Project acronym: MODELPLEX

Project full title: MODELling solution for comPLEX software systems

Contract n° 034081

Workpackage 3: Model Engineering

Deliverable D3.1.b.V5 - Model Weaving

Project co-funded by the European Community under the “Information Society Technology” Programme

Contract Number: 034081

Project Acronym: MODELPLEX

Title: MODELling solution for comPLEX software systems

Deliverable N°: D3.1.b

Due Date: 10/2009

Delivery Date: 10/2009

Short Description:

Prototype on Model Weaving

Lead Partner: Technische Universität Dresden (TUD)

Made available to: Public

Rev	Date	Author	Checked by	Internal Approval	Description
0.1	09/10/09	Jendrik Johannes, TUD			Initial version
0.2	18/10/09	Mirko Seifert, TUD			Content for Section 6
0.3	20/10/09	Matthias Schmidt, TUD			Content for Section 5
0.4	20/10/09	Jendrik Johannes, TUD			Added missing parts
	29/10/09	Jendrik Johannes, TUD		**1st Internal Reviewer:** Andy Evans, Xactium **2nd Internal Reviewer:** Stéphane Ménoret, TRT	Updates corresponding to reviews received
1.0	30/10/09	Final Deliverable			

Table of Content

1. Executive Summary 5
2. Introduction 5
3. Glossary of Abbreviations and Terms 5
4. Installation and Requirements 6
5. Facet-based Fragment Library 7
   5.1. Fragment Classification 7
   5.2. Fragment Browsing 8
   5.3. Fragment Reuse 9
6. Round-trip support 10
7. Conclusion 12
8. References 13

List of Figures

Figure 1 – Installing Reuseware through the update manager in Eclipse Galileo 6
Figure 2 – Classifying CIM fragments 8
Figure 3 – Selecting facet values for the protocol facet of the EthernetPort fragment 8
Figure 4 – Search process using the Fragment Browser 9
Figure 5 – Reusing fragments in a Reuseware Composition Program 10
Figure 6 – Reuseware applied to the TIS case study 11
Figure 7 – Derived integrated security model 12

1. Executive Summary

This report accompanies the model weaving tool (Reuseware Composition Framework) as month 38 (M38) incarnation of MODELPLEX Deliverable D3.1.b "Model Weaving". It explains how to acquire and run the tool which is an extension of the version delivered in month 26. The new features are explained and demonstrated on examples based on the TID and TIS case studies.

2. Introduction

This document describes the model weaving tool (the Reuseware Composition Framework[1]) that is the main artefact of this deliverable. In this document we show the usage of the tool in the context of the TID and TIS case studies.

With the Reuseware Composition Framework, *composition systems*[2] can be modelled for arbitrary languages. These systems can then be used to weave and compose *model fragments*[3]. This document demonstrates the new features of the prototype. It neither discusses nor introduces details of the concepts behind the composition framework. Consult the M14 report about those details, as well as [1] in which the results of this work have been published. A detailed revised version of the M14 report will be delivered in M41.

The remainder of this document is structured as follows. Section 3 contains a glossary of abbreviations. How to install the tool is explained in Section 4. Section 5 demonstrates the first new feature—the fragment library—on the TID case study. Section 6 demonstrates the second new feature—round-trip support—on the TIS case study. Section 7 concludes this document.

3. Glossary of Abbreviations and Terms

- CIM – Common Information Model
- DSL – Domain Specific Language
- RIO – Risk, Impact, Objective
- TID – Telefónica Investigación y Desarrollo
- TIS – Thales Information Systems
- TRT – Thales Research & Technology
- TUD – Technische Universität Dresden
- WP – Work Package

[1] http://reuseware.org
[2] A composition system (as defined in M18 prototype document) defines which models are reusable units and at what places these models can be extended during composition.
[3] Model fragments (as defined in M14 report) are reusable models that may contain variation points.

4. Installation and Requirements

The Reuseware Composition Framework consists of a set of Eclipse plug-ins that require an Eclipse Platform installation[4] of version 3.5 (Galileo) or higher. The framework can be installed within the Eclipse platform using the update manager (*Help > Install New Software…*).

To install the framework use the following URL in the *Work with* field:

http://reuseware.org/update

Select all components of the *Reuseware Composition Framework* category as shown below. Press *Next >* and follow the installation instructions. Missing dependencies will be automatically installed from the Galileo update site.

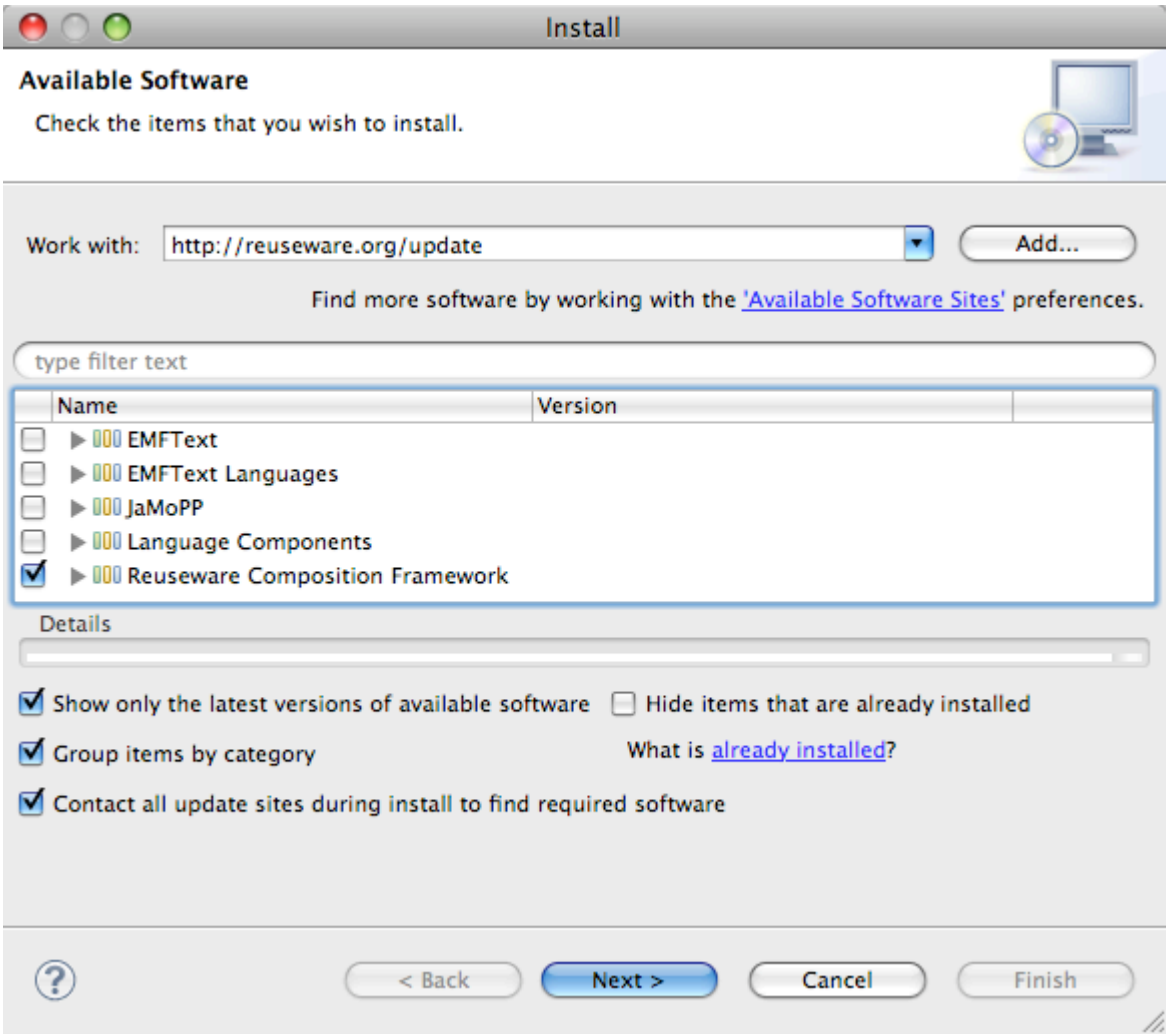

Figure 1 – Installing Reuseware through the update manager in Eclipse Galileo

.

[4] http://www.eclipse.org/downloads

5. Facet-based Fragment Library

This section introduces the new fragment library feature of Reuseware using the TID case study. With this new feature we address the part of Requirement 244 [2] that concerns library support. Concretely, Requirement 244 states [2, p. 39]:

> "Other important features for the [Teléfonica] DSL Tool would be: Library creation – to allow easy modelling in high levels of abstraction based on a pre-created library of lower-level components, like network devices, etc."

To provide a library that allows an intuitive retrieval of lower-level fragment components, the Fragment Library consists of three major features. First, classifying model fragments to prepare them for later reuse. Second, browsing a set of classified fragments. Third, retrieving fragments for reuse in a Reuseware composition program. The Fragment Library is based on the ideas of facetted classification that is well known from traditional libraries and was considered for software libraries often (e.g. in [3,4]).

In the following, we take a closer look at the three features using the TID case study. Thereby, Section 5.1 discusses the faceted classification of fragments that are defined in the TID DSL [5] based on the Common Interface Model (CIM) [6]. Section 5.2 presents the Fragment Library's faceted browser to search for classified fragments in the repository. Finally, Section 5.3 shows how to reuse fragments in Reuseware composition programs (which in the context of the TID case study correspond to CIM models on a higher abstraction level).

5.1. Fragment Classification

Figure 2 (b) shows a CIM model that is modeled in the TID DSL [5]. This DSL is based on the CIM standard [6] and allows the modeling of network devices and networks in a great level of detail. By putting the model into a Reuseware Fragment Store, it becomes a CIM *fragment*. CIM fragments can be reused and combined into new CIM models with Reuseware. For example, CIM fragments that model network devices can be reused for CIM models representing complete telecommunication networks.

Figure 2 also depicts how to classify a CIM model for reuse through the fragment library. After marking a folder as Fragment Store (a), the fragment shown in the TID DSL editor (b) can be classified using the Fragment Description View (d). This view allows to edit free-text attributes (d) and to select facets for faceted classification (e). After selecting a facet an additional view is presented (cf. Figure 3) showing the values grouped in this facet. Here, one has to select the values that describe the fragment best. This value will then be stored and form one part of the faceted classification. Furthermore, buttons on the left (f) allow to edit or delete selected facet-value-pairs or to clear the whole list. After a number of fragments were classified a separate Eclipse View can be used to browse the repository (g).

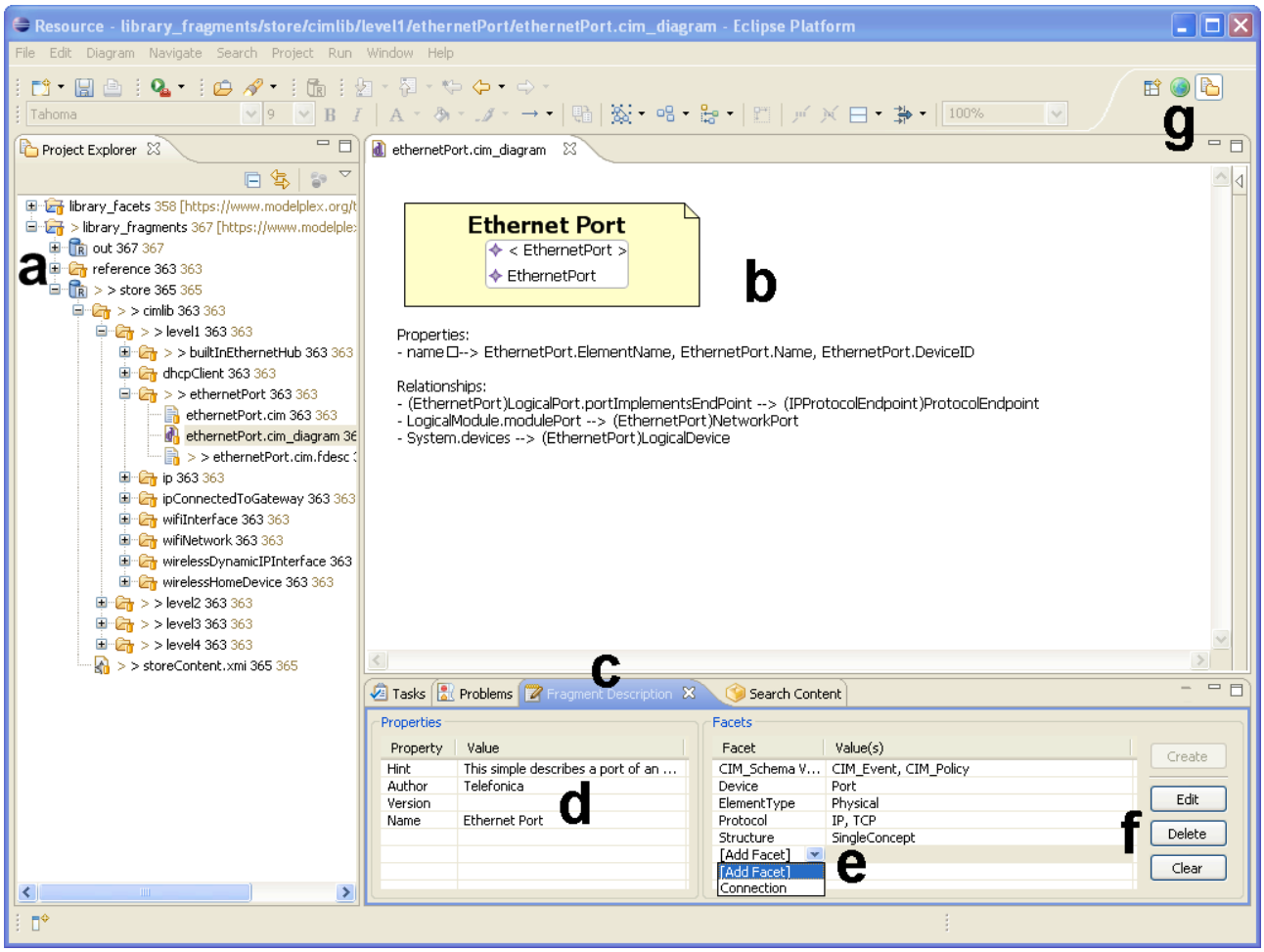

Figure 2 – Classifying CIM fragments

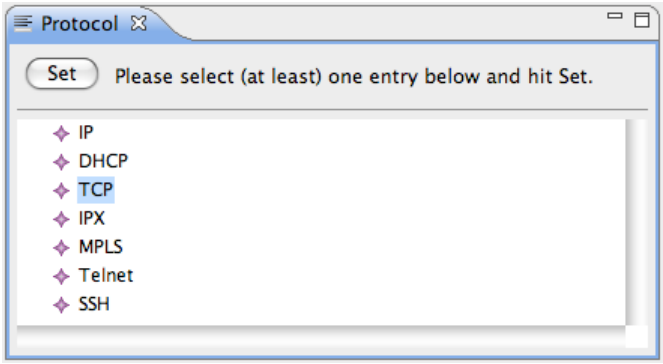

Figure 3 – Selecting facet values for the protocol facet of the EthernetPort fragment

5.2. Fragment Browsing

The Fragment Library provides a Fragment Browser that allows faceted browsing on the base of faceted classifications (cf. Figure 4). Using facet widgets on the left and right, one can select a facet value (a) to filter upon this criterion. The Search Content View (b) presents fragments that meet this criterion and selecting new values makes this list shrink (called zoom-in step). One can hit the Reset button (c) to deselect a value and perform a zoom-out step. Facet values not presented in widgets are listed in the Selected Facets View (d) and the X button (e) can be used to delete this criterion and perform the corresponding zoom-out. Furthermore, the restart button (f) allows starting the browsing from scratch and the reuse button (g) gives the opportunity to reuse a fragment selected in the list. However, this button cannot be used in this perspective and is therefore described in the following section.

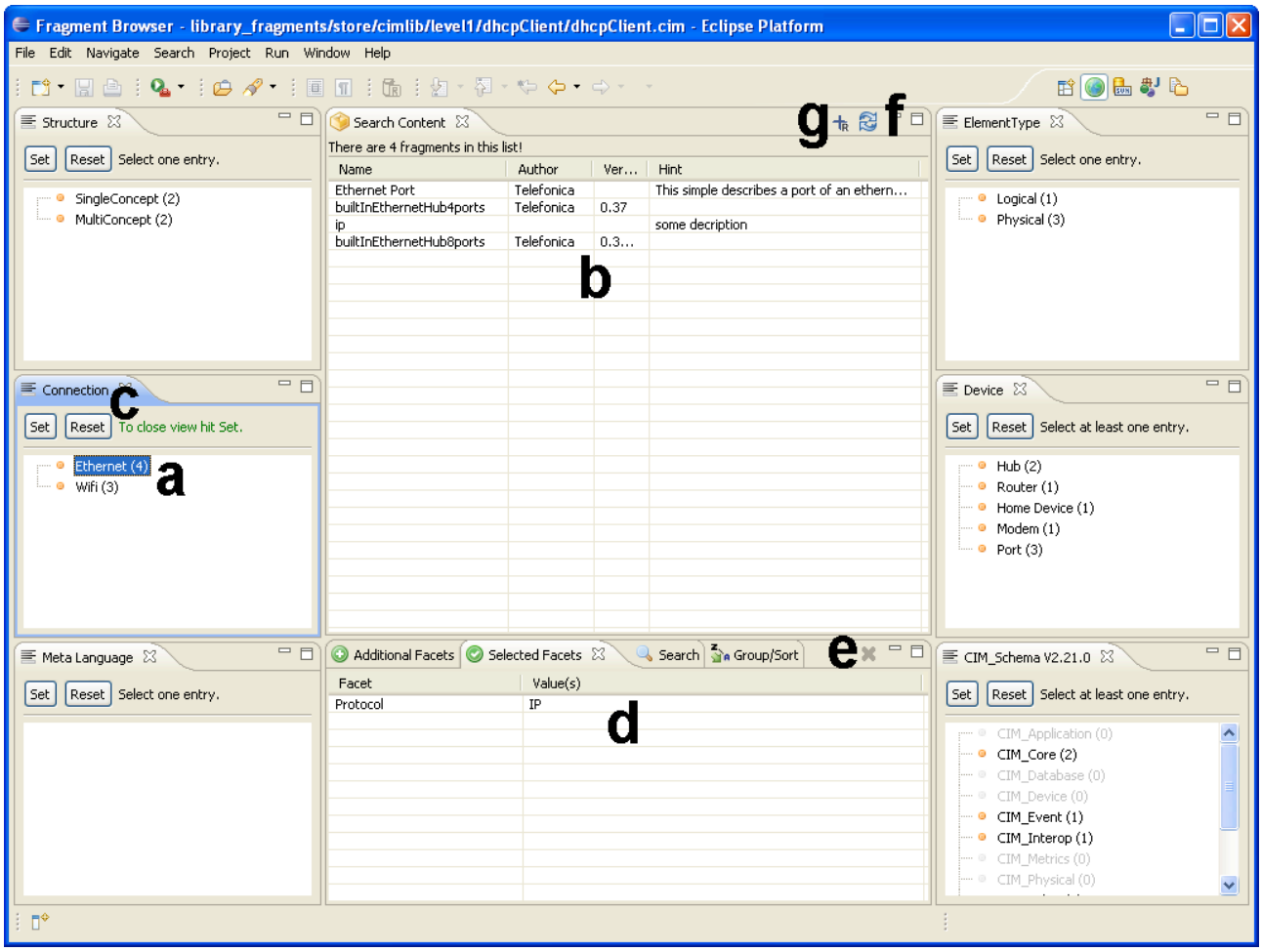

Figure 4 – Search process using the Fragment Browser

5.3. Fragment Reuse

Fragments that are the result of a search can directly be reused in a Reuseware composition program (cf. Figure 5). As the browser's Search Content view (a) can be opened in different perspectives it can be presented while editing a composition program (b). To reuse a fragment, simply select it in the Search Content view and hit the reuse button (c). The fragment will then appear in the composition program (d) and is ready to be connected with other fragments in the program.

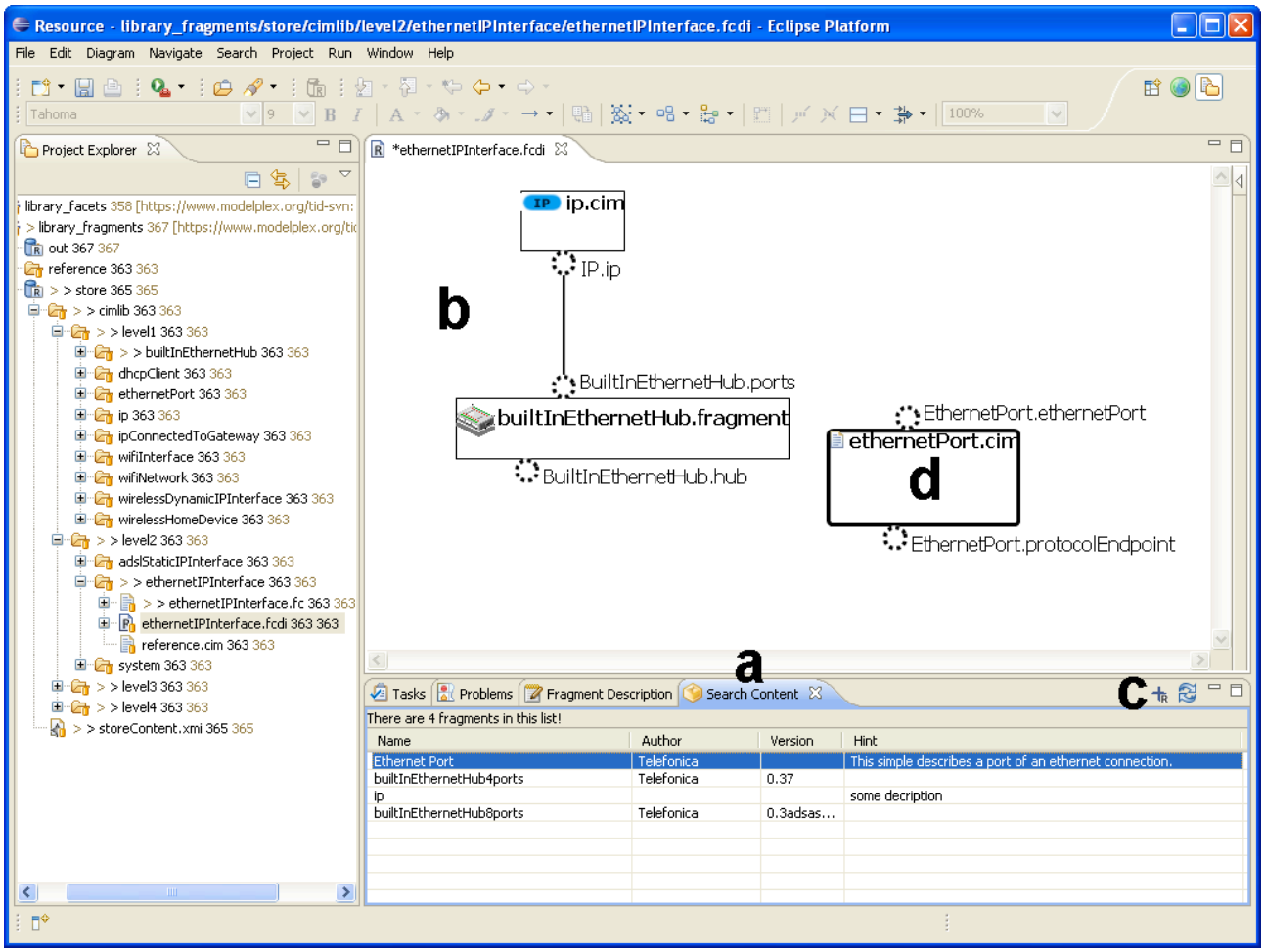

Figure 5 – Reusing fragments in a Reuseware Composition Program

6. Round-trip support

In this section we demonstrate the new round-trip feature of Reuseware on the TIS case study. We use the feature for viewpoint synchronization addressing parts of Requirement 179 [2]. Concretely, Requirement 179 states [2, p. 29]:

> "…several models will be built for a system, at different abstraction levels (e.g., logical vs. physical, PIM vs. PSM). The abstraction/decomposition support requested here addresses the support of the analysis/design or navigation process within one system model."

Furthermore, Requirement 179 requests [2, p. 29]:

> "- Need to manage impact on inter-model relationships
> - Need for tool mechanisms for editing, navigating, change mgt., etc."

In the TIS case study, the *system model* consists of different models, representing different viewpoints, defined in different modelling languages. Thus the requirement manifests in the goal to integrate different viewpoints on a single system and allow editing of each viewpoint separately without compromising the consistency of the overall system model. More precisely, four viewpoints are present in the TIS case study (i.e., the capability analysis model, the capability configuration models, the system services model and the security model). The first three viewpoints are specified in UML, which ensure some consistency based on the fact that all views use the same language. However, the last viewpoint is expressed in a DSL, namely the TRT Security DSL (which was internally developed by TRT and is available in the internal MODELPLEX repository). We use the round-trip feature to ensure consistency between the different viewpoints modeled in different languages.

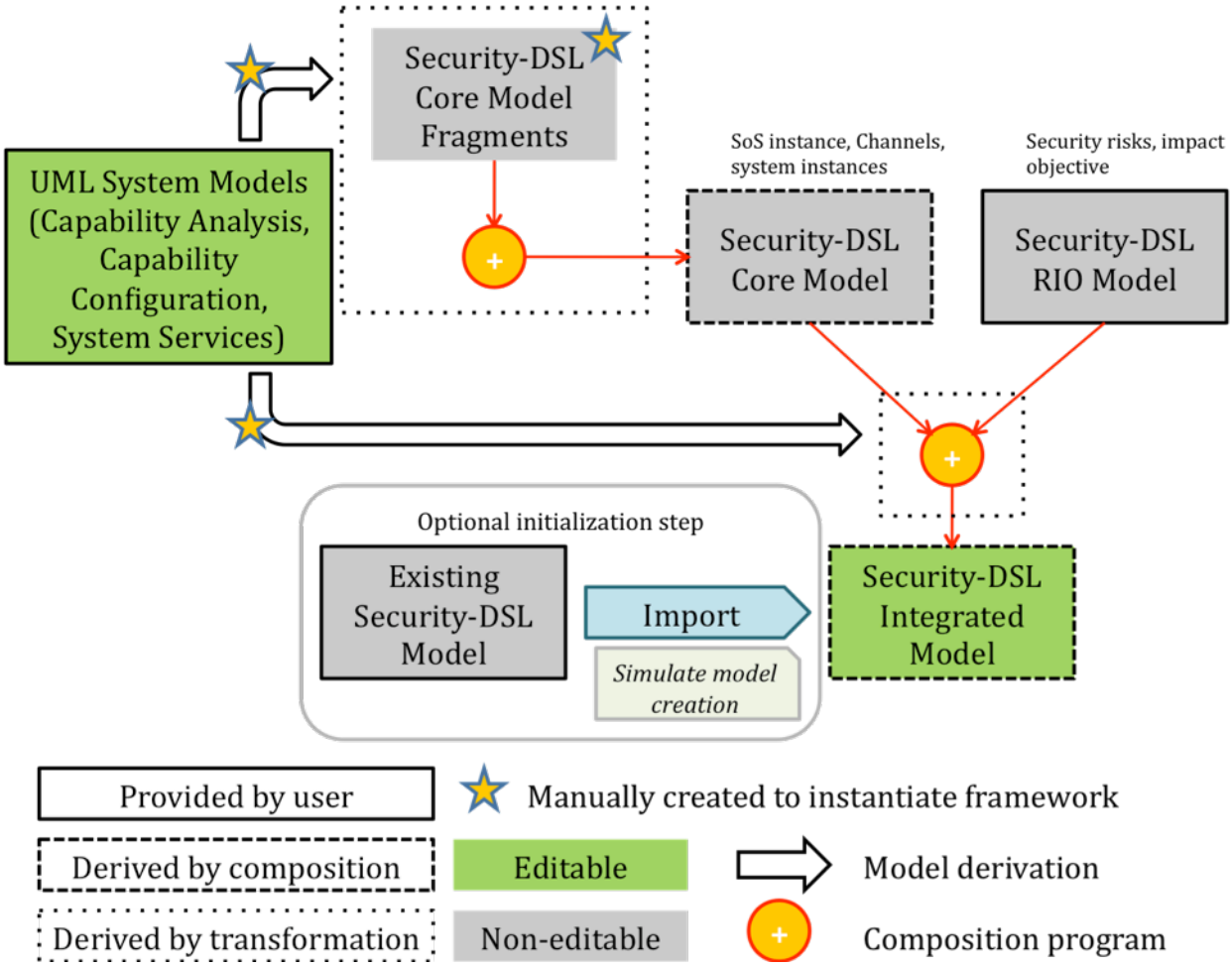

Figure 6 – Reuseware applied to the TIS case study

An overview of the round-trip scenario, which is addressed in the TIS case study, is shown in Figure 6. Starting from the UML system models (shown on the left) two composition programs are derived. The first uses Security DSL core model fragments to compose a Security DSL core model that represents the system core, which is modeled in UML, in the Security DSL (upper part). The second composition program enriches the Security DSL core model with security information (lower part). The transformations that perform these two derivation steps as well as the Security DSL core model fragments were manually created to instantiate Reuseware for the TIS scenario. That is, the transformations and Security DSL core model fragments are deployed such that Reuseware can work with them. This procedure needs to be taken only once.

The purpose of the core model fragments and the derived composition programs is to be able to compose the parts of the security models, which can be determined from the UML models and the parts, which are not present there. The latter are stored separately in a model called Security RIO[5] model. The derived composition program weaves the core model fragments to assemble the structure of the system, which can be extracted from the UML models and the additional security information from the RIO models to obtain an integrated security model (shown in the right part of Figure 6). An example of such an integrated security model is shown in Figure 7.

[5]Risk, Impact, Objective

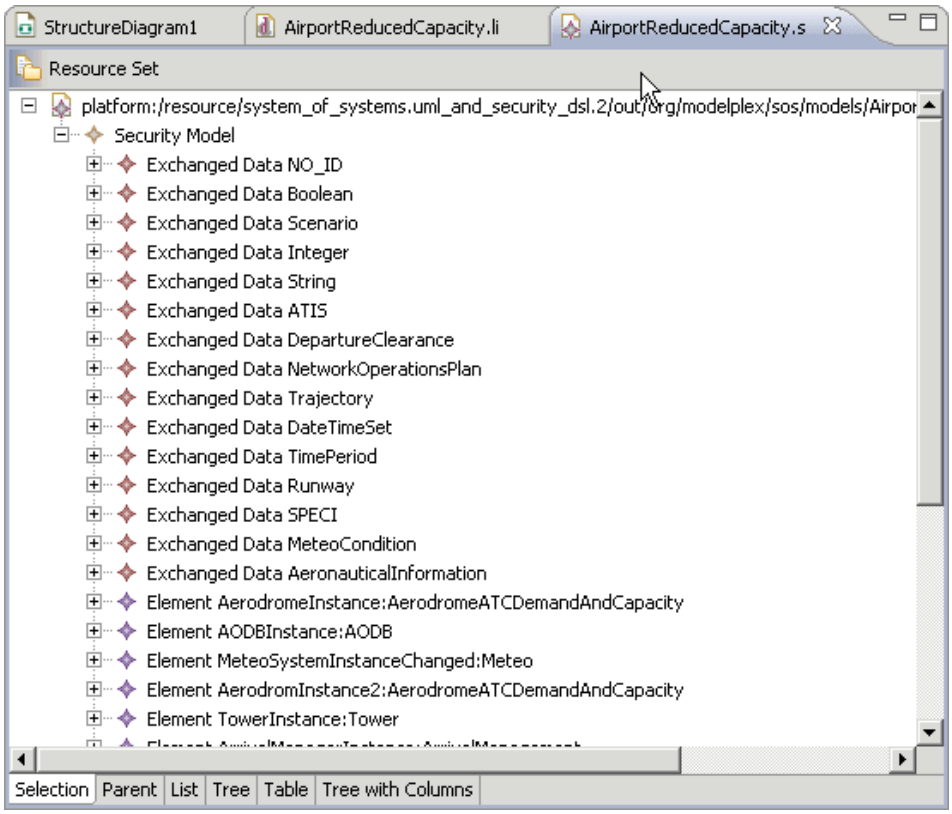

Figure 7 – Derived integrated security model

Having these transformations and composition available users can derive security models from the UML models and enrich them with additional information. As one of the goals of the TIS case study was to support editing of viewpoints, Reuseware was extended with support for round-trip engineering (more details are published in [7]). During model composition (i.e., while weaving models) additional trace information is captured by the Reuseware engine, which allows the editing of composed models by propagating changes to the respective source models. In the TIS case study, the integrated security model can therefore be changed and each change is automatically propagated to the RIO model. When the UML models are changed, the integrated view (i.e., the integrated security model) is recomposed preserving the contents of the RIO model. Thus, all viewpoints of a system (i.e., the UML and the security models) can be edited without compromising consistency.

7. Conclusion

This document presented the new features of the Reuseware model composition and weaving tool. With these features, additional concerns of Requirements 244 and 179 are supported. We will continuing the evaluation of the new features with the corresponding case studies and will present our final results in the M41 report of this deliverable.

Reuseware is now in a stable state to be further used and developed after the end of the MODELPLEX project. A lot of information about the tool is already available on www.reuseware.org. This site will again be updated at the end of the project in month 41 and 42. Furthermore, results around Reuseware were presented at various conferences and workshops throughout this year (cf. [8] for details).

8. References

[1] Heidenreich F., Henriksson, J., Johannes, J., Zschaler, S.: On Language-Independent Model Modularisation. In: Transactions on Aspect-Oriented Development VI, LNCS, vol. 5560, Springer (2009)

[2] MODELPLEX IP Deliverable D1.2.c: Updated Requirements Specification. (March 2008)

[3] R. Prieto-Díaz. A faceted approach to building ontologies. In Proc. IEEE International Conference on Information Reuse and Integration IRI 2003, pages 458–465, 2003.

[4] R. Prieto-Díaz. Implementing faceted classification for software reuse. Communications of the ACM, 34(5):88–97, 1991.

[5] MODELPLEX IP Deliverable D3.3.f: Telefónica DSL. (February 2008)

[6] Distributed Management Task Force Inc. (DMTF). Common Information Model Standards.

[7] Johannes, J., Samlaus, R., Seifert, M.: Round-trip Support for Invasive Software Composition Systems. In Proceedings of the "International Conference on Software Composition 2009", SC 2009, 2 - 3 July 2009, Zurich, Switzerland

[8] MODELPLEX IP Deliverable D6.6.2b: Newsletters Conferences Publications and Academic Lectures. (August 2009)