Simulator Development Consolidation - LoM (Library of Models)

Parsons, P.,Estades, P., Rodriguez, M.[(1)]*, Ellsiepen, P.*[(2)]

The Server Labs S.L.[(1)]
Calle Pinar, 5
28006, Madrid, Spain
E-mail: pparsons@theserverlabs.com , pestades@theserverlabs.com, mrodriguez@theserverlabs.com

Telespazio VEGA Deutschland GmbH[(2)]
Europaplatz 5
64293 Darmstadt, Germany
E-mail: peter.ellsiepen@vega.de

ABSTRACT

In order to enable the sharing of simulator models between projects the concept of a Library of Models (LoM) has been born. This library/server based repository aims to make it easier to get an overview of existing models and to ease sharing of new models.
The key features of the LoM include

- Fostering of a shared / distributed simulation development
- State-of-the-art centralised server-based repository approach
- Web based portal combined with library to allow different access models
- Modular deployment approach (via UMF solutions/projects concept)
- Full binary deployment of released components or sets of released components
- Dependency management between UMF solutions/projects
- License and user management

INTRODUCTION

The Library of Models (LoM) was born out of a need to facilitate the flow of components and runtime configurations between different simulator development projects and teams via check-in/check-out operations ant to ease the mechanism for the deployment of artefacts to deployment sites.

The management of release artefacts is well covered in modern software development platforms by the usage of standard **Repository Managers** that:

- Provide an organisation with control over what artefacts are available to both the development and operations teams.

- Facilitate the access to all types of artefacts (sources, binaries, documentation, etc.), enabling greater collaboration between development teams and wider distribution/reuse of software/components.
- Allow the definition of different repositories to distribute artefact types as required.

In the development of the SimSDE LoM it was decided at an early stage to develop the LoM Server on-top of one of these standard repositories. The one chosen for the project because of it's wide user-base and support was Sonatype Nexus [1].

The LoM system consists of a LoM Client that is integrated into UMF but that can be used standalone, a LoM Web Portal where users can search for and download simulation models, and the LoM repository itself.

LOM ARCHITECTURE

The architecture of the LoM is shown in the following diagram

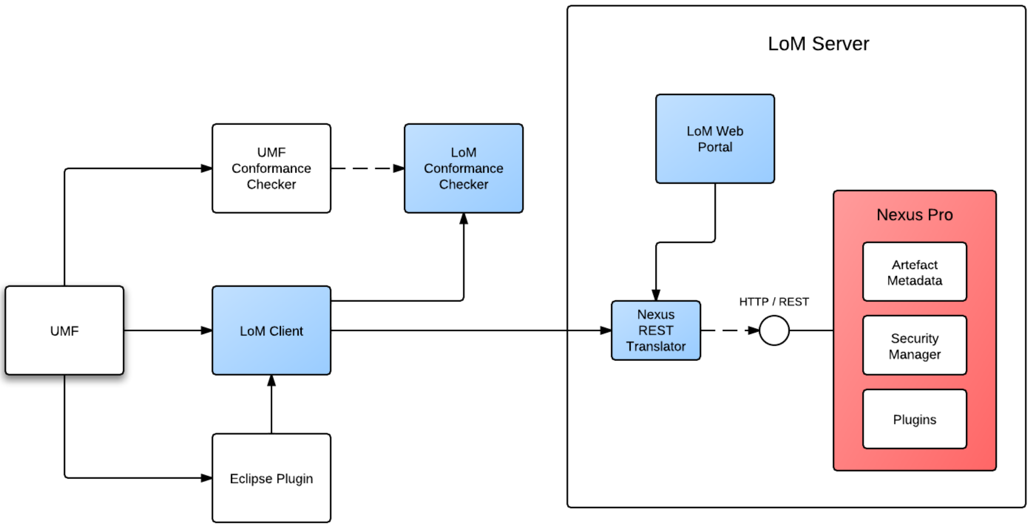


Figure 1 LoM architecture

The components of the architecture are:

- LoM Client: A Java based client that is integrated into UMF [2] by means of an Eclipse Plugin.
- LoM Conformance Checker: A module that ensures that simulation models have passed all requisite checks before they can be uploaded.

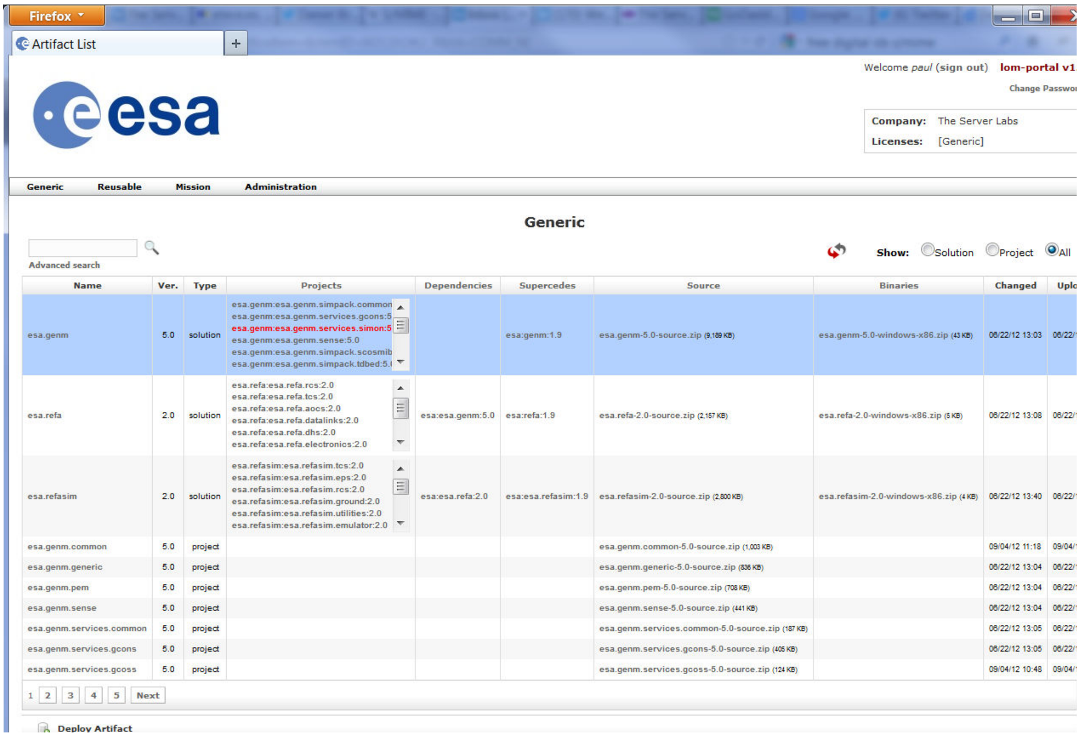
- LoM Web Portal: A web-based portal that allows developers to search for models and download them. It also provides an adminstration interface for configuring developers, companies and model licenses.
- Nexus REST Translator: A common module that is used by both the LoM Client and the LoM Portal to make requests to the repository. The idea is that if in the future a different repository was chosen, then this module would be replaced without having to make major changes to either the Client or the Portal.
- Nexus Pro: The Repository itself.

LOM FEATURES

LoM Web Portal

The principal interface for developers to the LoM Repository is via the LoM Portal. From here they can search for model solutions and projects, download source and binary artefacts and change metadata. Simulation models are split between three repositories, Generic Models, Reusable Models and Mission Specific Models.

Within the portal developers can see if models have been superceded by newer versions, and see the dependencies that the models have.



Welcome *paul* (sign out) lom-portal v1

Change Passwo

Company: The Server Labs

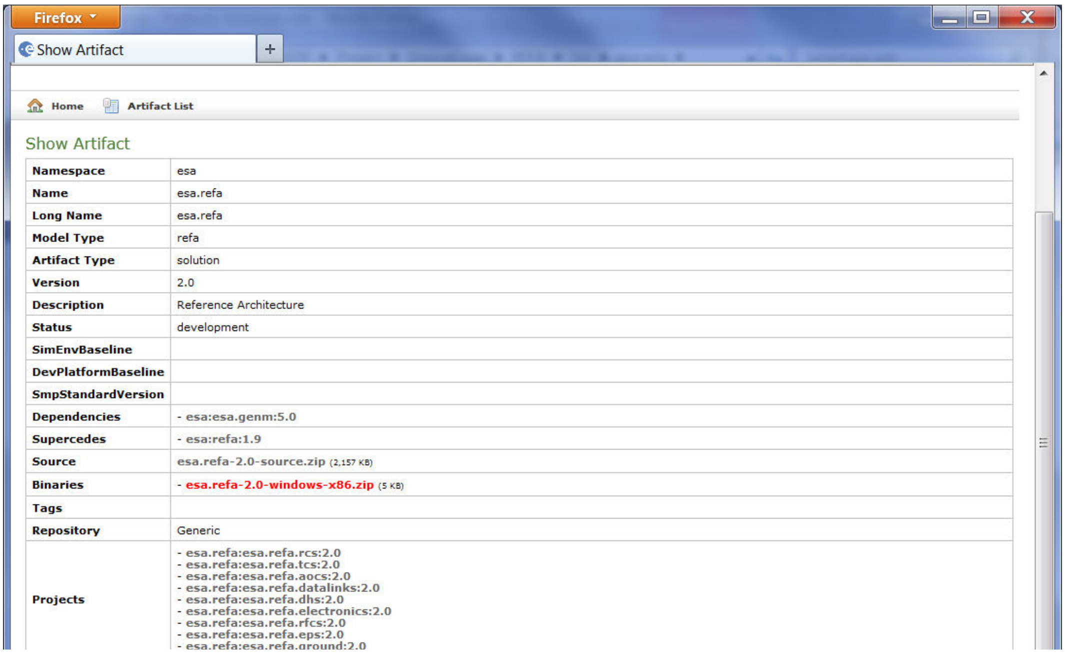
Licenses: [Generic]

Generic

Name	Ver.	Type	Projects	Dependencies	Supercedes	Source	Binaries	Changed	Uplo
esa.genm	5.0	solution	esa.genm:esa.genm.simpack.common esa.genm:esa.genm.services.gcons:5 esa.genm:esa.genm.services.simon:5 esa.genm:esa.genm.sense:5.0 esa.genm:esa.genm.simpack.scosmib esa.genm:esa.genm.simpack.tdbed:5.		esa:genm:1.9	esa.genm-5.0-source.zip (9,189 KB)	esa.genm-5.0-windows-x86.zip (43 KB)	06/22/12 13:03	06/22/
esa.refa	2.0	solution	esa.refa:esa.refa.rcs:2.0 esa.refa:esa.refa.tcs:2.0 esa.refa:esa.refa.aocs:2.0 esa.refa:esa.refa.datalinks:2.0 esa.refa:esa.refa.dhs:2.0 esa.refa:esa.refa.electronics:2.0	esa:esa.genm:5.0	esa:refa:1.9	esa.refa-2.0-source.zip (2,157 KB)	esa.refa-2.0-windows-x86.zip (5 KB)	06/22/12 13:08	06/22/
esa.refasim	2.0	solution	esa.refasim:esa.refasim.tcs:2.0 esa.refasim:esa.refasim.eps:2.0 esa.refasim:esa.refasim.rcs:2.0 esa.refasim:esa.refasim.ground:2.0 esa.refasim:esa.refasim.utilities:2.0 esa.refasim:esa.refasim.emulator:2.0	esa:esa.refa:2.0	esa:esa.refasim:1.9	esa.refasim-2.0-source.zip (2,800 KB)	esa.refasim-2.0-windows-x86.zip (4 KB)	06/22/12 13:40	06/22/
esa.genm.common	5.0	project				esa.genm.common-5.0-source.zip (1,003 KB)		09/04/12 11:18	09/04/
esa.genm.generic	5.0	project				esa.genm.generic-5.0-source.zip (836 KB)		06/22/12 13:04	06/22/
esa.genm.pem	5.0	project				esa.genm.pem-5.0-source.zip (708 KB)		06/22/12 13:04	06/22/
esa.genm.sense	5.0	project				esa.genm.sense-5.0-source.zip (441 KB)		06/22/12 13:04	06/22/
esa.genm.services.common	5.0	project				esa.genm.services.common-5.0-source.zip (187 KB)		06/22/12 13:05	06/22/
esa.genm.services.gcons	5.0	project				esa.genm.services.gcons-5.0-source.zip (405 KB)		06/22/12 13:05	06/22/
esa.genm.services.gcoss	5.0	project				esa.genm.services.gcoss-5.0-source.zip (124 KB)		09/04/12 10:48	09/04/

Figure 2 LoM Web Portal

When developers select a solution or a project, they can get more detail on the artefact, including the artefact's MetaData.



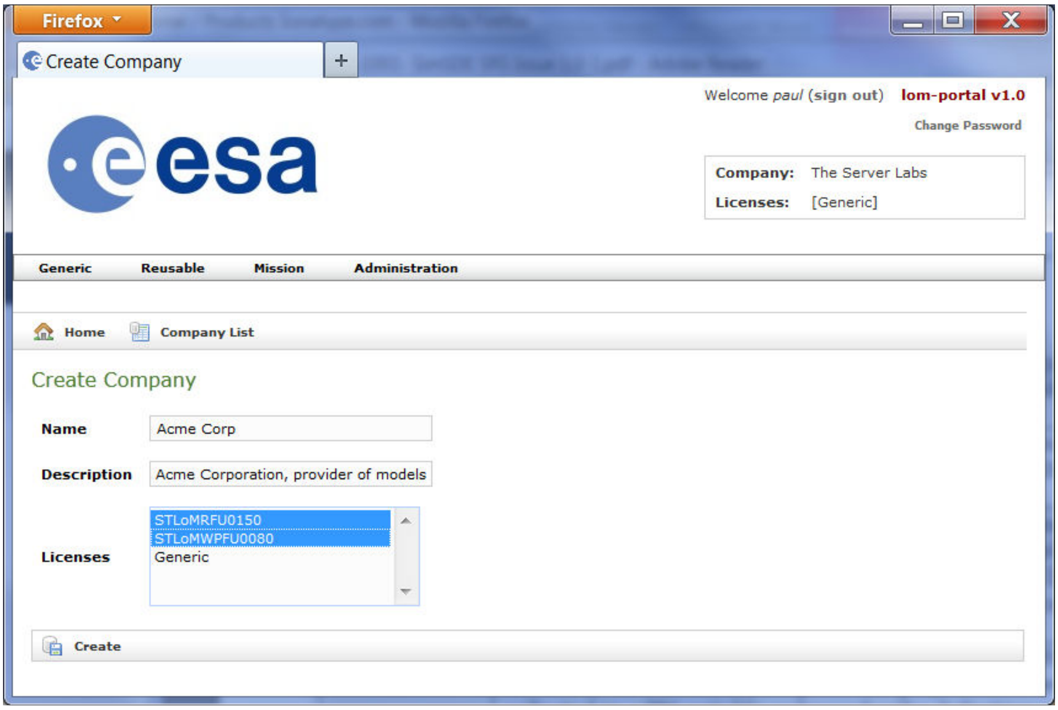
Show Artifact

Namespace	esa
Name	esa.refa
Long Name	esa.refa
Model Type	refa
Artifact Type	solution
Version	2.0
Description	Reference Architecture
Status	development
SimEnvBaseline	
DevPlatformBaseline	
SmpStandardVersion	
Dependencies	- esa:esa.genm:5.0
Supercedes	- esa:refa:1.9
Source	esa.refa-2.0-source.zip (2,157 KB)
Binaries	- esa.refa-2.0-windows-x86.zip (5 KB)
Tags	
Repository	Generic
Projects	- esa.refa:esa.refa.rcs:2.0 - esa.refa:esa.refa.tcs:2.0 - esa.refa:esa.refa.aocs:2.0 - esa.refa:esa.refa.datalinks:2.0 - esa.refa:esa.refa.dhs:2.0 - esa.refa:esa.refa.electronics:2.0 - esa.refa:esa.refa.rfcs:2.0 - esa.refa:esa.refa.eps:2.0 - esa.refa:esa.refa.ground:2.0

Figure 3 Artifact Details showing MetaData

In order to give company developers access to simulator models, the LoM has the concept of users, companies and licenses.

A company such as The Server Labs, has a number of licenses that it can use. Each of these licenses contain one or more simulation namespaces. A license for simsat for example, would give acces to the namespaces esa.simsat, esa.refa, esa.genm and esa.smp2.



Company: The Server Labs
Licenses: [Generic]

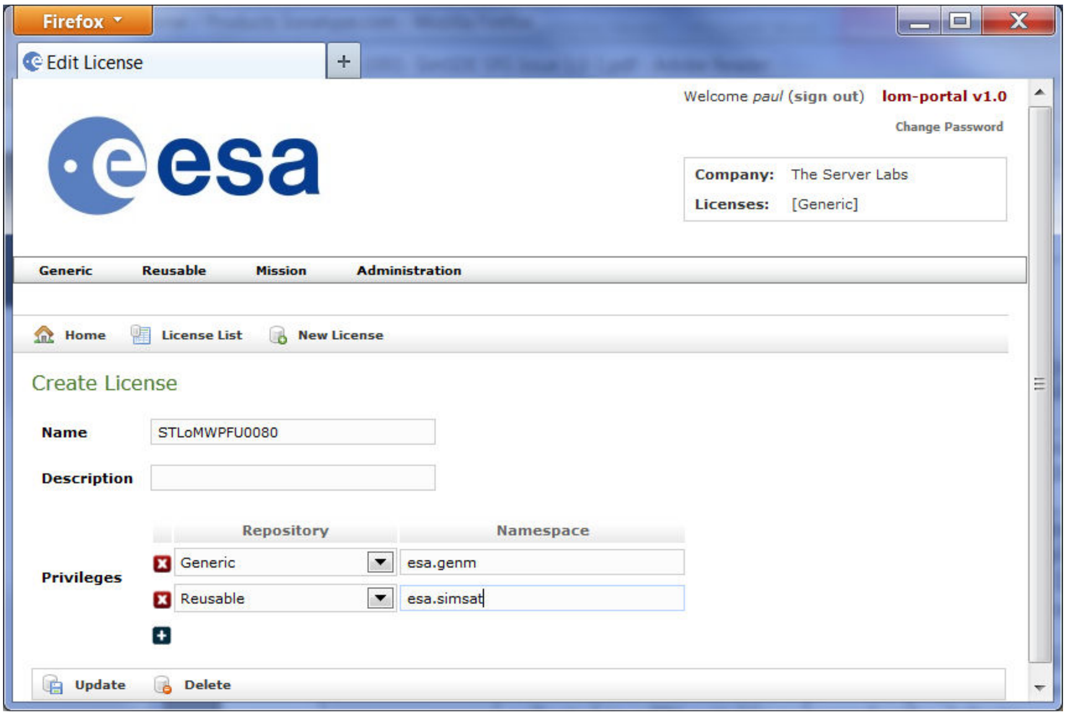
Create Company

Name: Acme Corp

Description: Acme Corporation, provider of models

Licenses: STLoMRFU0150, STLoMWPFU0080 (selected); Generic

Figure 4 Company Licenses



Company: The Server Labs
Licenses: [Generic]

Create License

Name: STLoMWPFU0080

Description:

Privileges

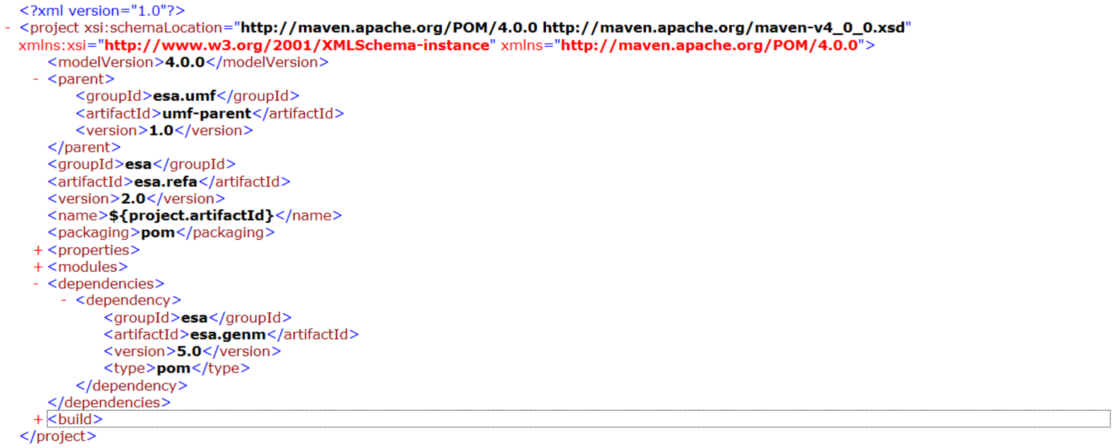
Repository	Namespace
Generic	esa.genm
Reusable	esa.simsat

Figure 5 License Namespaces

Dependency Management and Upload

The LoM manages dependencies using the Maven dependency mechanism that is standard in Repository Managers. In the case below, esa.refa depends upon esa.genm, which in turn depends upon esa.smp2. This information is stored in Maven pom files at the solution level.

esa.refa → *esa.genm* → *esa.smp2*



```
  <?xml version="1.0"?>
- <project xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://maven.apache.org/POM/4.0.0">
     <modelVersion>4.0.0</modelVersion>
   - <parent>
        <groupId>esa.umf</groupId>
        <artifactId>umf-parent</artifactId>
        <version>1.0</version>
     </parent>
     <groupId>esa</groupId>
     <artifactId>esa.refa</artifactId>
     <version>2.0</version>
     <name>${project.artifactId}</name>
     <packaging>pom</packaging>
   + <properties>
   + <modules>
   - <dependencies>
      - <dependency>
           <groupId>esa</groupId>
           <artifactId>esa.genm</artifactId>
           <version>5.0</version>
           <type>pom</type>
        </dependency>
     </dependencies>
   + <build>
  </project>
```

Simulator model solutions are uploaded or deployed to the repository using Maven (mvn deploy). Maven ensures that the LoM Conformance Checker passes, packages up the source and binary artefacts and metadata and deploys everything to the Repository.

UMF Integration

The integration with UMF is via the LoM Client which is made available within UMF via an Eclipse Plugin. The LoM Client provides the same functionality that is avaiable within the Portal: search, download, showing metadata and dependencies. In order to upload models, UMF uses Maven directly.

TEST AND DEVELOPMENT IN THE CLOUD

All the test and development for the LoM has been done in the Cloud, with test versions being deployed regularly for ESOC to test in Amazon’s EC2 Cloud.

CONCLUSION

The current version of the LoM is just the first phase. Future phases will include mirroring of repositories between developer sites and ESA with the possibility to promote models from one site to another. It is also envisaged that the LoM could be turned into a product that will be of interest outside of ESA.

REFERENCES

[1] Sonatype Nexus http://www.sonatype.com/Products/Nexus-Professional

[2] UMF Release 1.1.0 SRN, EGOS-SIM-SIM-SRN-1001, 1.7, 2010-12-03