



**D5.4.1-Large and-or Numerous Models**

Case Study Implementation

|  |
| --- |
| **D5.1.x <XX> Use Case Definition**D5.4.1-Large and-or Numerous Models Case Study ImplementationGalaxy use case definiti |



|  |  |  |  |
| --- | --- | --- | --- |
|  | NamE | partner | Date |
| Written by | W. KLING | AtlanMod | 02/02/2012 |
|  |  |  |
|  |  |  |
| Reviewed by |  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Record of Revisions

| Issue | Date | Effect on | Reasons For Revision |
| --- | --- | --- | --- |
| Page | Para |
|  | 02/02/2012 |  |  | Document creation |
|  |  |  |  |  |

Table of contents

1. Introduction 5

1.1 Goal of this document 5

1.2 Document organization 5

2. SCope 5

2.1 Description of the Webml Metamodel 6

2.2 Scope of the study 6

3. Validation method 7

4. Involved PARTNERS 7

5. Validation ScenariOS 7

5.1 Representation of complex models relations and ITS automatic discovery 7

5.2 gathering Metrics from multiple models 7

5.3 Selective model transformation execution 7

6. Involved models 8

7. tools used 8

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 |
|  |  |
|  |  |
|  |  |
|  |  |

Definitions

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

# Introduction

One of the goals of the Galaxy project is to provide a conceptual and technical solution to deal with large amounts of models effectively specifically regarding to models transformations of complex and heterogeneous models arrangements. This technical solution has been envisioned as a textual domain specific language (DSL) called MoScript, which has been specified and implemented as a prototype described in **D4.5.2 - Megamodel for transformations prototype**.

In order to validate the technical solution, a set of use cases have been design, and implemented to be executed with MoScript. This uses cases enable the MoScript validation in terms of functionality. The goal is to show how certain models manipulations may be performed with less effort and in a more consistent model driven oriented way than with current approaches.

## 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 **D5.4.1 Large and-or Numerous Models** study case.

## Document organization

The chapter 2 describes the context of the tests cases and explains how are them valuable for verifying the added value of MoScript to the Galaxy platform.

Chapter 3 describes the scenarios being used to assess the effort reduction for understanding and manipulating repositories of large numbers of models.

Chapter 4 describes the technical environment and tools used in the elaboration and execution of the test cases.

# SCope

The use cases proposed in this document aim to validate MoScript, a DSL for querying and manipulating model repositories, as a proper contribution for addressing the problems of scalability due to the large numbers of models and heterogeneity of development environments found in the development and maintenance of model driven complex systems.

Before going ahead with the environment description, it is important to make clear that for testing MoScript we are interested in the structure and physical characteristics of the model repository more than the system that is represented by the models in the repository.

For the purpose of this deliverable, we choose a WebML [1] project and the ATL transformation Zoo as testbed for validating that MoScript can help users to deal effectively with large amounts of complex and interrelated models.

WebML is a domain specific language and web applications design methodology mainly targeted for model driven design and development of data intensive Web applications i.e. web sites for accessing and maintaining large amounts of structured data usually stored in RDBMS.

WebML is implemented by the privative WebRatioTM tool, which is able to generate fully functional applications from WebML diagrams and is free for teaching and research purposes. WebML is successfully proven DSLs for Model Driven Development for data intensive systems.

Depending on the size of the system represented by a WebML model, WebML models may range from small to big (several Mb) models with hundred or thousands of components.

WebML projects count with the following characteristics that we find suitable to test MoScript:

* **Heterogeneity**: Although WebML is considered to follow a MDD approach; the WebRatio model editor stores the WebML model in a XML format with no reference to a predefined XML schema or DTD. In order to manipulate the WebML model we need first to bridge from the xmlware to a modelware [2] technical space.
* **Large amount and complex interrelations of models**: Although a WebML model can be seen as a single model, the WebRatio tool splits the WebML model in several sub models each one with a well-defined purpose. This fact, enable us to see a WebML project as model repository with complex interrelations.

The ATL transformation Zoo[[1]](#footnote-1) is large repository of models and ATL transformations. It counts also with characteristics that make it suitable for testing MoScript:

* **Heterogeneity**: As shown in Figure 1 the ATL Zoo counts with many model artefacts in different formats, which reflects the same situation we are going to find in any model driven complex system.



Figure : ATL Transformation Zoo modelling artefacts distribution

* **Large amount and complex interrelations of models**: The ATL Zoo counts with a total of 1704 modelling artefacts that all of them playing a specific role in the repository. From them, 259 artefacts correspond to ATL transformations, all of them related to models and metamodels and in some cases to other transformations by transformation chains.

# Validation ScenariOS

## Representation of complex models relations and ITS automatic discovery

One of the complexities introduced by MDE is the inclusion of new artefacts in the development process such as models, metamodels, transformations and the strong dependencies between them. This characteristic combined with the fact that model driven complex systems usually require large numbers of models to form up the system, makes this kind of systems difficult to understand and thus not suitable for collaboration. In order to easily understand those systems we require ways to view or query the complex model arrangements for extracting the information particular interests.

MoScript proposes the use of a Megamodel as a view describing the models of complex systems and its interactions, which may be queried in different way, facilitating the understanding of the model repository representing the system. Nevertheless, before being able to query the Megamodel, first it has to be populated. Each model, metamodel, transformation etc. and their relations must be registered in it. Doing this task manually is time consuming and error prone.

In this scenario we demonstrate how MoScript facilitates the automation of the population of the Megamodel through different features like the projection of existing artefacts as models to be able to query their content for inferring its role in the repository, and by providing operations to populate the megamodel with new model elements describing the repository artefacts.

Scenario description

## gathering Metrics from multiple models

Another characteristic of model driven collaborative systems is that developers may require to conveniently navigate between several models and even to consider several models simultaneously, for example to compare, align them or extract some kind of metrics from them.

MoScript allows the inspection of several models at the same time and thus the combination of their information to extract convenient metrics from model repositories facilitating its understanding.

In this use case we demonstrate how is possible with MoScript to gather information from several interrelated models of a big repository.

Scenario description

## Selective model transformation execution

Model driven collaborative systems are in constant evolution, although some of them can be considered to be minor as they only impact one particular project or few components other evolutions may be substantial impacting a large set of projects and components. Thus, developers cannot rely on static views of the repository, neither use traditional scripts that do not evolve along with the repository for executing model manipulations such as transformations.

MoScripts allows executing model manipulations according to the results of querying constantly updated repository view. For instance, this means that the transformations to be executed may be selected according to their content or the content of the models with which they are related etc. This gives a great flexibility at the moment of manipulating models avoiding the constant update of model manipulation scripts.

In this use case, we show how is possible with MoScript to select the transformations to be executed according to the content of the models.

Scenario description

# tools used

ATL VM

AM3

MoScript prototype

# REFERENCES

[1] Marco Brambilla, Sara Comai, Piero Fraternali and Maristella Matera, Designing Web Applications with Webml and Webratio in Web Engineering: Modelling and Implementing Web Applications, Human-Computer Interaction Series, 2008, Part II, 221-261.

[2] Wimmer, M., Kramler, G.: Bridging grammarware and modelware. In: Satellite Events at the MoDELS 2005 Conference, pp. 159–168. LNCS (2006)

1. http://www.eclipse.org/m2m/atl/atlTransformations/ [↑](#footnote-ref-1)