Deliverable D42.3
Factsheet #2
"Model Repository"

Document Owner:	Singular Logic
Contributors:	
Dissemination:	Public
Contributing to:	WP4.2
Date:	29/3/2013
Revision:	1.2

List of Abbreviations

IDE	Integrated Development Environment
JDK	Java Development Kit
SVN	Subversion (versioning control system)
WebDAV	Web Distributed Authoring & Versioning, an extension of HTTP protocol
BSM	Business-Specific Model
TIM	Technology-Independent Model
TSM	Technology-Specific Model
ACL	Access Control List

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/model-repository
Contact person	Jerry Dimitriou – ep6@singularlogic.eu

2 Architecture and Functionalities

2.1 Introduction

The Model Repository serves as a central repository for collaboratively creating and editing models of any type. Therefore it is used by business analysts and software developers alike, across the board of MSEE tools available for model creation, editing and consumption:

- SLM Toolbox, which provides modelling tools for BSM and TIM models
- Generic Service Development Platform, which loads TIM models from the model repository and enriches them with details required to become TSM models
- Mobile Development Platform, also loads TIM models from the model repository to generate code for REST services and clients development

2.2 Requirements

The Model Repository was designed to facilitate the following requirements, as analysed in [1]:

- A common repository to facilitate end-to-end process management, from business to technical level of analysis
- Enhance collaboration of actors across the service lifecycle, supporting multi-user access schemes (e.g. lock/release models)
- Search capability
- Access control of resources available on the repository
- Optionally, provide versioning support

2.3 Architecture

The Model Repository is composed of the following components:

- Model Repository Server, the centralized "single point of truth" repository.
- Model Repository Client API module, which is delivered as an Eclipse plugin and provides an API via which other Eclipse plugins can communicate with the Model Repository Server.
- The Model Repository View, which is an Eclipse plugin with a UI, providing the MSEE Model Repository View which allows end users to perform all interaction with the Model Repository Server.

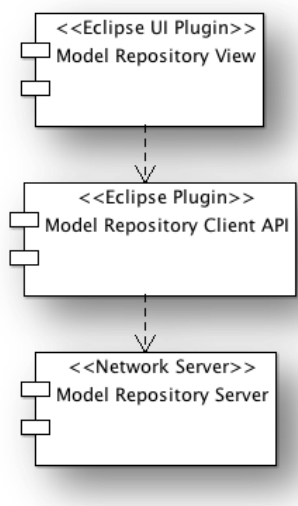

Figure 1 Model Repository Components

The Model Repository Server is deployed as a network server process to a central server, accessed by Workstations which execute MSEE Eclipse-based tools, using the Model Repository Eclipse plugins:

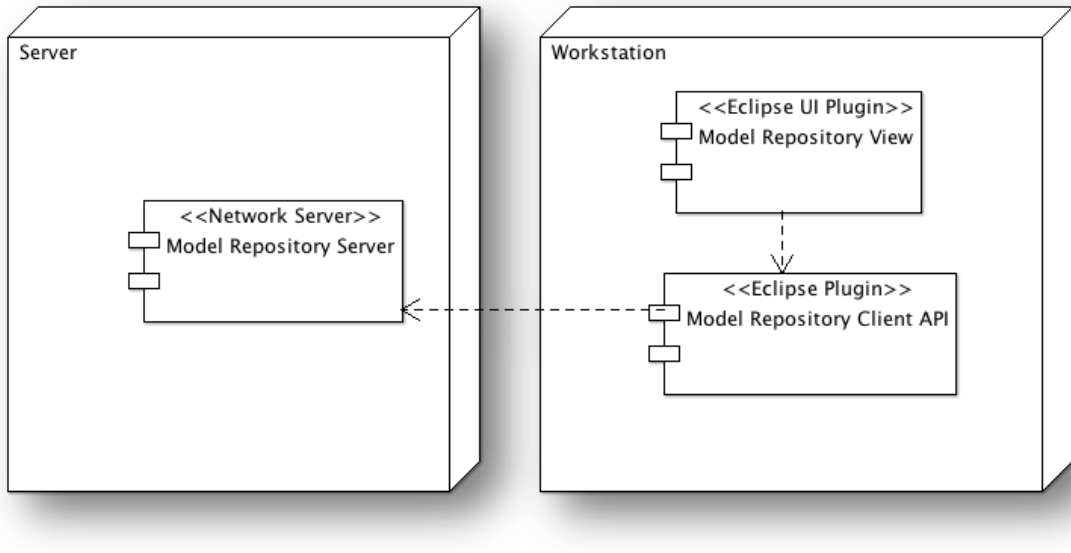

Figure 2 Model Repository Components Deployment

The Model Repository Server has been designed in the original Generic Service Development Platform specification ([1]) as a CDO model repository server ([2]). However, due to a number of advantages, the first prototype implementation is based on a central WebDAV server ([3]). Being an open web standard based on HTTP, there are several implementations, both open source and commercial, of the WebDAV protocol which meet and exceed the requirements for the Model Repository. The advantages of WebDAV approach include:

- Storage of any kind of resource is possible in WebDAV, while in CDO only ECore models are supported. Moreover, WebDAV allows the hierarchical organization of resources in collections, similarly to a filesystem organized with files and folders, allowing the categorization of models per level (BSM, TSM, …) or per function within the organization (HR, IT, Procurement etc).
- Support for distributed authoring is built into WebDAV with a multifaceted locking mechanism to support collaborative editing while maintaining information integrity.
- WebDAV supports annotation of each resource with certain properties, which are typical of file systems such as who is the creator, date/time created or last modified etc. The properties mechanism is extensible to support custom properties as well, allowing the annotation of models as required for proper operation of the model editors.
- Search is integrated in WebDAV protocol. Combined with the possibility to setup custom metadata, it is possible to perform semantically enhanced searches.
- WebDAV comes with a standard security model based on ACLs which will be customized to use MSEE security model with the Federated SSO Utility Service ([4]).

3 Technical Information

3.1 Model Repository Server

The Model Repository Server is a standard web server with WebDAV extensions. For example, such a server could be Apache HTTPD with WebDAV modules or Apache JackRabbit ([6]) content repository with WebDAV extensions enabled. For the first prototype, all testing was performed against Apache HTTPD, therefore it is recommended for now to use Apache HTTPD.
A Vagrant ([5]) virtual machine configuration is provided in MSEE SVN repository which automatically sets up an Ubuntu Linux virtual machine with Apache HTTPD and WebDAV configuration, for quick and easy setup of a development environment.

3.2 Model Repository Client API and View Plugins

The Model Repository Client API and View plugins are both Eclipse plugins. The Model Repository Client API can be used as a library by any Eclipse plugin which requires connectivity to the Model Repository server from within its own UI, while the Model Repitory View plugin supplies a separate Eclipse View which offers standalone browsing and management of resources in the Model Repository Server.

3.2.1 Operating systems

No specific constraint. (Both modules have been successfully installed and tested on Windows 7 Enterprise Edition and Mac OS X).

3.2.2 Java development kit

This module requires a Java SE 6 environment.

3.2.3 Eclipse platform

Both plugins require Eclipse Juno SR2 (v4.2.2) environment.

3.2.4 Development environment

Both plugins have been developed using Eclipse Juno SR2 and Plugin Development Environment.

3.3 Technical details

Nature	Source code
Programming Language	Java
Development Tools	Eclipse Juno SR2
Additional libraries	Sardine WebDAV Client Apache Commons Lang Apache Commons HTTP Client
Server	Apache HTTPD

4 Licensing

4.1 Service license

To be defined.

4.2 Third party licenses

Third party software	License
Sardine WebDAV client	Apache License version 2.0
Apache Commons HTTP Client Apache Commons Lang	Apache License version 2.0

5 Technical Manual

5.1 Installation

Installation of the Model Repository Server is a standard Apache HTTPD Server setup process, for example see the Ubuntu Server Guide for setup instructions on how to setup Apache Server in Ubuntu Linux: https://help.ubuntu.com/12.04/serverguide/httpd.html. For a quick development setup, it is possible to use the Vagrant configuration provided in MSEE SVN which creates a virtual machine with an Apache HTTPD server and WebDAV extensions with a single command.

Workstation installation of plugins is simply a matter of adding each plugin's JAR file in your current ECLIPSE_HOME/plugins/ folder. It is easy to verify the plugins were started successfully by inspecting the Plugins view after Eclipse starts (Window → Show View → Plug-ins) and locating `org.msee.modelrepos.client` and `org.msee.modelrepos.browser` plugins are loaded.

5.2 Retrieving sources from the SVN

Source code for the Model Repository modules is available at MSEE project SVN repository: svn://repo.nimbus-ware.com/MSEE/SP4/WP42/D42.3/trunk/model-repository

6 User Manual

In order to browse a Model Repository, user needs to display the Model Repository View in their Eclipse environment. To locate this view, navigate Window → Show View → Other.. then expand "MSEE Model Repository" category and select "Model Repository".

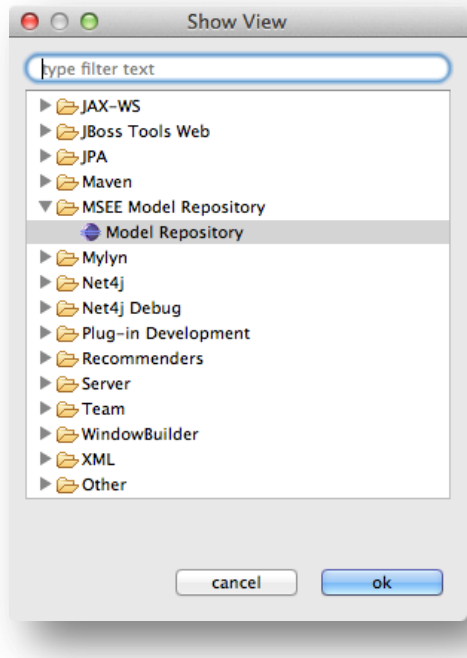

Figure 3 Displaying the Model Repository View in Eclipse

The Model Repository view is initially empty as it is not connected to any remote repository. All tools for interacting with the remote repository are available in the view's toolbar.

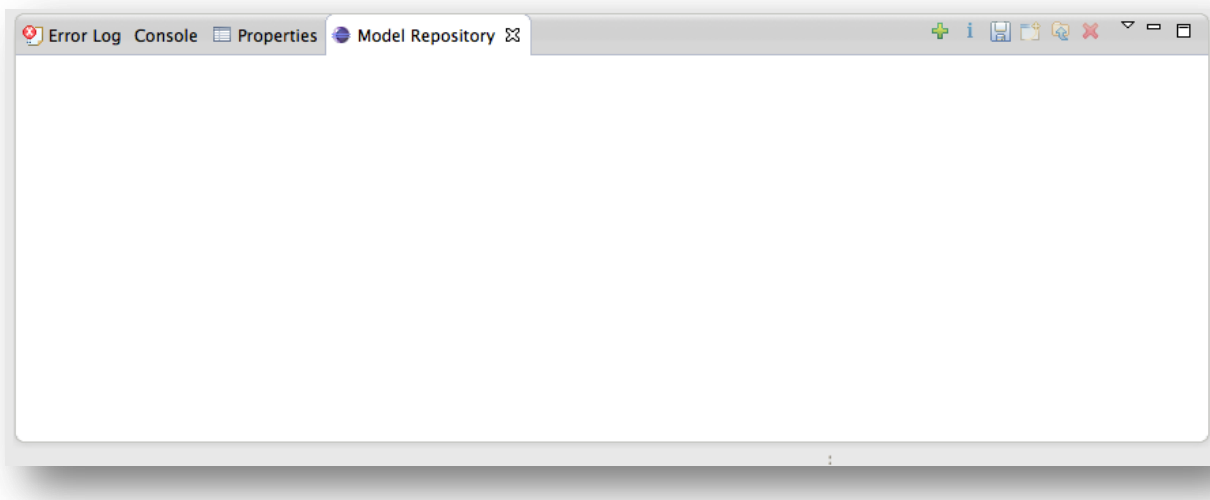

Figure 4 Model repository view - initially empty

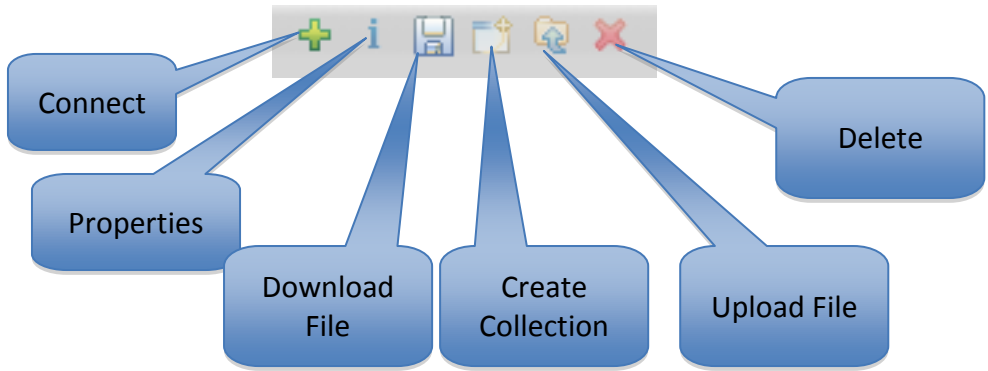

Figure 5 Model Repository View toolbar

Using the connect button, the connect dialog shows up, prompting to enter host, port and context path of the WebDAV server. In the example shown below, the WebDAV repository is hosted at 'http://192.168.33.12/repos/'.

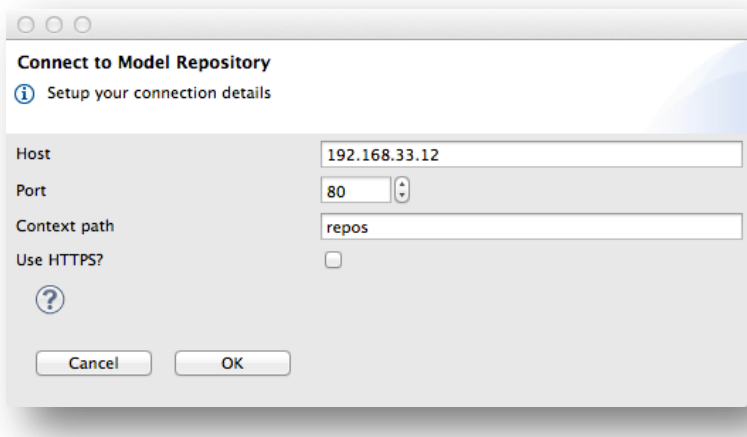

Figure 6 Connection dialog

When connected, the repository view is populated with a tree view which represents the hierarchical structure of the remote repository.

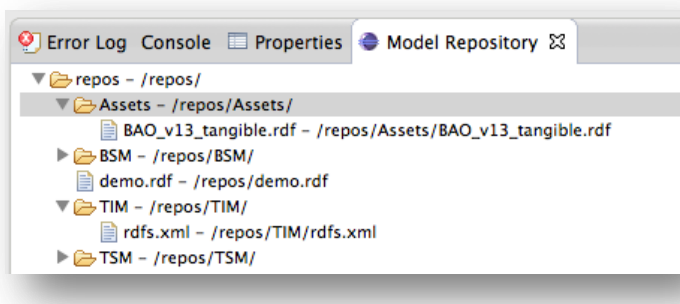

Figure 7 Sample repository structure

All further actions available are self-explanatory, allowing complete file management interaction with the Model Repository server.

7 Future plans

7.1 Implementation of access control, integration with Federated SSO Utility Service

Access control will be implemented along with Federated SSO Utility Service integration. This enhancement will allow central control of MSEE users access rights in the Model Repository according to the MSEE security model implemented by the Federated SSO Utility Service.

7.2 Distributed collaboration

Users will be able to take advantage of the distributed lock mechanism in WebDAV to enable the possibility for remote teams to collaborate while maintaining information integrity in the Model Repository.

7.3 Other enhancements

Other general improvements will be performed, as required according to usage and integration in pilot use cases.

8 References

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

[2] CDO Model Repository, http://www.eclipse.org/cdo/

[3] RFC 4918, "HTTP Extensions for Web Distributed Authoring and Versioning (WebDAV)", http://www.webdav.org/specs/rfc4918.html

[4] MSEE Deliverable D33.3 FI Utility Services First Prototype M18

[5] Vagrant, a wrapper around VirtualBox virtual machines for creating and configuring lightweight, reproducible, and portable development environments, http://www.vagrantup.com/

[6] Apache Jackrabbit content repository, http://jackrabbit.apache.org/