



|  |
| --- |
| **D5.1.3 Model Fragmentation Use Case Definition**Galaxy use case definition |



|  |  |  |  |
| --- | --- | --- | --- |
|  | NamE | partner | Date |
| Written by | L. BARESSE | AKKA |  |
|  |  |  |
|  |  |  |
| Reviewed by |  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Record of Revisions

| Issue | Date | Effect on | Reasons For Revision |
| --- | --- | --- | --- |
| Page | Para |
| 01A |  |  |  | Document creation |
|  |  |  |  |  |

Table of contents

[1. Introduction 5](#_Toc302989590)

[1.1 Goal of this document 5](#_Toc302989591)

[1.2 Document organization 5](#_Toc302989592)

[2. SCope 5](#_Toc302989593)

[2.1 Description of the <XX> system 5](#_Toc302989594)

[2.2 Scope of the study 5](#_Toc302989595)

[3. Validation method 5](#_Toc302989596)

[4. Involved PARTNERS 6](#_Toc302989597)

[5. Validation ScenariOS 6](#_Toc302989598)

[6. Involved models 6](#_Toc302989599)

[7. tools used 6](#_Toc302989600)

Table of APPLICABLE DOCUMENTS

| N° | title | Reference | Issue | Date | Source |
| --- | --- | --- | --- | --- | --- |
| Siglum | Name |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

Table of ReferenceD DOCUMENTS

| N° | title | Reference | Issue |
| --- | --- | --- | --- |
|
| 1.
 | Galaxy glossary | D1.2.2 |  |
|  | Goal and metrics document | D1.1 |  |
|  | Architecture specification | D4.1 |  |
|  |  |  |  |

ACRONYMS AND DEFINITIONS

Except if explicitly stated otherwise the definition of all terms and acronyms provided in [R1] is applicable in this document. If any, additional and/or specific definitions applicable only in this document are listed in the two tables below.

Acronymes

|  |  |
| --- | --- |
| Acronym  | DESCRIPTION |
| MF | Model Fragment  |
| MFC | Model Fragment Client |
|  |  |
|  |  |

Definitions

|  |  |
| --- | --- |
| TERMS | DESCRIPTION |
|  |  |
|  |  |
|  |  |
|  |  |

# Introduction

The goal of the Galaxy project is to work on the technical hard points related to the fragmentation and to the distributiveness of huge models, and to their synchronization in regards of the communication means classically used by development teams. Galaxy partners believe that a set of technical solution integrated in a common platform of service (called: “the Galaxy platform”) may greatly help in dealing with these hard points.

Based on a selected subset of candidate technologies, the architecture of the platform has been specified. As planned in the project proposal this platform is assessed using use cases where scalability issues can be identified and characterized.

## Goal of this document

A specific task of the Galaxy project (T5.1) is dedicated to the definition of the use cases. This document is a product of this task which describes the Model Fragmentation study case.

## Document organization

The chapter 2 describes the context of the test case, including an overview of the domain and a focus on the scope on which of particular interest for the study.

Chapter 3 explains how the approach we have selected to verify and validate the added value of the services provided by the Galaxy platform.

Chapter 4 presents the partners involved on this study case and the way they have contributed to it.

Chapter 5 specifies the scenarios which are used to assess the performance offered by the Galaxy platform.

Chapter 6 describes the models involved in validation scenarios are played: viewpoints, views, sizes and organizations.

Last, Chapter 7 is about the tools software tools used for the validation scenarios.

# SCope

## Description of the model fragmentation system

Describe here the context of the domain of the study case and specific concepts of interest which helps in getting a better understanding of it.

Model Fragments are an important part of the Galaxy platform. They represent model elements that are a subset of a larger model. Any participant can create a Model Fragment and publish it to the Galaxy repository. Another Participant can later fetch these Model Fragments into their model.

Model Fragments are versioned and have the particularity to be in a non mutable state once they are deployed. So the elements of a first model, represented by a Model Fragment, can be referenced in another one. Also, Model Fragments can have dependency between them, meaning that importing a Model Fragment with such dependencies, all the Model Fragments it depends on must have been imported as well.

The Model Fragment technology is a mean to fragment huge models into a subset of smaller models, with a robust versioning and inter fragment dependency system. This technology emphasizes on encapsulation, which always is a good point when working on engineering projects.

A Model Fragment is a published archive of a subset of models. This subset should be lighter than the original models (filter mechanism must be setup: e.g. only public model elements will be included in the Model Fragment archive). A Model Fragment is tightly coupled to a set of model elements which are managed by a logical team (not everybody can work on the model elements). Should another team need to reference or to use model elements coming from this set of model elements, this last team will have to get the published archive of the Model Fragment.

It means only a logical team will have access to the real set of model elements in read/write mode. This logical team will have to publish versioned archives into the Galaxy repository in order to collaborate with other teams.

This archive is published in a non mutable state (read only mode) on a Galaxy repository. This archive will be then used by other users/entities without having access the real model elements details.

## Scope of the study

Focus here on the perimeter directly involved in the study case. Describe the scalability issues which are used to assess the added value of the platform

This use case will focus on the Model Fragment contribution for the Galaxy architecture. This contribution mainly addresses scalability issues and is part of the core solution provided by Galaxy. Model Fragment covers the following requirements as stated in the D.4.1 document:

* GlxR\_Coll\_14: The system shall support bottom-up development approaches
* GlxR\_Intg\_01: The system shall allow connecting heterogeneous tools to exchange models or parts of model (i.e. building blocks) between them
* GlxR\_Intg\_02: The system shall be connectable to the Softeam Modelio tool
* GlxR\_Intg\_03: The system shall be connectable to the Eclipse MDT Papyrus tool

# Validation method

Explain the approach and the process applied on this study case.

As previously stated there are two main scenarios. Both have to be validated and are loosely coupled.

The first scenario is related to the Model Fragment edition and publication. It is the Model Fragment producer point of view. During this scenario there is a dedicated team of Galaxy users which work on the model elements which compose a Model Fragment. These users should be able to work on their fully detailed model elements without constraint: read / write, commit / check out, etc. Should their work could be published as a valid version of the models, thanks to the Galaxy agent in their IDE, they will be able to create and publish a Model Fragment to do it.

The second scenario will focus on the Model Fragment consumer point of view. Thanks to Galaxy agent in their IDE, users will be able to find and retrieve published Model Fragments for a specific version and all their dependencies (which are also versioned published Model Fragments and their dependencies).

# Involved PARTNERS

All partner directly involved in this study case are listed here with their respective contribution.

The involved partners are Airbus, AKKA and Softeam.

# Validation ScenariOS

Describe the validation scenarios here:

* Involved service of the Galaxy platform
* procedure,
* assessed metrics and corresponding objectives

## Model fragment edition & publication

It is the Model Fragment producer point of view.

During this scenario there is a dedicated team of Galaxy users which work on the model elements which compose a Model Fragment. These users should be able to work on their fully detailed model elements without constraint: read / write, commit / check out, etc.

Should their work could be published as a valid version of the models, thanks to the Galaxy agent in their IDE, they will be able to create and publish a Model Fragment to do it.

### Initialize a Model Fragment definition

This scenario aims to assess:

* Ability to set and update metadata for a Model Fragment definition.

Model Fragment Client (MFC) shall permit users to set and update metadata for a MF. These metadata will be stored in the MF dependency file. Metadata are name, description and version.

### Add/Remove/Modify model elements to/from a Model Fragment definition

This scenario aims to assess:

* Ability to add model elements to a Model Fragment definition.
* Ability to remove model elements from a Model Fragment definition.
* Ability to update model elements in a Model Fragment definition.

Model Fragment definition shall contain the complete list of model elements which shall be packaged and then published with the Model Fragment. MFC shall support users during this task and at least raise error message should something impossible is required during this phase. For instance, at least one model element depends of something which is not in the current MF and not in another MF described in the dependency list.

### Add/Remove/Modify dependency to/from a Model Fragment definition

This scenario aims to assess:

* Ability to add a Model Fragment dependency to a Model Fragment definition.
* Ability to remove a Model Fragment dependency from a Model Fragment definition.
* Ability to update a Model Fragment dependency in a Model Fragment definition.

Model Fragment Client (MFC) shall permit users to set and update dependencies information for a MF. These metadata will be stored in the MF dependency file. A dependency is defined by the MF (qualified) name and its version. This couple shall be unique in the Galaxy Repository.

Note: Transitive dependencies shall be handled by the Galaxy architecture. As a consequence, if A depends of B and B depends of C; a project (or a MF) P which declare a dependency on A will automatically depends of B and C. It’s not required for P to declare explicit dependencies to B and C.

### Add/Remove/Modify file to/from a Model Fragment definition

This scenario aims to assess:

* Ability to add a file to a Model Fragment definition.
* Ability to remove a file from a Model Fragment definition.
* Ability to update a file in a Model Fragment definition.

Model Fragment Client (MFC) shall permit users to add, remove and update files which will be packaged with the MF. These files are not managed by the Galaxy architecture, it’s only a convenient way to share unverified files related to the MF it-self such as PNG files to exchange diagrams, but also generated source code files, etc.

### Create and package a Model Fragment from its definition

This scenario aims to assess:

* Ability to create and package a Model Fragment according to its definition.

Model Fragment Client (MFC) shall permit users to create and package a MF according to its previously established definition. During this phase the model elements should be simplify in order to not publish irrelevant information (e.g. only public model elements should be exported during the MF creation process).

A Model Fragment shall be packaged in a ZIP file format to facilitate its exchange and storage on the Galaxy Repository via the Galaxy Server. The ZIP file shall contain the exported model elements, the list of files described in the MF definition and a dependency file which must contain all dependencies declarations and metadata information.

### Publish a Model Fragment on the Galaxy Repository

This scenario aims to assess:

* Ability to publish a Model Fragment archive on the Galaxy Repository via the Galaxy Server.

Model Fragment Client (MFC) shall permit users to publish on the Galaxy Repository via the Galaxy Server a packaged MF. The packaged MF file shall contain the exported model elements, the list of files described in the MF definition and a dependency file which must contain all dependencies declarations and metadata information.

Should a MF with the same qualified name and version already published on the Galaxy Repository, the Galaxy server shall return an error to the MFC. Published MF are non mutable. It shall be impossible to update / delete them from the Galaxy Repository. The reason is since a MF is published on the Galaxy Repository, it becomes public. Anyone can retrieve and use a public MF published on the Galaxy Repository. It’s therefore a non-sense to think about updating or deleting something that someone could have been retrieved. Should anything required to be modified in a MF (e.g. the Model Fragment Foo version 1.0 published on the Galaxy Repository is inconsistent), a new version of this MF shall be published on the Galaxy Repository.

## Model fragment find & retrieve

### Find a published Model Fragment from the Galaxy Repository

This scenario aims to assess:

* Ability to find a published Model Fragment from the Galaxy Repository.

Model Fragment Client (MFC) shall permit users to find a published Model Fragment from the Galaxy Repository. MFC shall be able to retrieve all the published MF in all available versions, or all with a matching name or the MF with a matching name and an available matching version.

### Retrieve from the Galaxy Repository a published Model Fragment for a given version

This scenario aims to assess:

* Ability to retrieve form the Galaxy Repository a published Model Fragment for a given version and all its dependencies.

Model Fragment Client (MFC) shall permit users to retrieve form the Galaxy Repository a published Model Fragment for a given version. Should the MF have dependencies, these MFs shall also be retrieved within the same operation (transitive dependencies mechanism shall be implemented here).

# Involved models

List and describe the kind of models (view, viewpoint) involved in the study case and the way they are organized (relationships)

To be completed

# tools used

List the tools used in this study case with their corresponding version and purpose.

The Model Fragment infrastructure shall be available since the 1st validation phase, the following tools will be used:



* Galaxy Server
	+ Model Fragment Engine

* Galaxy Repository
* Topcased/Papyrus based MDE Tool
	+ Galaxy Agent
		- Model Fragment Client
* Modelio based MDE Tool
	+ Galaxy Agent
		- Model Fragment Client