



**Trusted Computing Engineering
for Resource Constrained
Embedded Systems Applications**

D4.5

Requirements for Repository Access Tools

Project: TERESA
Project Number: IST-248410
Deliverable: D4.5
Title: Requirements for Repository
Access Tools
Version: V1.0
Confidentiality: Public
Author: Trialog
Date: 16 June 2011



Part of the Seventh Framework Programme
Funded by the EC - DG INFSO

Table of Contents

DOCUMENT HISTORY4

1 EXECUTIVE SUMMARY4

2 SCOPE OF THE TOOLS.....5

2.1 INTRODUCTION..... 5

2.2 IDENTIFICATION OF THE TARGET ACTIVITIES..... 5

2.3 DEFINITIONS 6

2.4 REQUIREMENT DESCRIPTION TEMPLATE 6

2.5 ORGANIZATION OF THE DOCUMENT 7

3 REQUIREMENTS FOR THE TOOLS7

3.1 GENERAL REQUIREMENTS 7

3.2 BROWSING VIEW BASED ACTIVITY REQUIREMENTS..... 9

3.2.1 *High Level Requirements* 9

3.2.2 *Design Requirements* 10

3.3 EDITING REQUIREMENTS..... 11

3.3.1 *High Level Requirements* 11

3.3.2 *Design Requirements* 12

4 ARCHITECTURE13

4.1 LINK WITH THE REPOSITORY 13

4.2 OVERVIEW OF THE MYARTEFACTS APPROACH 14

4.3 USING MYARTIFACTS PROCESS AND TOOLS TO DEVELOP THE ACCESS TOOLS 15

5 CONCLUSION.....16

6 BIBLIOGRAPHY16

List of Tables

Table 1 Template for the Requirements	7
Table 2 TR-01 Compliance with the repository structure.....	7
Table 3 TR-02 Open exchange approach	8
Table 4 TRB-HL-01 Role-based viewing of pattern	9
Table 5 TRB-HL-02 Consistency checking between views	9
Table 6 TRB-HL-03 API access to the repository	10
Table 7 TRB-D-01 Dynamic configuration rules	10
Table 8 TRB-D-02 consistency rules	10
Table 9 TRB-D-03 Non-Configured artefacts exhibition	10
Table 10 TRB-D-04 Dependency exhibition	11
Table 11 TRE-HL-01 User interface	11
Table 12 TRE-HL-02 API Edition of a pattern	11
Table 13 TRE-D-01 Edition of a new viewpoint.....	12
Table 14 TRE-D-02 Edition of configuration rules.....	12
Table 15 TRE-D-03 Edition of consistency rules.....	12
Table 16 TRE-D-04 Edition of non-configured artefacts	13
Table 17 TRE-D-05 Creation of a pattern from an existing one.....	13

Document History

Version	Status	Date	Description
V0.1	Draft	15/04/2011	C.G. (TRI): initial version with identification of requirements
V0.2	Draft	28/04/2011	G. M. (TRI): update the requirements
V0.3	Draft	06/04/2011	C.J. (TRI): update the architecture and the conclusion
V1.0	Final	02/06/2011	C.G (TRI): add definitions and take into account reviewer comments

Approval		
	Name	Date
Prepared	Antonio Kung	16 06 2011
Reviewed	All Project Partners	16 06 2011
Authorised	Antonio Kung	16 06 2011
Circulation		
Recipient	Date of submission	
Project partners	30 05 2011	
European Commission	16 06 2011	

1 Executive Summary

This document addresses the set of requirements involved in the interaction between the TERESA tools and the TERESA repository.

As a first step for the identification of the requirements, the document gives an overview of the vision of the TERESA project in order to take into account its specific aspects (RCES, S&D properties) and the strategy used (MDE, process sub-division) to handle them.

A description of each identified requirement is defined. These requirements are expressed from the viewpoint of the stakeholder which will access to the repository in order to create, browse and edit patterns.

The architecture of the access tools is based on the MyArtifacts' background work which is described in this document.

2 Scope of the Tools

This document represents the deliverable D4.5 titled “Requirements for Repository Access Tools”.

2.1 Introduction

These documents define a set of requirements for the tools that will be use along the TERESA processes. Five processes have been defined in [1] as follow:

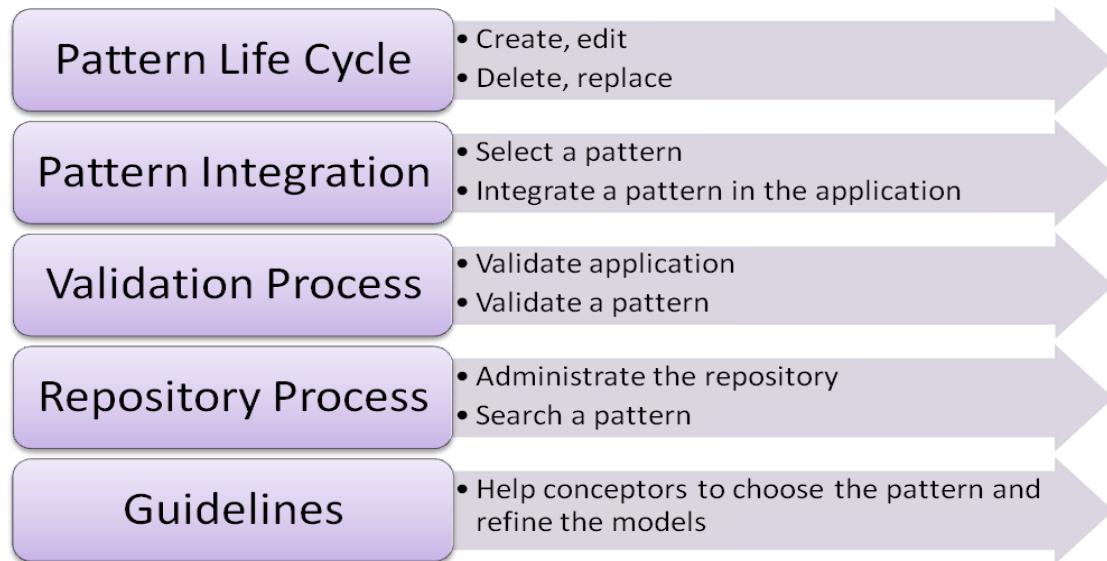


Figure 1: Main processes in a pattern-based application development

The **Pattern Life Cycle process** corresponds to all the tasks related to the pattern development (to create, to update, to delete, etc). At the end, the pattern is stored in a repository.

The goal of **Pattern Integration process** consists to plug a pattern in an application. For this purpose, in function of S&D properties, the security engineer can require all patterns stored in the repository which meet his needs. Then, through model transformations, the pattern is integrated in the application model.

The **Validation process** is transversal and interacts with the pattern life cycle and pattern integration processes. At each step of these processes, it is possible to validate a pattern or an application from an S&D point of view.

The **Repository process** manages the pattern repository and provides a pattern search mechanism.

At last, **Guidelines** are needed in order to help designers during the pattern life cycle and pattern integration processes.

2.2 Identification of the target activities

This deliverable addresses the repository access tools, some processes are obviously out of scope.

- Validation Process – This process is the main concern of WP5. Any possible tools linked to this process should be presented in the according deliverable.
- Guidelines – The guidelines are out of scope of this document. Nevertheless the guidelines must to incorporate the “good practice of tools” aspect.
- Repository Process – This process is not in the scope of the document. It is necessary to highlight that the tools are directly link with the repository. In this sense, the tools and the repository must have a sound development. In the rest of the deliverable, it will be assumed that the constituting elements (also called artefacts) of a pattern (e.g. parameters, interfaces or dynamics) can be flagged one or more time. A flag is dedicated to tag an element as “part of elements to be exhibit for the related view”. For instance, a security engineer view will show all the related flagged artefacts as the security properties provided by a pattern.

Unlike the three former the Pattern Lifecycle Process is our clear target processes. The tools connected to the repository will help various roles to interact with the repository (e.g., Security Engineer or Embedded System Engineer).

The Pattern Integration will use some tools connected to the repository to search a pattern according specific requirements (e.g. security properties, resource constraint or time constraint) but it is not the focus of this document. Nevertheless most of the general and browsing requirements defined in the next section can be applied for these tools too.

Therefore, the main focus of this document is to define the requirements for the tools that will be use for browsing and editing patterns. As TERESA is dedicated to support various roles in the development of a pattern it is necessary to keep specific views depending on the stakeholder (or role) using them.

2.3 Definitions

View: a view is specific representation of a pattern according to a concern.

Configuration: The definition of the kind of artefact that are exhibit to be shown in a view. An artefact configured or flagged for a specific view means that it will be presented by the view.

Consistency checking: The verification that two views are consistent with each other.

2.4 Requirement Description Template

In the following, a simple template for requirement is used based on (TERESA Partners, 2010) and presented in Table 1 Template for the Requirements

<i>ID</i>	<i>Name</i>
Description:	
Rational:	
Type:	Validation criteria:
Date:	Version:

Table 1 Template for the Requirements

2.5 Organization of the document

The rest of the document is divided into two main sections. Section 3 provides the identified requirement for the tools from different perspectives. Section 4 presents the proposed architecture for TERESA to cope with the identified requirements.

The last section is a conclusion leading to the next step of the Project

3 Requirements for the Tools

The objective of the document is to identify requirement for a tool which will browse pattern with a specific role and will enable editing a pattern. For this purpose, in this section, requirements are listed and are organized as follow:

1. General requirement. These requirements are available for the both activities (i.e., browsing and editing capabilities)
2. Browsing Activity with a role base concern
3. Editing Activity which enables to update a pattern

Each requirement will be described with the requirement template defined in Section 2.4.

3.1 General Requirements

<i>TR-01</i>	<i>Compliance with the Repository structure</i>
Description: The Tool, through various API (see later) must be compliant with the repository structure.	
Rational: The creation of specific role based view needs to be able to add specific information on the elements of a pattern. It enables the capabilities to flag some elements for a specific view.	
Type: General	Validation criteria: Inspection
Date: 18/04/2011	Version: 0.1

Table 2 TR-01 Compliance with the repository structure

<i>TR-02</i>	<i>Open Exchange Approach</i>
Description: The API must use Open Format to communication between the tool and the repository	
Rational: The data format between tools and the repository must be an open source type	

Type: General	Validation criteria: Inspection
Date: 18/04/2011	Version: 0.1

Table 3 TR-02 Open exchange approach

3.2 Browsing View Based Activity Requirements

3.2.1 High Level Requirements

TRB-HL-01	Role-based viewing of pattern
Description: The tool can show the relevant artefacts of a pattern for a given stakeholder.	
Rational: It allows to exhibit the relevant artefacts to edit or specialise a pattern from a specific point of view (e.g., S&D Engineer or RCES Engineer)	
Type: Browsing	Validation criteria: Inspection
Date: 18/04/2011	Version: 0.1

Table 4 TRB-HL-01 Role-based viewing of pattern

TRB-HL-02	Consistency checking between views
Description: The tool can verify and display consistency between two different views	
Rational: The tool must be able to check that the various views of a same pattern are consistent. This is an important requirement if the tool enables modification of artefacts from a specific view point.	
Type: Browsing	Validation criteria: Inspection
Date: 18/04/2011	Version: 0.1

Table 5 TRB-HL-02 Consistency checking between views

TRB-HL-03	API Access to the Repository
Description: The repository must provide an API in order to access and display all artefacts of a pattern	
Rational: The tool must be able to reach the repository to display the requested patterns and artefacts. In order to provide a smooth integration of tools with the repository, it is necessary to have a standardized API for browsing	
Type: Browsing	Validation criteria: Inspection
Date: 18/04/2011	Version: 0.1

Table 6 TRB-HL-03 API access to the repository

3.2.2 Design Requirements

<i>TRB-D-01</i>	<i>Dynamic Configuration Rules</i>
Description: The tool must include configuration rules to filter out the relevant artefacts for a given type of stakeholder	
Rational: It will enable a real capability to browse efficiently the pattern from a specific view point	
Type: Browsing	Validation criteria: Inspection
Date: 18/04/2011	Version: 0.1

Table 7 TRB-D-01 Dynamic configuration rules

<i>TRB-D-02</i>	<i>Consistency Rules</i>
Description: The tool must include rules for consistency checking between two views	
Rational: It enables the capability for the tools to provide to a specific user some guarantee about the consistency its view with respect to the whole pattern	
Type: Browsing	Validation criteria: Inspection
Date: 18/04/2011	Version: 0.1

Table 8 TRB-D-02 consistency rules

<i>TRB-D-03</i>	<i>Non-Configured Artefacts exhibition</i>
Description: The tool must include an option that exhibit all the non-flagged/non-configured artefacts for a specific view	
Rational: It enables the capability for the tools to provide to a specific user a view on the artefacts that are not configured to be shown in any view	
Type: Browsing	Validation criteria: Inspection
Date: 18/04/2011	Version: 0.1

Table 9 TRB-D-03 Non-Configured artefacts exhibition

TRB-D-04	Dependency exhibition
Description: The tool should provide an option to exhibit dependencies between artefacts.	
Rational: It enables the capability for the tools to provide users a better overview of the changes they make when editing a pattern.	
Priority: Optional	
Type: Browsing	Validation criteria: Inspection
Date: 29/04/2011	Version: 0.2

Table 10 TRB-D-04 Dependency exhibition

3.3 Editing Requirements

3.3.1 High Level Requirements

TRE-HL-01	User Interface
Description: The tool must provide a graphical interface to edit Pattern	
Rational: It enables the capability for the tools to provide to a specific user an easy to edit a pattern	
Type: Editing	Validation criteria: Inspection
Date: 18/04/2011	Version: 0.1

Table 11 TRE-HL-01 User interface

TRE-HL-02	API Edition of a Pattern
Description: The repository must provide an API in order to access and edit a pattern	
Rational: The tool must be able to reach the repository to edit the requested pattern. In order to provide a smooth integration of tools with the repository, it is necessary to have a standardized API for editing	
Type: Editing	Validation criteria: Inspection
Date: 18/04/2011	Version: 0.1

Table 12 TRE-HL-02 API Edition of a pattern

3.3.2 Design Requirements

<i>TRE-D-01</i>	<i>Edition of New Viewpoint</i>
Description: The tool must enable the creation of new viewpoint	
Rational: As each domain specific role can be involved in the creation and the edition of pattern, it is necessary to provide the capabilities to define new viewpoint	
Type: Editing	Validation criteria: Inspection
Date: 18/04/2011	Version: 0.1

Table 13 TRE-D-01 Edition of a new viewpoint

<i>TRE-D-02</i>	<i>Edition of Configuration Rules</i>
Description: The tool must enable the edition of configuration rules	
Rational: As the need of a specific role can evolved, it is necessary to offer the capability to edit the configuration rule for one viewpoint	
Type: Editing	Validation criteria: Inspection
Date: 18/04/2011	Version: 0.1

Table 14 TRE-D-02 Edition of configuration rules

<i>TRE-D-03</i>	<i>Edition of Consistency Rules</i>
Description: The tool must allow one to edit the consistency rules if necessary	
Rational: By editing or creating new viewpoint, it is necessary to enable the capability to enrich the set of rule to check consistency between various view	
Type: Editing	Validation criteria: Inspection
Date: 18/04/2011	Version: 0.1

Table 15 TRE-D-03 Edition of consistency rules

<i>TRE-D-04</i>	<i>Edition of non-Configured Artefacts</i>
Description: The tool, through the edition API, must allow the flagging of artefacts	
Rational: To enrich a view or to create a new one, it is necessary to select the type of artefact to be exhibit. In this way, the tool must allow one to flag some artefacts to define them as viewable	
Type: Editing	Validation criteria: Inspection
Date: 18/04/2011	Version: 0.1

Table 16 TRE-D-04 Edition of non-configured artefacts

<i>TRE-D-05</i>	<i>Creation a Pattern from an existing one</i>
Description: The tool, through the edition API, must allow the use of existing Patterns while editing another.	
Rational: This enables to reuse existing Patterns and thus allows to rely on already proven behaviour and eases the creation of Patterns	
Priority: Mandatory	
Type: Editing	Validation criteria: Inspection
Date: 02/05/2011	Version: 0.2

Table 17 TRE-D-05 Creation of a pattern from an existing one

4 Architecture

Currently, some tools exist for browsing patterns. Some of them are online such as [3] or [2]. These tools are more similar to pattern libraries and do not fit the TERESA requirements. Indeed, all of them offer browsing and editing capabilities. However, they do not use a role based view. For this purpose, we need a tool which filters the pattern properties according to the user role and uses its usual representation.

In this section, the links with the repository is summarized and how the future tool and the repository will interact. Then, the MyArtifacts approach is presented. MyArtifacts is both a methodology and a set of tools used by TRIALOG since 2010, and TRIALOG proposes to use this background information for the project.

4.1 Link with the repository

The repository contains all patterns and offers different capabilities such as (i) designing a pattern, (ii) updating a pattern, (iii) browsing the overall repository, (iv) searching a pattern according to properties. The tool identified in this document will interact with the repository. For this purpose, an API is needed

in order to obtain a pattern. Then, the tool will filter the elements according to the role needs. Currently, the API is not defined yet.

Finally, if a user updates a pattern through the tool, the new version must be uploaded to the repository.

The interactions with the repository are detailed in Section 4.3.

4.2 Overview of the MyArtefacts Approach

MyArtefacts is a process and a tool dedicated to the management of consistency between heterogeneous models. Trialog has developed this process and tools to cope with the following problems: Industries often use multiple tools during the process which are dedicated to one concern. Unfortunately, these tools are not compliant between them and needs manual transformations. For managing the complexity of current systems, many stakeholders step in and manipulate specific data with their own representation and annotation. MyArtefacts enables the data centralization with a consistency check. Figure 2 illustrates the process defined in the MyArtefacts approach. First of all, a MDE expert (i.e., with inputs from domain process expert) will model all the artefacts used by stakeholders and the relationships between them. He can also add constraints between two artefacts. Notes, that relationships and constraints can be applied between two artefacts from different domains.

In a second step, an instance of MyArtefacts is generated from the artefact model. This generated tool can provide several services as followings:

- Domain viewer. It is possible to represent different artefacts coming from different domain. The view is conformed to the usual representation of the stakeholder domain.
- Consistency check. When data are imported in the tool, consistency rules are executed in order to verify the consistency between all domain models.

The last step consists in using MyArtefacts by importing data and browsing the results.

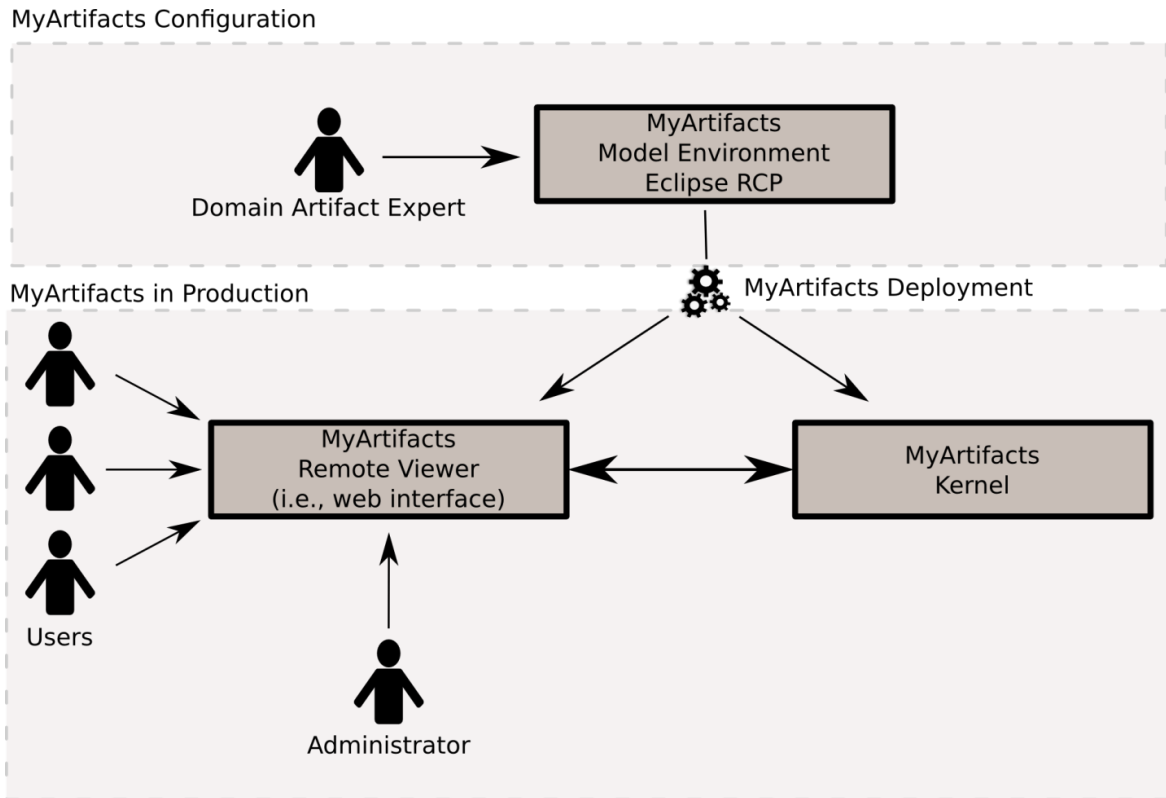


Figure 2 MyArtifacts process

4.3 Using MyArtifacts Process and Tools to develop the Access Tools

In the context of TERESA, MyArtifacts can be very helpful. Indeed, it is possible to provide two kinds of services:

1. Leveraging on the specific views capabilities of MyArtifacts to the repository. The repository stores all the patterns. If a user wants to edit or check a pattern, he will get all pattern information. MyArtifacts can provide a high level view which is dedicated to the user concern e.g. artefacts for the pattern designers (possibly domain specific), artefacts for the formal verifiers or artefacts for the domain specific integrators.
2. Consistency checking between views (including the three here above). A pattern will be constituted by orthogonal properties. A stakeholder is specialist of his domain and has no knowledge for the other concerns. For this purpose, it is possible to define consistency rules in MyArtifacts in order to help the checking.

For this purpose, MyArtifacts will provide a new layer of the repository in order to fit with the user requirements. The final architecture is shown in Figure 3.

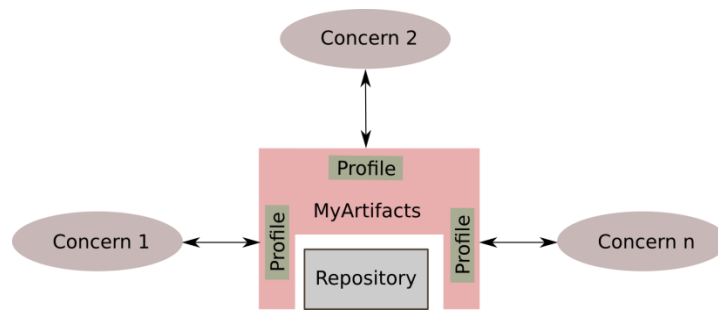


Figure 3 First version of the architecture

As depicted in Figure 3, MyArtifacts is around the repository in order to provide high level services. In order to define the specific views, some profiles are specified. A profile is constituted by the artefacts (i.e., pattern properties) manipulated in a concern and their representation.

5 Conclusion

This document represents the deliverable D4.5 which is dedicated to a repository access tool. The main need is to obtain a concern-based viewer for the final users. For this purpose, the use and the tuning the MyArtifacts approach is proposed. Profiles will customize the different views and can provide an important added value for customers. Indeed, this architecture is useful for different context:

- The repository is used in a company but it needs to be customized for each stakeholder.
- The repository is shared by several companies (i.e., an open source repository) but each supplier can tuned the information contained in a pattern to its specific requirements.

The tool will be developed in the task 4.5.2 and will be concluded by the deliverable D4.6 (i.e., a prototype).

6 Bibliography

- [1] TERESA Partners, "Identification of Requirements," TERESA Project, Public Report D2.2, D3.1 & D4.1, 2010.
- [2] Serenity Forum. <http://www.serenity-forum.com/patterns-library>
- [3] Portland Pattern Repository. <http://c2.com/ppr/index.html>