figure 8 Generated Java sources

Similarly to the process shown above to import UML class diagrams, the Generic Service Development Platform is also capable of importing BPMN diagrams from the Model Repository, in order enhance them with the required details so that they become TSM models and can be packaged and deployed on Innovation Ecosystem Platform for execution.

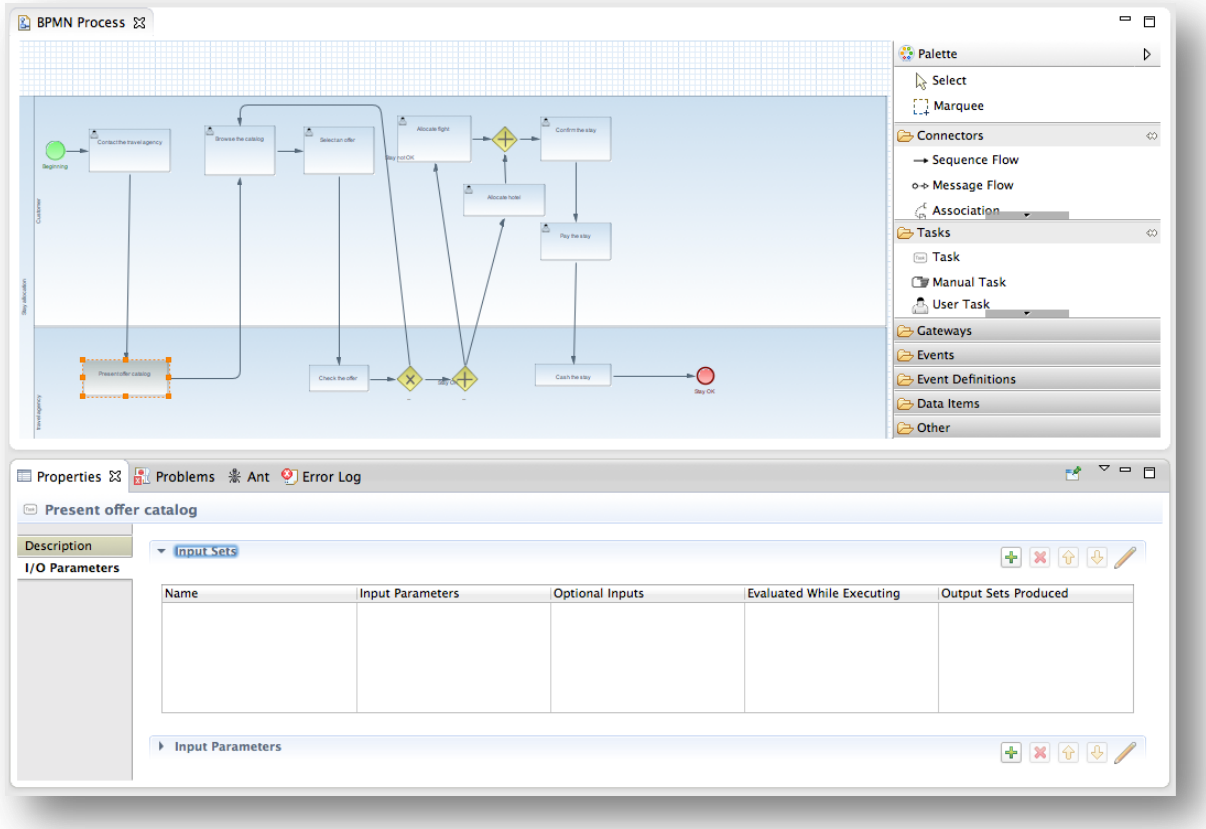

Figure 9 Integrated BPMN process editor

The Generic Service Development Platform is able to enrich "Automated Task" steps in BPMN processes, in order to bind them with invocation of service registered with the Delivery Platform. The service developer is able to use Service Delivery Platform's Discovery capabilities to locate a service by category or by namespace and method name and bind it to an automated task process step.

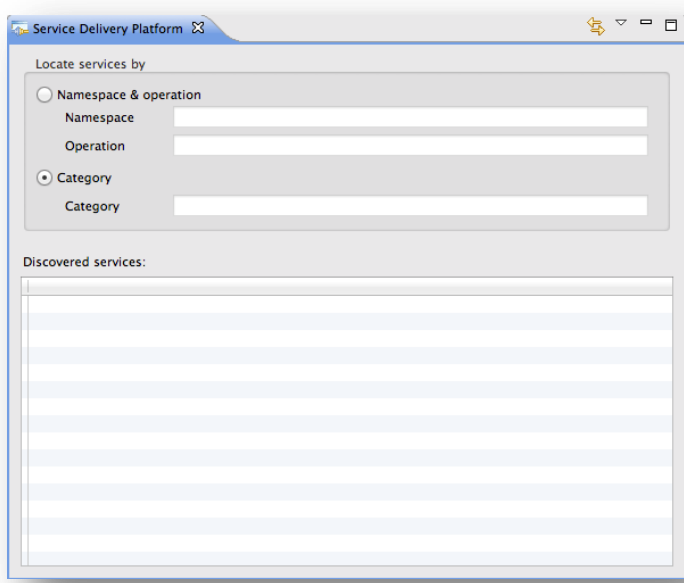

Figure 10 Locating services registered in the Service Delivery Platform

Finally, the "New MSEE Web Service Client" wizard, also uses the Service Delivery Platform discovery dialog shown in Figure 10 to identify a service and then generate web service client source code that will be used to invoke the service from other applications or services.

7 Future plans

The first prototype of the Generic Service Development Platform will be integrated and used in the framework of the MSEE IT System ([3]). Future development will focus on integration, usability and expanded functionality, according to the results of integration testing as specified in [4].

8 References

[1] MSEE Deliverable D42.1 Generic Service Development Platform specifications and architecture.

[2] MSEE Factsheet D42.3 Generic Service Development Platform first prototype – Factsheet #2

[3] MSEE Deliverable D45.1 MSEE Service-System Integrated

[4] MSEE Deliverable D45.3 Test Plan and Documentation M18 issue