



# **Trusted Computing Engineering For Resource Constrained Embedded Systems Applications**

## **Identification of Requirements**

**D2.2 Requirements Application Viewpoint  
D3.1 Requirements Engineering Viewpoint  
D4.1 Requirements on Repository**

Project: TERESA  
Project Number: IST-248410  
Deliverable: D2.2, D3.1 and D4.1  
Title: Identification of Requirements  
for WP2, WP3 and WP4  
Version: v1.0  
Confidentiality: Public  
Author: All partners  
Date: 3 December 2010



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

# Table of Contents

<b>1</b>	<b>DOCUMENT HISTORY</b> .....	<b>5</b>
<b>2</b>	<b>EXECUTIVE SUMMARY</b> .....	<b>6</b>
<b>3</b>	<b>ACRONYMS</b> .....	<b>7</b>
<b>4</b>	<b>INTRODUCTION</b> .....	<b>8</b>
4.1	THE VISION OF THE TERESA PROJECT.....	8
4.2	THE TERESA CONCEPTUAL MODEL .....	10
4.2.1	<i>The Fault Propagation Model</i> .....	10
4.2.2	<i>The Trust Model</i> .....	11
4.2.3	<i>The S&amp;D Properties Model</i> .....	11
4.2.4	<i>The Pattern and Artefact Model</i> .....	12
4.3	ORGANIZATION OF THE DOCUMENT .....	13
<b>5</b>	<b>REQUIREMENTS ON PATTERN LIFECYCLE</b> .....	<b>14</b>
5.1	CONCEPTUAL LEVEL .....	14
5.1.1	<i>Artefacts</i> .....	14
5.1.1.1	Unified S&D Artefact Ontology.....	14
5.1.1.2	Unified S&D Integration Requirements Artefact Ontology.....	14
5.1.1.3	Unified S&D Formal Validation Artefact Ontology .....	15
5.1.1.4	Pattern Integration Constraints .....	15
5.1.1.5	Artefacts for Internal Structure Description .....	16
5.1.1.6	Artefacts for Interfaces.....	16
5.1.1.7	Artefacts for Attributes.....	17
5.1.1.8	Variability of Artefact .....	17
5.1.1.9	Extendibility of Existing Artefacts .....	18
5.1.1.10	Extendibility of the Set of Artefacts .....	18
5.1.2	<i>Models</i> .....	19
5.1.2.1	Role based Specification (Pattern S&D Expert, Formal Validation, Domain Integrator).....	19
5.1.2.2	Top-down Multi-level Specification (Domain Independent, Domain Dependent Platform Independent, Domain Dependent Platform Specific) .....	19
5.1.2.3	Role based and Level based Pattern Representation .....	20
5.1.2.4	Portability .....	20
5.1.2.5	Reusability .....	21
5.1.2.6	Availability of Needed Artefacts Per Role and Per Level .....	21
5.1.2.7	Detailed Description of Patterns Internal Behaviour.....	22
5.1.3	<i>Transformations</i> .....	22
5.1.3.1	Traceability of Artefacts During the Pattern Life Cycle.....	22
5.1.3.2	Traceability of Properties Mechanism .....	23
5.1.3.3	Consistency between Pattern Levels .....	23
5.1.3.4	Guidelines for Easy Pattern Definition .....	24
5.2	MODELLING LEVEL .....	24
5.2.1	<i>Editable Artefacts</i> .....	24
5.2.1.1	Pattern Properties .....	24
5.2.1.2	Expressiveness of S&D Property Definition .....	25
5.2.1.3	Pattern Integration Constraints/Recommendations.....	25
5.2.1.4	Validation Constraints .....	26
5.2.1.5	Repository Search Keys.....	26
5.2.1.6	Adaptable Pattern Description According to Role .....	27
5.2.1.7	Adaptable Pattern Description According to Level .....	27
5.2.1.8	Adaptable Pattern Internal Structure According to Role.....	28

5.2.1.9	Adaptable Pattern Internal Structure According to Level.....	28
5.2.1.10	Editable Domain Parameters .....	29
5.2.1.11	Pattern Domain Dependent Information.....	29
5.2.2	<i>Editable Models</i> .....	30
5.2.2.1	Pattern Search Mechanism (Level, Role).....	30
5.2.2.2	Description of Pattern Functionality .....	30
5.2.2.3	Pattern Description .....	31
5.2.2.4	Establish Dependency between Level Specific Views of Model.....	31
5.2.2.5	Pattern Characterization .....	32
5.2.2.6	Combining Patterns .....	32
5.2.2.7	Identification of Pattern Interfaces.....	33
5.2.2.8	Substitute Pattern .....	33
5.2.3	<i>Editable Transformations</i> .....	34
5.2.3.1	Transformation Rules.....	34
5.2.3.2	Assistance / Guidelines for Easy Pattern Integration .....	34
5.2.3.3	Assurance of Pattern Dependency Consistency .....	35
5.2.3.4	Traceability of Properties during Pattern Life Cycle .....	35
5.2.3.5	Repository - Pattern Validation .....	36
<b>6</b>	<b>REQUIREMENTS ON INTEGRATION PROCESS TRANSFORMATIONS.....</b>	<b>37</b>
6.1	DOMAIN INDEPENDENT PATTERN INTEGRATION.....	37
6.1.1	<i>Conceptual level</i> .....	37
6.1.1.1	Pattern Selection / Discrimination .....	37
6.1.1.2	Integration – Pattern Validation.....	37
6.1.1.3	Traceability of S&D Requirements During Pattern Integration .....	38
6.1.2	<i>Modelling</i> .....	38
6.1.2.1	Specifying S&D Integration Requirement.....	38
6.1.2.2	Determination of the User's System Model .....	39
6.1.3	<i>Transformation</i> .....	40
6.1.3.1	Guidelines for Easy Pattern Integration (Domain Independent Actions) .....	40
6.1.3.2	Match the Interfaces between the Pattern and the Application .....	40
6.1.3.3	Consistency between Domain Independent and Domain Specific Models .....	41
6.2	DOMAIN SPECIFIC PATTERN INTEGRATION.....	41
6.2.1	<i>Editable Conceptual Level</i> .....	41
6.2.1.1	Project's Organizational Model .....	41
6.2.1.2	Specialization of Artefact Instances.....	42
6.2.2	<i>Editable Modelling</i> .....	42
6.2.2.1	Using Coding Standards .....	42
6.2.2.2	Adaptation of Pattern to Resource Constraints .....	43
6.2.3	<i>Editable Transformations</i> .....	43
6.2.3.1	Guidelines for Easy Pattern Integration (Domain Specific Actions).....	43
6.2.3.2	Adaptation of Pattern to Domain Specific Constraints .....	44
6.2.3.3	Certification.....	44
6.2.3.4	Integration of the Formal Validation .....	45
<b>7</b>	<b>REQUIREMENTS ON REPOSITORY STRUCTURE.....</b>	<b>46</b>
7.1	CONCEPTUAL LEVEL .....	46
7.1.1	<i>Artefacts</i> .....	46
7.1.1.1	Domain Independent Repository Construction.....	46
7.1.1.2	Domain Specific Repository Construction .....	46
7.1.1.3	Domain Independent Repository Organization .....	47
7.1.1.4	Domain Specific Repository Organization.....	47
7.1.1.5	Domain Independent Pattern Retrieval .....	48
7.1.1.6	Domain Specific Patterns Retrieval.....	48
7.1.1.7	Visualisation of a Pattern .....	49
7.1.1.8	Role Dependent Visualisation of the Repository .....	49
7.1.2	<i>Modelling</i> .....	50
7.1.2.1	Guidelines on Interactions with the Repository .....	50
7.1.2.2	Multiview Support for Domain and Actor Specific Adaptation .....	50
7.1.2.3	Repository Compliant Description of the Pattern .....	51
7.1.2.4	Domain and Role Based Pattern Searching.....	51
7.1.2.5	Support Dependencies between Patterns.....	52

7.1.2.6	Guideline for Defining Pattern Dependencies .....	52
7.1.2.7	Support the Visualisation of the Pattern Dependencies .....	53
7.1.3	<i>Transformation</i> .....	53
7.1.3.1	Adaptation of the visualisation according to the user role .....	53
7.1.3.2	Data Structure to Implement the Repository .....	54
7.2	<b>MODELLING LEVEL</b> .....	54
7.2.1	<i>Editable Artefacts</i> .....	54
7.2.1.1	Edit Repository Search Keys .....	54
7.2.1.2	Edit Visualisation of a Pattern .....	55
7.2.1.3	Edit Visualisation of a Pattern Dependencies .....	55
7.2.1.4	Edit User Role View .....	56
7.2.2	<i>Editable Models</i> .....	56
7.2.2.1	Support Pattern Search Function According to Level .....	56
7.2.2.2	Support Pattern Search Function According to Role.....	57
7.2.2.3	Edit the Functionality Relating the Visualisation of a Pattern .....	57
7.2.2.4	Edit the Functionality Relating the Visualisation of the Pattern Dependencies .....	58
7.2.3	<i>Editable Transformations</i> .....	58
7.2.3.1	Edit the Data Structure to Implement the Repository .....	58
<b>8</b>	<b>CONCLUSION</b> .....	<b>59</b>
<b>9</b>	<b>BIBLIOGRAPHY</b> .....	<b>60</b>

# 1 Document History

Version	Status	Date
v0.1	draft	28/04/2010
V0.2	Adding the vision of Teresa and the conceptual model	21/09/2010
V0.3	Restructuring requirements	22/09/2010
V0.4	Adding the vision of Teresa	24/09/2010
V0.5	Modifying requirements during the trilateral meeting	27/09/10
V0.6	Reorganising and correcting requirements	15/10/10
V0.7	Reorganising the requirements	02/11/10
V1.0	Taking into account comments from the cross-checking	12/11/10

Approval		
	Name	Date
Prepared	All Project Partners	02 11 2010
Reviewed	Andreas Fuchs, Annika Paus	08 11 2010
Authorised	Antonio Kung	03 12 2010
Circulation		
Recipient	Date of submission	
Project partners	03 12 2010	
European Commission	03 12 2010	

## 2 Executive Summary

This document addresses the set of requirements involved in the TERESA process and represents the deliverables D2.2, D3.1 & D4.1. 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 show the restrictions to be faced (RCES, S&D properties) and the strategies to follow (MDE, process sub-division) to handle this issues. Moreover, a brief overview is provided of the TERESA conceptual model to illustrate the context and to have a common understanding of the concepts used in this deliverable.

Once the context of the TERESA process is described, a description of each identified requirement is defined. These requirements are mainly focused on the viewpoint of engineering process definition in order to develop pattern-based applications (define, integrate, transform, and validate a pattern in an application). In addition, the document identifies ways in which the engineering process may serve the engineer in the implementation of this process, and finally, it identifies the restrictions that have to be considered in the definition of the repository.

For clarity, the requirements have been structured in three main sub-processes that are directly related with the main actions the engineering process supports, which are *Pattern Life Cycle process*, *Integration Process Transformation* and *Repository Structure*.

### 3 Acronyms

Acronym	Meaning
HW	Hardware
MDE	Model Driven Engineering
PIM	Platform Independent Model
PSM	Platform Specific Model
RCES	Resource Constrained Embedded Systems
SW	Software

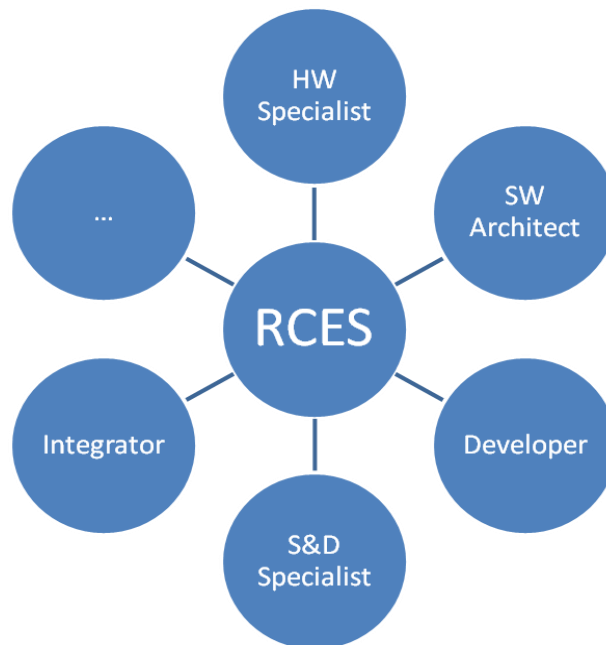
## 4 Introduction

The main purpose of this document is to identify requirements related to TERESA concerns. This document corresponds to the deliverables D2.2 Requirements Application Viewpoint, D3.1 Requirements Engineering Viewpoint and D4.1 Requirements on Repository. Section 4.1 gives a detailed description of the vision of TERESA, Section 4.2 introduces the TERESA conceptual model. Section 4.3 deals with the organization of the document and highlights why all deliverables are merged in one document.

### 4.1 The Vision of the TERESA Project

Embedded systems correspond to information processing systems integrating HW and SW embedded into enclosing products to fulfil a specific function. This kind of system is referred to RCES (Resource Constrained Embedded Systems) if they have memory, autonomy, and/or computation processing restrictions. The industry is friendly to manage RCES with time and computation constraints. Although the issue is complex, solutions exist.

Currently, security and dependability (S&D) becomes a strong requirement. In many industries, this need is reflected in changing their processes to integrate this new business. As shown in Figure 1, during the RCES development, many engineering roles have to be considered. A clear separation of these roles is required while ensuring the coherence and the consistency between them.



**Figure 1: Synergy of all stakeholders**

Model Driven Engineering (MDE) can provide some solutions for this issue. A model provides an abstraction of specific views of the system. For each concern, a specific view can be developed. For RCES, some MDE-based initiatives already exist like the MARTE UML profile [1].

In MDE, as depicted in Figure 2, the global process distinguishes two levels:

- PIM – Platform Independent Model. A PIM is a model that is independent of any platform (i.e., at the HW & SW levels) and doesn't contain information on technologies that will be used to deploy the application. At the start, this model is independent of the domain. Then, during the refinement of the model, domain specific concepts are introduced.
- PSM – Platform Specific Model. This model shows the details of the platform. The PSM is essentially the basis for executable code loaded on the platform. There are several levels of PSM. The first is the result of the transformation of a PIM. It may be represented using UML diagrams related to a specific platform. Other PSMs are obtained using successive transformations until the code generation in a specific language (i.e., C, Java, C #, etc). In this case, a PSM contains information such as program code, related libraries and descriptors of deployment.



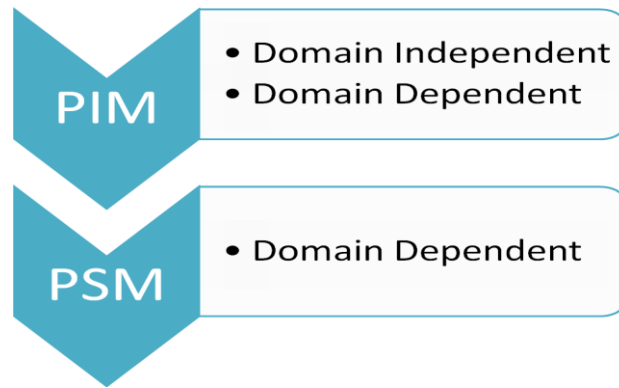


Figure 2: OMG MDE-based product process

S&D requirements can be fulfilled by several solutions. These solutions are often reused in several projects and can be represented by a pattern. A pattern is like a template which responds to a specific issue. Patterns can be stored into a repository and then, they can be reused at the development time. For this purpose, it is necessary to search in the repository a pattern which meets the application requirements. By consequence, the use of patterns can ease the integration of one of these solutions due to reusability. However, this technique has an impact on the engineering process which needs to be completely redesigned to efficiently support the pattern paradigm.

In TERESA, we focus on pattern-based application development. As depicted in Figure 3, several kinds of processes are needed. Note that the product process (i.e., application development process) is ignored in Figure 3.

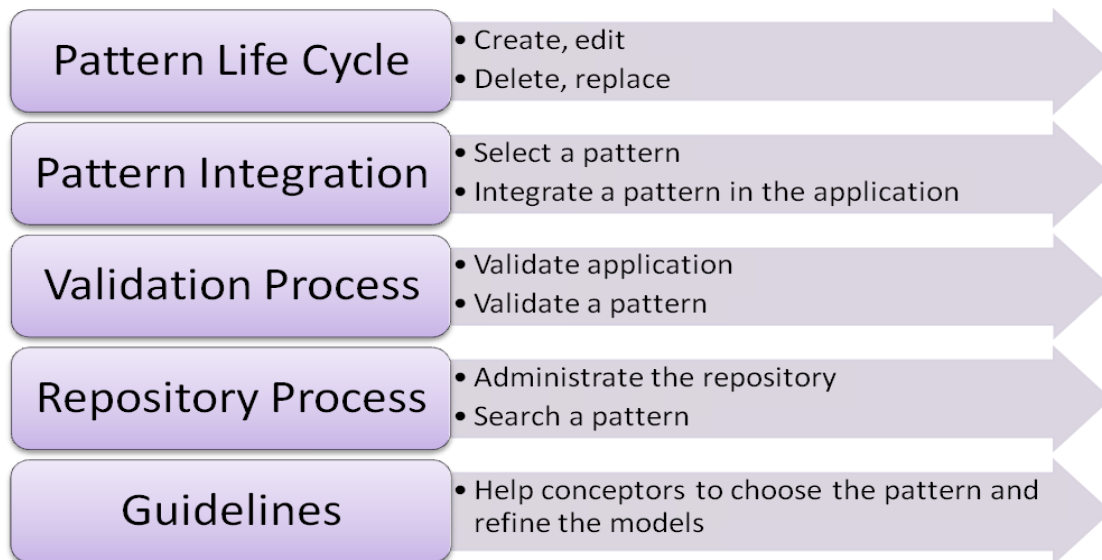


Figure 3: Main processes in a pattern-based application development

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.

As depicted in Figure 4, the TERESA vision consists in application development by pattern construction. The TERESA vision is based on two main processes centralised around a repository. The repository provides S&D patterns which can be used during the application development. In order to design patterns, some guidelines help the security specialists. At the end of the pattern development, formal validation guarantees that the pattern achieves the required S&D properties.

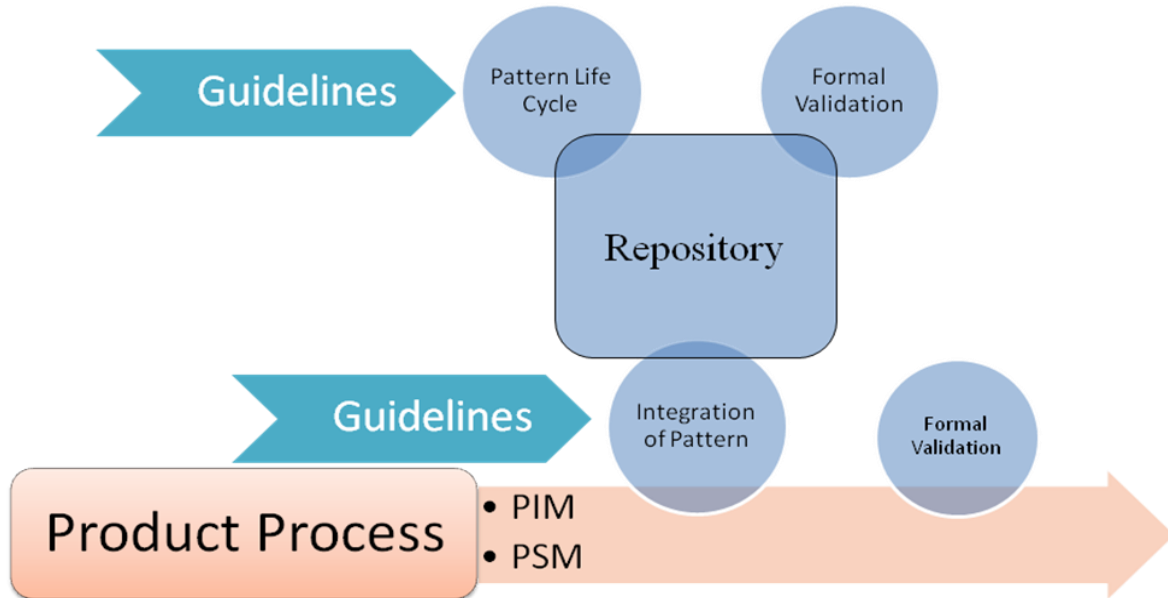


Figure 4: The TERESA Vision

Currently, the state of the art covers some parts of this global process for specific issues. However, there is no solution which provides an S&D pattern repository, guidelines and uses model driven engineering.

## 4.2 The TERESA Conceptual Model

In this section, we present the TERESA conceptual model. The goal of this model is to illustrate the addressed problems and to have a common understanding of all the concepts used in this deliverable. In the following text, all concepts are typeset in bold letters at the moment of their definition. The TERESA conceptual model is composed in four models presented in the following sections. The goal of this conceptual model is to define the different notions used in TERESA.

### 4.2.1 The Fault Propagation Model

A system is a combination of interacting elements organized to achieve one more stated purposes. [2] In TERESA, we make use of this model to comprehend all the factors that may lead to an error of an embedded system and hence to determine the trust level of a system. Here, we mainly focus on RCES. Basically, the trust level of a system decreases if a fault could damage the system. Typically, a fault can have a direct influence on S&D. Generally, the AVI (Attack, Vulnerability, and Intrusion) fault model is used to demonstrate the *fault* → *error* → *failure* paradigm (see the Figure 5) [3]. Figure 5 explains how a fault could be propagated into the system and how it could lead to a failure.

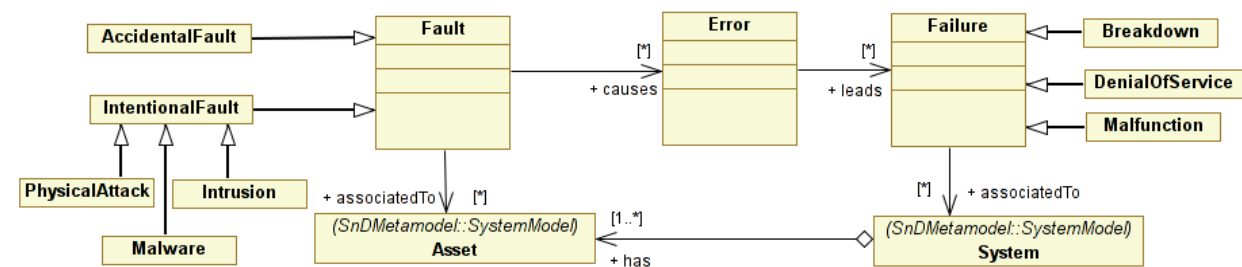


Figure 5: Fault Propagation Model

As shown in Figure 5, a **fault** is the presumed or hypothesized cause of an **error**. Faults may be classified according to several criteria. A fault is related to an asset of the system. An asset could be hardware, software, people, etc... Here, we classify faults as **accidental** or **intentional**. Accidental faults can arise during either the system design and development process or during operation (through a violation of an operating or maintenance procedure).

Intentional faults fall into three classes: **malware**, **physical attacks** and **intrusions**. Typically an attack exploits a fault or vulnerability of an **asset** to make an intrusion. An attack may use physical means to cause faults, such as power fluctuations, radiation, or wire-tapping. [4]

An error is detected if its presence in the system triggers an error message or error signal that originates from within the system. The possibility to detect errors contributes to improved security and trust in the system. Errors that are present but not detected are called latent errors. In [5], Avizenis defines an error as the part of the system state that may lead to a **failure**.

A failure leads to the inability of the system or some of its parts to meet their specifications (functional and non-functional requirements). Possible failures could be: **breakdown**, **denial of service** and **malfunction**.

### 4.2.2 The Trust Model

In order to limit the probability of occurrence as well as the impact of a fault, TERESA uses a Trust Model, depicted in Figure 6, to visualize the dependencies of a RCES's trust that is built on a number of factors.

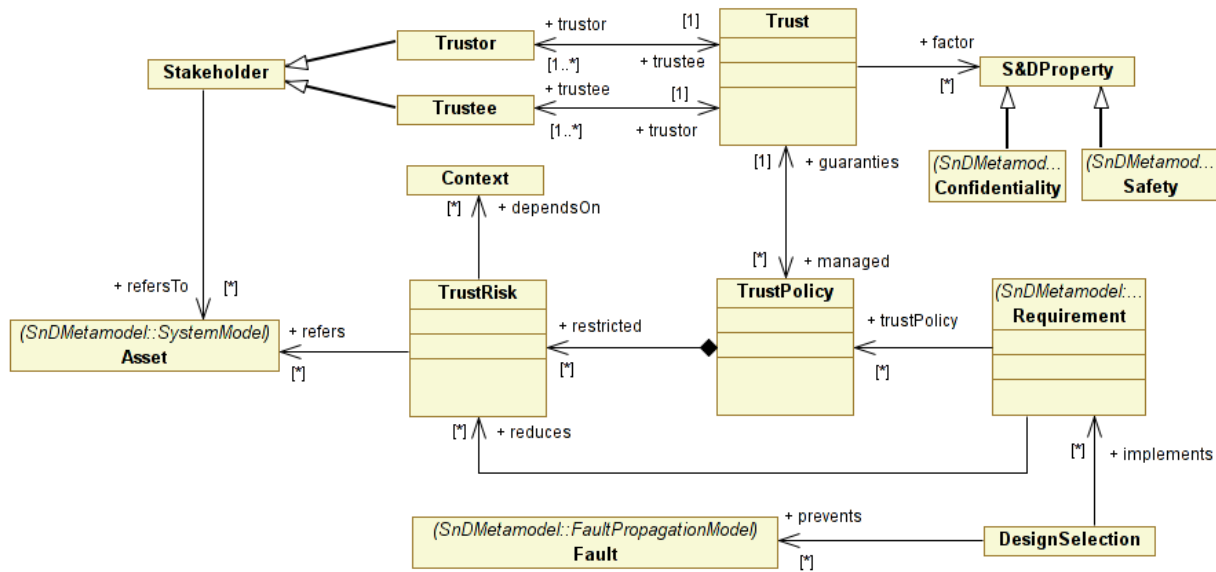


Figure 6: Trust Model

As depicted in Figure 6, a **trustor** can trust a **trustee** if, and only if, at least one factor of S&D is guaranteed (these factors are called **S&D Property** in Figure 6). For this reason, during the development process, requirement identification is a main issue. [6] In order to fulfil the **requirements**, it is possible to define **policies** which enforce the trust level. Then, during the system implementation, **design selection** will implement the requirements and reduces the **risk of fault**.

### 4.2.3 The S&D Properties Model

While the Trust Model in Figure 6 figures out that trust is related to several factors or dimensions [7], the model in Figure 7 summarizes possible criteria in S&D. It shows, for instance, that a user could trust a system because it is confidentiality-aware and will protect his personal data. Of course, it is needed that design selections implement mechanisms which ensure this quality of service (i.e., access control, anonymity, non-repudiation, etc).

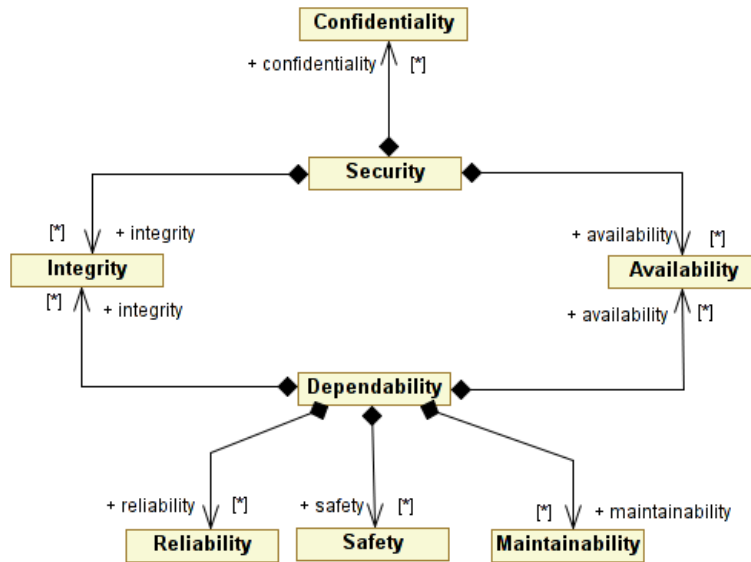


Figure 7: S&D Properties

#### 4.2.4 The Pattern and Artefact Model

In the TERESA project, different application sectors are considered. Each sector has its own needs and constraints for the development of its applications. In TERESA we adapt the application through the use of patterns in order to handle these needs and constraints. As shown in Figure 8, applications can be composed of different patterns.

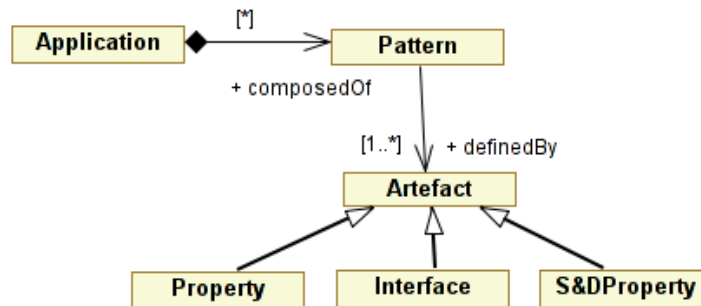
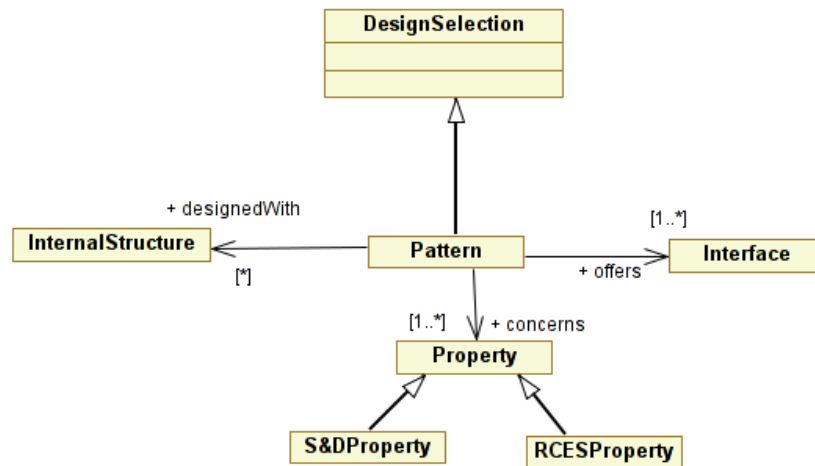


Figure 8: Pattern & Artefact definition

At the same time, the patterns need a set of artefacts such as interfaces, attributes, internal structure, and others that support the description of the features/qualities that the pattern provides.

An artefact is a substance or structure not naturally present in the matter being observed but formed by artificial means. [8] In computer science, an artefact is an artificial product or effect observed in a natural system, especially one introduced by the technology used in scientific investigation. It allows creating something to represent every thing as a nonexistent entity in the field. That is, an artefact is something viewed as a product of human conception rather than an inherent element. For instance, a relevant artefact for embedded systems is time or resource constraint.



**Figure 9: Pattern Structure**

As shown in Figure 9, the pattern must address the features of the design selection through the adaptation of design features or by means of including extra features in order to take into account the concern properties (S&D properties, RCES properties) in the design.

Furthermore, to support the definition of the pattern and its integration in the design, the pattern is provided with a description of its internal structure and interfaces offered to communicate with the rest of the system.

### 4.3 Organization of the Document

The work plan of the TERESA project consists of several technical work packages (WP). For all of these WPs, a task consists of specifying the requirements. In the following text, a summary of this task is given:

- T2.2: the objective of this task is to identify the requirements of the trusted engineering process, from the engineer viewpoint. These requirements will be worked out in parallel with the requirements of WP3, which identify requirements from an engineering process viewpoint. Examples of domain specific requirements are standards and assurance approaches.
- T3.1: the objective of this task is to identify the requirements from an engineering process definition viewpoint. While the requirements from WP2 take a black box approach (i.e. what the engineering process should do to serve the engineer), the requirements in WP3 take an implementation approach (i.e. how the engineering process should be structured and constructed to serve WP2). For the definition of the requirements at the process level, it is important to take the domain specific requirements (e.g. existing standards for engineering) into account here.
- T4.1: the objective of this task is to specify and develop the repository that will be used to store the meta-models (engineering and trust meta-models), models (engineering process model, trust models, patterns models). The requirements for such repository will first be identified (e.g. administration, interoperability with other tools, format...)

For each task, a deliverable describes the respective requirements. Finally, the TERESA partners decided to write one document for all of them. The advantages of this process are to avoid redundant requirements and to allow the partners being involved in the requirement identification. However, the organization of this document highlights the work in each task.

A common organization of the requirements, presented in Section 5, 6, and 7, is used for each identified process. The first section describes the requirements at a conceptual level (relevant to T3.1 or T4.1 with respect to the target process). Then, a section specifies requirements relevant for the final user (i.e., T2.2).

## 5 Requirements on Pattern Lifecycle

In this section, we will identify and detail all requirements related to the pattern life cycle. This section is divided in two subsections. The first one, called “Conceptual Level”, is relevant for the task 4.1 and represents a part of the deliverable D4.1. The second one, titled “Modelling Level” corresponds to the deliverable D2.2.

### 5.1 Conceptual Level

#### 5.1.1 Artefacts

##### 5.1.1.1 Unified S&D Artefact Ontology

<b>REQ-PL-001</b>		<b>Unified S&amp;D artefact ontology</b>
<b>Description:</b> In the process of pattern definition, a unified representation of the concepts to be used in the identification and definition of the S&D artefacts must be established.		
<b>Source:</b> General		
<b>Reference process:</b> Pattern Life cycle		
<b>Rational:</b> To establish a common understanding and also to identify different kind of artefacts that can be used for the definition of the pattern.		
<b>Type:</b> <i>Mandatory</i>	<b>Validation criteria:</b> <i>Inspection</i>	<b>WP Relevance:</b> 4
<b>Date:</b> 23/08/10		<b>Version:</b> 1.0

##### 5.1.1.2 Unified S&D Integration Requirements Artefact Ontology

<b>REQ-PL-002</b>		<b>Unified S&amp;D integration requirements ontology</b>
<b>Description:</b> In the process of pattern definition, a unified formal representation of the concepts to be used in the identification and definition of the S&D integration requirements must be established.		
<b>Source:</b> General		
<b>Reference process:</b> Pattern Life cycle		
<b>Rational:</b> To establish a common understanding and also to identify different kind of objects and concepts that can be used for the definition of the S&D integration requirements.		
<b>Type:</b> <i>Mandatory</i>	<b>Validation criteria:</b> <i>Inspection</i>	<b>WP Relevance:</b> 4
<b>Date:</b> 07/06/10		<b>Version:</b> 1.0

### 5.1.1.3 Unified S&D Formal Validation Artefact Ontology

<b>REQ-PL-003</b>		<b>Unified S&amp;D formal validation artefact ontology</b>	
<b>Description:</b> In the process of pattern description, a unified formal representation of the concepts to be used in the identification and definition of the S&D formal validation artefacts must be established.			
<b>Source:</b> General			
<b>Reference process:</b> Pattern Life cycle			
<b>Rational:</b> To establish a common understanding and also to identify different kind of concepts that can be used for the definition of the S&D formal validation artefacts.			
<b>Type:</b> <i>Mandatory</i>	<b>Validation criteria:</b> <i>Inspection</i>	<b>WP Relevance:</b> 4	
<b>Date:</b> 27/09/10		<b>Version:</b> 1.0	

### 5.1.1.4 Pattern Integration Constraints

<b>REQ-PL-004</b>		<b>Pattern integration constraints/recommendations</b>	
<b>Description:</b> In the application development, some recommendations or constraints have to be taken into account before integrating the pattern in the application. The developer needs this information available in the pattern description. This information that the pattern may require (i.e., resources, protocols, architecture, implementation techniques, etc) must be specified.			
<b>Source:</b> General			
<b>Reference process:</b> Pattern Lifecycle			
<b>Rational:</b> To support the developer in the integration of the pattern			
<b>Type:</b> <i>Mandatory</i>	<b>Validation criteria:</b> <i>Inspection</i>	<b>WP Relevance:</b> 4	
<b>Date:</b> 31/08/10		<b>Version:</b> 1.0	

### 5.1.1.5 Artefacts for Internal Structure Description

<b>REQ-PL-005</b>		<b>Artefacts for internal structure description</b>	
<p><b>Description:</b> In the process of pattern description, the internal structure of the pattern has to be defined. For this purpose, artefacts which are needed to describe the internal structure must be available.</p>			
<p><b>Source:</b> General</p>			
<p><b>Reference process:</b> Pattern Life cycle</p>			
<p><b>Rational:</b> To support the description of the pattern's internal structure.</p>			
<p><b>Type:</b> <i>Mandatory</i></p>		<p><b>Validation criteria:</b> <i>Inspection</i></p>	<p><b>WP Relevance:</b> 4</p>
<p><b>Date:</b> 07/06/10</p>		<p><b>Version:</b> 1.0</p>	

### 5.1.1.6 Artefacts for Interfaces

<b>REQ-PL-006</b>		<b>Artefacts for interfaces</b>	
<p><b>Description:</b> In the process of pattern description, the pattern's interfaces necessary to communicate with the rest of the design have to be defined. To this end, the artefacts needed to describe the interfaces must be available.</p>			
<p><b>Source:</b> General</p>			
<p><b>Reference process:</b> Pattern Life cycle</p>			
<p><b>Rational:</b> To support the definition of the pattern's interfaces.</p>			
<p><b>Type:</b> <i>Mandatory</i></p>		<p><b>Validation criteria:</b> <i>Inspection</i></p>	<p><b>WP Relevance:</b> 4</p>
<p><b>Date:</b> 27/09/10</p>		<p><b>Version:</b> 1.0</p>	



**5.1.1.7 Artefacts for Attributes**

<b>REQ-PL-007</b>		<b>Artefacts for attributes</b>
<b>Description:</b> In the process of pattern description, the attributes (e.g. S&D properties) of the pattern have to be defined. To this end, the artefacts needed to describe the attributes must be available.		
<b>Source:</b> General		
<b>Reference process:</b> Pattern Life cycle		
<b>Rational:</b> To support the definition of the attributes of the pattern.		
<b>Type:</b> <i>Mandatory</i>	<b>Validation criteria:</b> <i>Inspection</i>	<b>WP Relevance:</b> 4
<b>Date:</b> 10/11/10		<b>Version:</b> 1.0

**5.1.1.8 Variability of Artefact**

<b>REQ-PL-008</b>		<b>Variability of artefacts</b>
<b>Description:</b> To be applicable in many domains, the process must support the variability of the pattern. Therefore, the artefacts required to define the pattern must be configurable or adaptable.		
<b>Source:</b> General		
<b>Reference process:</b> Pattern Life cycle		
<b>Rational:</b> To support the use and adaptation of the pattern to different application domains.		
<b>Type:</b> <i>Mandatory</i>	<b>Validation criteria:</b> <i>Inspection</i>	<b>WP Relevance:</b> 4
<b>Date:</b> 07/06/10		<b>Version:</b> 1.0

**5.1.1.9 Extensibility of Existing Artefacts**

<b>REQ-PL-009</b>		<b>Extensibility of existing artefacts</b>
<b>Description:</b> Sometimes, the set of artefacts used to describe and to support the definition of the pattern are not enough. If a gap in the pattern definition is detected, the existing artefacts must be extensible to support the complete definition of the pattern.		
<b>Source:</b> General		
<b>Reference process:</b> Pattern Life cycle		
<b>Rational:</b> To allow extensibility of artefact for new necessities		
<b>Type:</b> <i>Desirable</i>	<b>Validation criteria:</b> <i>Inspection</i>	<b>WP Relevance:</b> 4
<b>Date:</b> 31/08/10		<b>Version:</b> 1.0

**5.1.1.10 Extensibility of the Set of Artefacts**

<b>REQ-PL-010</b>		<b>Extensibility of the set of artefacts</b>
<b>Description:</b> If there are no artefacts available, that support the required needs, the process must support the creation and the integration of new artefacts.		
<b>Source:</b> General		
<b>Reference process:</b> Pattern Life cycle		
<b>Rational:</b> To take into account and to express the new artefacts needed for describing a pattern.		
<b>Type:</b> <i>Mandatory</i>	<b>Validation criteria:</b> <i>Inspection</i>	<b>WP Relevance:</b> 4
<b>Date:</b> 07/06/10		<b>Version:</b> 1.0

## 5.1.2 Models

### 5.1.2.1 Role based Specification (Pattern S&D Expert, Formal Validation, Domain Integrator)

<b>REQ-PL-011</b>		<b>Role based Specification</b>
<b>Description:</b> In the process of pattern description, roles that interact with the pattern in the different phases of the application development must be taken into account in pattern specification.		
<b>Source:</b> General		
<b>Reference process:</b> Pattern Life cycle		
<b>Rational:</b> To identify the roles that interacts with the pattern at an early phase of pattern specification.		
<b>Type:</b> <i>Mandatory</i>	<b>Validation criteria:</b> <i>Inspection</i>	<b>WP Relevance:</b> 4
<b>Date:</b> 07/06/10		<b>Version:</b> 1.0

### 5.1.2.2 Top-down Multi-level Specification (Domain Independent, Domain Dependent Platform Independent, Domain Dependent Platform Specific)

<b>REQ-PL-012</b>		<b>Top-down Multi-level Specification</b>
<b>Description:</b> In the process of pattern description it is advisable to follow a design approach in which the level of detail used in the description of the pattern is ever-increasing. For that reason, the specification of the pattern must be performed from a high level description to a detailed level description.		
<b>Source:</b> Industrial Control / Dependability		
<b>Reference process:</b> Pattern Lifecycle		
<b>Rational:</b> To ease the comprehension of the pattern description.		
<b>Type:</b> <ul style="list-style-type: none"> <li>• <i>Mandatory for Industrial Control</i></li> <li>• <i>Optional for other domains</i></li> </ul>	<b>Validation criteria:</b> <i>Inspection/Test</i>	<b>WP Relevance:</b> 4
<b>Date:</b> 07/06/10		<b>Version:</b> 1.0

### 5.1.2.3 Role based and Level based Pattern Representation

<b>REQ-PL-013</b>		<b>Role based and Level based Pattern representation</b>
<p><b>Description:</b> A series of diagrams have been defined to model the structure and the behaviour of the pattern. The process must provide a list of necessary diagrams for the pattern representation and a description of the approach chosen to apply them in this process. The information provided will vary according to the context in which it will be used (i.e., role, level).</p>		
<p><b>Source:</b> General</p>		
<p><b>Reference process:</b> Pattern Life cycle</p>		
<p><b>Rational:</b> To identify and establish a sequence of the diagrams depending on the context.</p>		
<p><b>Type:</b> <i>Mandatory</i></p>	<p><b>Validation criteria:</b> <i>Inspection</i></p>	<p><b>WP Relevance:</b> 4</p>
<p><b>Date:</b> 27/05/10</p>		<p><b>Version:</b> 1.0</p>

### 5.1.2.4 Portability

<b>REQ-PL-014</b>		<b>Portability</b>
<p><b>Description:</b> In the process of pattern definition, the specification of the pattern must be platform-independent such that the pattern can be rebuilt for a new platform, without affecting any other pattern.</p>		
<p><b>Source:</b> General</p>		
<p><b>Reference process:</b> Pattern Life cycle</p>		
<p><b>Rational:</b> To ease the portability of the pattern to different kinds of platform.</p>		
<p><b>Type:</b> <i>Mandatory</i></p>	<p><b>Validation criteria:</b> <i>Inspection</i></p>	<p><b>WP Relevance:</b> 4</p>
<p><b>Date:</b> 01/09/10</p>		<p><b>Version:</b> 1.0</p>

### 5.1.2.5 Reusability

<b>REQ-PL-015</b>		<b>Reusability</b>
<b>Description:</b> The patterns are designed primarily to be reused in different applications. For that reason, the pattern developer must be aware of the needed artefacts which must be described. A pattern has to be as flexible as possible.		
<b>Source:</b> General		
<b>Reference process:</b> Pattern Life cycle		
<b>Rational:</b> To ease the reusability of the pattern in different domains		
<b>Type:</b> <i>Mandatory</i>	<b>Validation criteria:</b> <i>Inspection</i>	<b>WP Relevance:</b> 4
<b>Date:</b> 01/09/10		<b>Version:</b> 1.0

### 5.1.2.6 Availability of Needed Artefacts Per Role and Per Level

<b>REQ-PL-016</b>		<b>Availability of needed artefacts per role/level</b>
<b>Description:</b> A set of artefacts have been identified to support the pattern description. The process must provide a list of artefacts. These must be organized in such a way that takes into account both the roles involved in the pattern's design process, as the pattern's design level where the artefacts will be used.		
<b>Source:</b> General		
<b>Reference process:</b> Pattern Life cycle		
<b>Rational:</b> To ease the identification of the artefacts required for defining a pattern.		
<b>Type:</b> <i>Mandatory</i>	<b>Validation criteria:</b> <i>Inspection</i>	<b>WP Relevance:</b> 4
<b>Date:</b> 07/06/10		<b>Version:</b> 1.0

**5.1.2.7 Detailed Description of Patterns Internal Behaviour**

<b>REQ-PL-017</b>		<b>Detailed Description of Patterns Internal Behaviour</b>
<b>Description:</b> A pattern must be described in a sufficient level of detail that allows for the formalization and formal validation of its provided properties under the assumption of its required properties.		
<b>Source:</b> General		
<b>Reference process:</b> Pattern Life cycle		
<b>Rational:</b> To have a sufficient description level for using formal validation.		
<b>Type:</b> <i>Mandatory</i>	<b>Validation criteria:</b> <i>Inspection</i>	<b>WP Relevance:</b> 4
<b>Date:</b> 27/05/10		<b>Version:</b> 1.0

**5.1.3 Transformations**

**5.1.3.1 Traceability of Artefacts During the Pattern Life Cycle**

<b>REQ-PL-018</b>		<b>Traceability of artefacts during the pattern life cycle</b>
<b>Description:</b> In the pattern development different artefacts are required for the description of the pattern. The process must provide a mechanism to trace the changes made to the artefacts through the pattern development process.		
<b>Source:</b> General		
<b>Reference process:</b> Pattern Life cycle		
<b>Rational:</b> To trace the different artefacts used in the pattern definition		
<b>Type:</b> <i>Mandatory</i>	<b>Validation criteria:</b> <i>Inspection</i>	<b>WP Relevance:</b> 4
<b>Date:</b> 07/06/10		<b>Version:</b> 1.0

### 5.1.3.2 Traceability of Properties Mechanism

<b>REQ-PL-019</b>		<b>Traceability of properties during mechanism</b>
<b>Description:</b> In the pattern design process, it is necessary to check that properties identified in early stages are adequately implemented in later stages. The process must provide a mechanism to trace these properties through the pattern life cycle.		
<b>Source:</b> General		
<b>Reference process:</b> Pattern Lifecycle		
<b>Rational:</b> To trace the properties implementation throughout the pattern definition.		
<b>Type:</b> <i>Mandatory</i>	<b>Validation criteria:</b> <i>Inspection</i>	<b>WP Relevance:</b> 4
<b>Date:</b> 27/05/10		<b>Version:</b> 1.0

### 5.1.3.3 Consistency between Pattern Levels

<b>REQ-PL-020</b>		<b>Consistency between pattern levels</b>
<b>Description:</b> In order to develop a pattern, different levels to design the pattern have been identified. Throughout the development of these levels, the pattern is involved in a set of transformations. The process must ensure consistency between these transformations during the pattern design.		
<b>Source:</b> General		
<b>Reference process:</b> Pattern Lifecycle		
<b>Rational:</b> To maintain the consistency during pattern design.		
<b>Type:</b> <i>Mandatory</i>	<b>Validation criteria:</b> <i>Inspection</i>	<b>WP Relevance:</b> 4
<b>Date:</b> 27/05/10		<b>Version:</b> 1.0

### 5.1.3.4 Guidelines for Easy Pattern Definition

<b>REQ-PL-021</b>		<b>Guidelines for easy pattern definition</b>
<b>Description:</b> In addition to the diagrams used for pattern description, a specific set of activities has to be carried out for pattern definition. For that reason, the set of actions (procedures) that form the pattern design process must be identified and organized to allow the pattern developer to carry out the pattern design in a clear and precise way.		
<b>Source:</b> Industrial Control / Dependability		
<b>Reference process:</b> Pattern Life cycle		
<b>Rational:</b> To establish a process in which the pattern developer can find the main steps and actions to carry out for pattern definition.		
<b>Type:</b> <i>Mandatory</i>	<b>Validation criteria:</b> <i>Inspection</i>	<b>WP Relevance:</b> 4
<b>Date:</b> 27/05/10		<b>Version:</b> 1.0

## 5.2 Modelling Level

### 5.2.1 Editable Artefacts

#### 5.2.1.1 Pattern Properties

<b>REQ-PL-022</b>		<b>Pattern Properties</b>
<b>Description:</b> In the process of pattern description, the set of properties (e.g. dependability, security, RCES, quality, specific context) provided by the pattern must be specified.		
<b>Source:</b> General		
<b>Reference process:</b> Pattern Lifecycle		
<b>Rational:</b> To easy the classification and selection of the pattern.		
<b>Type:</b> <i>Mandatory</i>	<b>Validation criteria:</b> <i>Inspection</i>	<b>WP Relevance:</b> 2
<b>Date:</b> 27/09/10		<b>Version:</b> 1.0



### 5.2.1.2 Expressiveness of S&D Property Definition

<b>REQ-PL-023</b>		<b>Expressiveness of S&amp;D property definition</b>
<b>Description:</b> In the process of pattern definition, the S&D properties that are addressed by the pattern must be described in certain ways that they are precise, clear and easy to understand.		
<b>Source:</b> General		
<b>Reference process:</b> Pattern Lifecycle		
<b>Rational:</b> To easily understand the meaning of the S&D properties.		
<b>Type:</b> <i>Mandatory</i>	<b>Validation criteria:</b> <i>Inspection</i>	<b>WP Relevance:</b> 2
<b>Date:</b> 27/09/10		<b>Version:</b> 1.0

### 5.2.1.3 Pattern Integration Constraints/Recommendations

<b>REQ-PL-024</b>		<b>Pattern integration constraints/recommendations</b>
<b>Description:</b> In the application development, some recommendations or constraints have to be taken into account before integrating the pattern in the application. The developer needs this information available in the pattern description. This information that the pattern may require (i.e., resources, protocols, architecture, implementation techniques, etc) must be specified. In some cases the constraints available may vary in a pre-defined range in order to support the adaptability of the pattern.		
<b>Source:</b> General		
<b>Reference process:</b> Pattern Lifecycle		
<b>Rational:</b> To support the developer in the integration of the pattern		
<b>Type:</b> <i>Mandatory</i>	<b>Validation criteria:</b> <i>Inspection</i>	<b>WP Relevance:</b> 2
<b>Date:</b> 31/08/10		<b>Version:</b> 1.0

#### 5.2.1.4 Validation Constraints

<b>REQ-PL-025</b>		<b>Validation Constraints</b>
<p><b>Description:</b> At the end of the pattern integration process, the integration must be validated in order to check that the provided functionality has been reached. The constraints to be considered (related to the pattern) in the pattern validation process must be available.</p>		
<p><b>Source:</b> General</p>		
<p><b>Reference process:</b> Pattern Lifecycle</p>		
<p><b>Rational:</b> To identify the restrictions to take into account in the pattern validation process.</p>		
<p><b>Type:</b> <i>Mandatory</i></p>	<p><b>Validation criteria:</b> <i>Inspection</i></p>	<p><b>WP Relevance:</b> 2</p>
<p><b>Date:</b> 27/09/10</p>		<p><b>Version:</b> 1.0</p>

#### 5.2.1.5 Repository Search Keys

<b>REQ-PL-026</b>		<b>Repository Search Keys</b>
<p><b>Description:</b> In the pattern definition process, information relevant to the description of the pattern has been defined. Keywords that can help in the search of patterns must be available.</p>		
<p><b>Source:</b> General</p>		
<p><b>Reference process:</b> Pattern Lifecycle</p>		
<p><b>Rational:</b> To ease the search of the patterns.</p>		
<p><b>Type:</b> <i>Mandatory</i></p>	<p><b>Validation criteria:</b> <i>Inspection</i></p>	<p><b>WP Relevance:</b> 2</p>
<p><b>Date:</b> 27/05/10</p>		<p><b>Version:</b> 1.0</p>

### 5.2.1.6 Adaptable Pattern Description According to Role

<b>REQ-PL-027</b>		<b>Adaptable Pattern Description According to Role</b>
<p><b>Description:</b> In the process of pattern integration, the function or set of functions that the pattern is able to provide must be available. The set of functions provided must vary according to the role of the person using the pattern.</p>		
<p><b>Source:</b> General</p>		
<p><b>Reference process:</b> Pattern Lifecycle</p>		
<p><b>Rational:</b> In order to ease the identification and selection of the pattern in the repository.</p>		
<p><b>Type:</b> <i>Mandatory</i></p>	<p><b>Validation criteria:</b> <i>Inspection</i></p>	<p><b>WP Relevance:</b> 2</p>
<p><b>Date:</b> 31/08/10</p>		<p><b>Version:</b> 1.0</p>

### 5.2.1.7 Adaptable Pattern Description According to Level

<b>REQ-PL-028</b>		<b>Adaptable Pattern Description According to Level</b>
<p><b>Description:</b> In the process of pattern integration, function or set of functions that the pattern is able to provide must be available. The set of functions provided must vary according to the level where the pattern is used.</p>		
<p><b>Source:</b> General</p>		
<p><b>Reference process:</b> Pattern Lifecycle</p>		
<p><b>Rational:</b> In order to ease the identification and selection of the pattern in the repository.</p>		
<p><b>Type:</b> <i>Mandatory</i></p>	<p><b>Validation criteria:</b> <i>Inspection</i></p>	<p><b>WP Relevance:</b> 2</p>
<p><b>Date:</b> 27/09/10</p>		<p><b>Version:</b> 1.0</p>

### 5.2.1.8 Adaptable Pattern Internal Structure According to Role

<b>REQ-PL-029</b>		<b>Adaptable Pattern internal structure According to Role</b>	
<b>Description:</b> In the process of pattern integration, the pattern internal structure must be available. The details of the pattern internal structure provided may vary according to the role of the person using the pattern.			
<b>Source:</b> General			
<b>Reference process:</b> Pattern Life cycle			
<b>Rational:</b> In order to ease the identification and selection of the pattern in the repository.			
<b>Type:</b> <i>Mandatory</i>	<b>Validation criteria:</b> <i>Inspection</i>	<b>WP Relevance:</b> 2	
<b>Date:</b> 07/06/10		<b>Version:</b> 1.0	

### 5.2.1.9 Adaptable Pattern Internal Structure According to Level

<b>REQ-PL-030</b>		<b>Adaptable Pattern internal structure According to Level</b>	
<b>Description:</b> In the process of pattern integration, the pattern internal structure must be available. The details of the pattern internal structure provided may vary according to the level where the pattern is used.			
<b>Source:</b> General			
<b>Reference process:</b> Pattern Life cycle			
<b>Rational:</b> In order to ease the identification and selection of the pattern in the repository.			
<b>Type:</b> <i>Mandatory</i>	<b>Validation criteria:</b> <i>Inspection</i>	<b>WP Relevance:</b> 2	
<b>Date:</b> 07/06/10		<b>Version:</b> 1.0	

### 5.2.1.10 Editable Domain Parameters

<b>REQ-PL-031</b>		<b><i>Editable Domain Parameters</i></b>
<p><b>Description:</b> The patterns can be used for different application sectors. It can also be implemented in different ways to improve the security, dependability, time of execution; etc. The parameters that can be useful for the pattern specialization must be provided. These parameters must be editable to support the variability of the pattern.</p>		
<p><b>Source:</b> General</p>		
<p><b>Reference process:</b> Pattern Life cycle</p>		
<p><b>Rational:</b> To identify the characteristics that allows the adaptation of the pattern in order to reach the desire functionality in different ways.</p>		
<p><b>Type:</b> <i>Mandatory</i></p>	<p><b>Validation criteria:</b> <i>Inspection</i></p>	<p><b>WP Relevance:</b> 2</p>
<p><b>Date:</b> 27/05/10</p>		<p><b>Version:</b> 1.0</p>

### 5.2.1.11 Pattern Domain Dependent Information

<b>REQ-PL-032</b>		<b><i>Pattern Domain dependent information</i></b>
<p><b>Description:</b> In the event that the identified pattern is used primarily in a specific application domain, the application domain to which the pattern belongs must be specified.</p>		
<p><b>Source:</b> General</p>		
<p><b>Reference process:</b> Pattern Lifecycle</p>		
<p><b>Rational:</b> In the process of selecting the pattern, this domain dependent information helps the designer to decide the most appropriate pattern for the application.</p>		
<p><b>Type:</b> <i>Mandatory</i></p>	<p><b>Validation criteria:</b> <i>Inspection</i></p>	<p><b>WP Relevance:</b> 2</p>
<p><b>Date:</b> 07/06/10</p>		<p><b>Version:</b> 1.0</p>

## 5.2.2 Editable Models

### 5.2.2.1 Pattern Search Mechanism (Level, Role)

REQ-PL-033		Pattern Search Mechanism (Level, Role)
<p><b>Description:</b> The patterns are stored in a repository of patterns. To access them, a mechanism to search the patterns must be provided. This mechanism can take into account certain search criteria that can simplify the search. Moreover, the information provided about the pattern can change depending on the level where the pattern is used and the role of the person involved with the pattern.</p>		
<p><b>Source:</b> General</p>		
<p><b>Reference process:</b> Pattern Lifecycle</p>		
<p><b>Rational:</b> To ease the search of the patterns, and the interaction with the patterns.</p>		
<p><b>Type:</b> <i>Mandatory</i></p>	<p><b>Validation criteria:</b> <i>Inspection</i></p>	<p><b>WP Relevance:</b> 2</p>
<p><b>Date:</b> 27/05/10</p>		<p><b>Version:</b> 1.0</p>

### 5.2.2.2 Description of Pattern Functionality

REQ-PL-034		Description of pattern functionality
<p><b>Description:</b> Once the artefacts for pattern description have been identified, and the strategies that have to be taken into account for pattern definition have been established, the pattern developer must specify the functionality, attributes and interfaces of the pattern.</p>		
<p><b>Source:</b> General</p>		
<p><b>Reference process:</b> Pattern Life cycle</p>		
<p><b>Rational:</b> To identify the desired functionality to be achieved with the pattern.</p>		
<p><b>Type:</b> <i>Mandatory</i></p>	<p><b>Validation criteria:</b> <i>Inspection</i></p>	<p><b>WP Relevance:</b> 3</p>
<p><b>Date:</b> 06/10/10</p>		<p><b>Version:</b> 1.0</p>

### 5.2.2.3 Pattern Description

<b>REQ-PL-035</b>		<b>Pattern Description</b>
<p><b>Description:</b> In the pattern definition process, a set of information (documents, diagrams, etc) to describe the pattern is generated. A mechanism to have the pattern description available must be provided.</p>		
<p><b>Source:</b> General</p>		
<p><b>Reference process:</b> Pattern Lifecycle</p>		
<p><b>Rational:</b> To ease the organization and storing of the pattern description.</p>		
<p><b>Type:</b> <i>Mandatory</i></p>	<p><b>Validation criteria:</b> <i>Inspection</i></p>	<p><b>WP Relevance:</b> 2</p>
<p><b>Date:</b> 27/05/10</p>		<p><b>Version:</b> 1.0</p>

### 5.2.2.4 Establish Dependency between Level Specific Views of Model

<b>REQ-PL-036</b>		<b>Establish Dependency between Specific views of the Model</b>
<p><b>Description:</b> The patterns can be used in different kinds of applications. These applications require that the pattern must be adapted. To support the correct use, instantiation and transformation of the patterns in the application, a relation between Specific views of the model (Level / Role Specific views) must be established.</p>		
<p><b>Source:</b> General</p>		
<p><b>Reference process:</b> Pattern Life cycle</p>		
<p><b>Rational:</b> To guaranty that the pattern variability has been taken into account in the integration of the pattern in the application.</p>		
<p><b>Type:</b> <i>Mandatory</i></p>	<p><b>Validation criteria:</b> <i>Inspection</i></p>	<p><b>WP Relevance:</b> 2</p>
<p><b>Date:</b> 31/08/10</p>		<p><b>Version:</b> 1.0</p>

### 5.2.2.5 Pattern Characterization

<b>REQ-PL-037</b>		<b>Pattern characterization</b>
<b>Description:</b> Once the parameters required for customizing the pattern have been identified, they must be accessible to customize the pattern for the application.		
<b>Source:</b> General		
<b>Reference process:</b> Pattern Life cycle		
<b>Rational:</b> In the process of pattern characterization, a means that support the storage, access and manipulation of these parameters will allow an easy adaption of the pattern.		
<b>Type:</b> <i>Mandatory</i>	<b>Validation criteria:</b> <i>Inspection</i>	<b>WP Relevance:</b> 2
<b>Date:</b> 27/05/10		<b>Version:</b> 1.0

### 5.2.2.6 Combining Patterns

<b>REQ-PL-038</b>		<b>Combining Patterns</b>
<b>Description:</b> In some applications, the functionality to be achieved can only be reached by grouping a number of patterns. The set of constraints and characteristics to take into account in the combination of the patterns must be provided.		
<b>Source:</b> General		
<b>Reference process:</b> Pattern Life cycle		
<b>Rational:</b> To consider all the characteristics in the pattern design process to support the composition of a number of patterns in an application.		
<b>Type:</b> <i>Optional</i>	<b>Validation criteria:</b> <i>Inspection</i>	<b>WP Relevance:</b> 2
<b>Date:</b> 27/09/10		<b>Version:</b> 1.0



### 5.2.2.7 Identification of Pattern Interfaces

<b>REQ-PL-039</b>		<b>Identification of pattern interfaces</b>
<b>Description:</b> In each application, the pattern must interact with other components or patterns. The interfaces of the pattern with the rest of the application must be provided.		
<b>Source:</b> General		
<b>Reference process:</b> Pattern Life cycle		
<b>Rational:</b> To establish the way the pattern interacts with the rest of the components of the application (included other patterns).		
<b>Type:</b> <i>Optional</i>	<b>Validation criteria:</b> <i>Inspection</i>	<b>WP Relevance:</b> 2
<b>Date:</b> 07/06/10		<b>Version:</b> 1.0

### 5.2.2.8 Substitute Pattern

<b>REQ-PL-040</b>		<b>Substitute Pattern</b>
<b>Description:</b> Right now, the applications are used for different kind of products where some functionality can be affected. To support the change or adaption of this functionality in the application, the structure of the patterns must be defined as modular as possible in order to ease the replacement of the pattern.		
<b>Source:</b> General		
<b>Reference process:</b> Pattern Life cycle		
<b>Rational:</b> To ease the replacement of the pattern in the application.		
<b>Type:</b> <i>Mandatory</i>	<b>Validation criteria:</b> <i>Inspection</i>	<b>WP Relevance:</b> 2
<b>Date:</b> 01/09/10		<b>Version:</b> 1.0

## 5.2.3 Editable Transformations

### 5.2.3.1 Transformation Rules

<b>REQ-PL-041</b>		<b>Transformation Rules</b>
<b>Description:</b> In the pattern integration process, the pattern must be adapted to take into account the constraints of the application. The rules required to adapt / transform the pattern (interfaces, internal structure, etc) for the application must be available.		
<b>Source:</b> General		
<b>Reference process:</b> Pattern Life cycle		
<b>Rational:</b> To easy the pattern integration process.		
<b>Type:</b> <i>Mandatory</i>	<b>Validation criteria:</b> <i>Inspection</i>	<b>WP Relevance:</b> 2
<b>Date:</b> 27/10/10		<b>Version:</b> 1.0

### 5.2.3.2 Assistance / Guidelines for Easy Pattern Integration

<b>REQ-PL-042</b>		<b>Assistance / Guidelines for easy pattern integration</b>
<b>Description:</b> Within the pattern integration process, a set of steps and restrictions has to be followed. Assistance in the pattern integration may be provided to carry out the actions related to this process.		
<b>Source:</b> Industrial Control / Dependability		
<b>Reference process:</b> Pattern Life cycle		
<b>Rational:</b> To ease the integration of the pattern in the application.		
<b>Type:</b> <i>Optional</i>	<b>Validation criteria:</b> <i>Inspection</i>	<b>WP Relevance:</b> 2
<b>Date:</b> 27/05/10		<b>Version:</b> 1.0

### 5.2.3.3 Assurance of Pattern Dependency Consistency

<b>REQ-PL-043</b>		<b>Assurance of pattern dependency consistency</b>	
<p><b>Description:</b> In some cases, the patterns can be defined using some characteristics of other patterns in order to reduce the time to develop a new pattern and to avoid the introduction of a new source of errors in the new pattern. For that reason, when a pattern is deleted in the repository, all the dependencies with other patterns must be checked and replaced to avoid any undesired effect on the functionality of the others patterns in the repository.</p>			
<p><b>Source:</b> General</p>			
<p><b>Reference process:</b> Pattern Life cycle</p>			
<p><b>Rational:</b> Deleting a pattern must not compromise the consistency of the repository.</p>			
<p><b>Type:</b> <i>Optional</i></p>		<p><b>Validation criteria:</b> <i>Inspection</i></p>	
		<p><b>WP Relevance:</b> 2</p>	
<p><b>Date:</b> 03/06/10</p>		<p><b>Version:</b> 1.0</p>	

### 5.2.3.4 Traceability of Properties during Pattern Life Cycle

<b>REQ-PL-044</b>		<b>Traceability of properties during pattern life cycle</b>	
<p><b>Description:</b> In the pattern design process, the properties that the pattern implements must be traced trough the pattern life cycle.</p>			
<p><b>Source:</b> General</p>			
<p><b>Reference process:</b> Pattern Lifecycle</p>			
<p><b>Rational:</b> To trace the properties implementation throughout the pattern definition.</p>			
<p><b>Type:</b> <i>Mandatory</i></p>		<p><b>Validation criteria:</b> <i>Inspection</i></p>	
		<p><b>WP Relevance:</b> 3</p>	
<p><b>Date:</b> 27/05/10</p>		<p><b>Version:</b> 1.0</p>	

### 5.2.3.5 Repository - Pattern Validation

<b>REQ-PL-045</b>		<b>Repository - Pattern validation</b>
<p><b>Description:</b> Any pattern (whether new or modified) that is added to the repository must be validated according to the S&amp;D validation method / approach associated with the pattern. The patterns provided in the repository are validated.</p>		
<p><b>Source:</b> General</p>		
<p><b>Reference process:</b> Pattern Life cycle</p>		
<p><b>Rational:</b> To avoid the introduction of S&amp;D lacks in the patterns added to the repository.</p>		
<p><b>Type:</b> <i>Mandatory</i></p>	<p><b>Validation criteria:</b> <i>Inspection</i></p>	<p><b>WP Relevance:</b> 2, 3</p>
<p><b>Date:</b> 03/06/10</p>		<p><b>Version:</b> 1.0</p>

## 6 Requirements on Integration Process Transformations

In this section, we will identify and detail all requirements related to the integration process. This section is divided in two subsections. The first one, called “Domain Independent Pattern Integration”, is relevant for the task 3.1 and represents a part of the deliverable D3.1. The second one, titled “Domain Specific Pattern Integration” corresponds to the deliverable D2.2.

### 6.1 Domain Independent Pattern Integration

#### 6.1.1 Conceptual level

##### 6.1.1.1 Pattern Selection / Discrimination

<b>REQ-IPT-001</b>		<b>Pattern selection / discrimination</b>
<b>Description:</b> Some information (i.e., properties, interfaces, etc) has been defined to provide a complete description of the pattern. This information must be available for the pattern user to select the appropriated pattern for the application.		
<b>Source:</b> General		
<b>Reference process:</b> Domain Independent Pattern Integration process		
<b>Rational:</b> The information about the pattern description must be available to ease the selection of the pattern for the application developer.		
<b>Type:</b> <i>Mandatory</i>	<b>Validation criteria:</b> <i>Inspection</i>	<b>WP Relevance:</b> 3
<b>Date:</b> 07/06/10		<b>Version:</b> 1.0

##### 6.1.1.2 Integration – Pattern Validation

<b>REQ-IPT-002</b>		<b>Integration - pattern validation</b>
<b>Description:</b> Once the pattern has been integrated in the application, it must be guaranteed that the features provided by the integrated pattern (e.g. S&D properties) have been validated and that the pattern has been properly integrated in the application.		
<b>Source:</b> General		
<b>Reference process:</b> Domain Independent Pattern Integration process		
<b>Rational:</b> To prove that the integration of the pattern in the application have been performed properly.		
<b>Type:</b> <i>Mandatory</i>	<b>Validation criteria:</b> <i>Inspection</i>	<b>WP Relevance:</b> 3
<b>Date:</b> 27/05/10		<b>Version:</b> 1.0

### 6.1.1.3 Traceability of S&D Requirements During Pattern Integration

<b>REQ-IPT-003</b>		<b>Traceability of S&amp;D requirements during pattern integration</b>
<b>Description:</b> Before integrating a pattern in a specific application, some S&D requirements have been defined. These requirements are met by the patterns and must be traced through the integration, verification, and validation of the pattern in the application.		
<b>Source:</b> General		
<b>Reference process:</b> Domain Independent Pattern Integration process		
<b>Rational:</b> To trace the S&D requirements in the application		
<b>Type:</b> <i>Mandatory</i>	<b>Validation criteria:</b> <i>Inspection</i>	<b>WP Relevance:</b> 3
<b>Date:</b> 27/05/10		<b>Version:</b> 1.0

## 6.1.2 Modelling

### 6.1.2.1 Specifying S&D Integration Requirement

<b>REQ-IPT-004</b>		<b>S&amp;D integration requirements definition</b>
<b>Description:</b> There are some specific constraints against the application e.g. environmental constraints which must be taken into account in the process of integration of the pattern in a specific application. With this information, the S&D integration requirements must be defined in such ways that they are precise, clear and easy to understand.		
<b>Source:</b> General		
<b>Reference process:</b> Domain Independent Pattern Integration process		
<b>Rational:</b> To identify the requirements to fulfil for the integration of the pattern in the application and also to easily understand the meaning of the S&D integration requirements.		
<b>Type:</b> <i>Mandatory</i>	<b>Validation criteria:</b> <i>Inspection</i>	<b>WP Relevance:</b> 3
<b>Date:</b> 27/05/10		<b>Version:</b> 1.0

**6.1.2.2 Determination of the User's System Model**

<b>REQ-IPT-005</b>	<b>Determination of the user's system model</b>	
<p><b>Description:</b> For a successful pattern integration in the user's system or application it is necessary to take into account several information about the user's system model in order to validate if the provided pattern functionality has been reached. For instance this includes system components and their interfaces, S&amp;D objectives, and S&amp;D threats that needs to be handled. The user should be leaded by a list of questions to specify these properties. The questionnaire should be context sensitive and should make suggestions (e.g., Did you think of threat XY? Is that relevant for you?). It should collect information in a structured, iterative process:</p> <ol style="list-style-type: none"> <li>1. Actors/roles</li> <li>2. Data/interfaces</li> <li>3. Functions/use cases</li> <li>4. Security assets</li> <li>5. Potential attackers</li> <li>6. Potential threats</li> <li>7. Misuse cases</li> <li>8. Security environment</li> <li>9. Attacker model</li> <li>10. Attack paths</li> <li>11. Attack potentials</li> <li>12. Potential damages</li> <li>13. Risk evaluation</li> <li>14. Necessary security functionality</li> <li>15. Security properties</li> </ol> <p>and finally lead to a solution, i.e., preliminary selection of patterns relevant for the user's problem.</p>		
<b>Source:</b> General		
<b>Reference process:</b> Domain Independent Pattern Integration		
<p><b>Rational:</b> To easily determine an extensive, unambiguous view of the user's system model that allows validating the pattern integration and furthermore can be used to improve the efficiency of the pattern search process, i.e., automatically make a preliminary selection of patterns relevant for the user's problem.</p>		
<b>Type:</b> <i>Mandatory</i>	<b>Validation criteria:</b> Inspection	<b>WP Relevance:</b> 3
<b>Date:</b> 09/11/2010		<b>Version:</b> 1.0

### 6.1.3 Transformation

#### 6.1.3.1 Guidelines for Easy Pattern Integration (Domain Independent Actions)

<b>REQ-IPT-006</b>		<b>Guidelines for easy pattern integration (domain independent actions)</b>	
<b>Description:</b> The set of domain independent actions (procedures) required for integrating the pattern in an application must be defined.			
<b>Source:</b> General			
<b>Reference process:</b> Domain Independent Pattern Integration process			
<b>Rational:</b> To reduce the complexity of the integration of the pattern in the application.			
<b>Type:</b> <i>Mandatory</i>	<b>Validation criteria:</b> <i>Inspection</i>	<b>WP Relevance:</b> 3	
<b>Date:</b> 27/05/10		<b>Version:</b> 1.0	

#### 6.1.3.2 Match the Interfaces between the Pattern and the Application

<b>REQ-IPT-007</b>		<b>Match the interfaces between the pattern and the application</b>	
<b>Description:</b> In the process of integrating the pattern into the application, the interfaces between the application and the pattern must be analyzed and adapted in order to get a correct integration.			
<b>Source:</b> General			
<b>Reference process:</b> Domain Independent Pattern Integration process			
<b>Rational:</b> To get a correct interaction between the pattern and the application			
<b>Type:</b> <i>Mandatory</i>	<b>Validation criteria:</b> <i>Inspection</i>	<b>WP Relevance:</b> 3	
<b>Date:</b> 23/08/10		<b>Version:</b> 1.0	



### 6.1.3.3 Consistency between Domain Independent and Domain Specific Models

<b>REQ-IPT-008</b>		<b>Consistency between domain independent and domain specific models</b>
<b>Description:</b> During the pattern integration process, transformations are required to take into account some domain specific capabilities and constraints. The pattern user must guarantee consistency between these transformations during pattern integration.		
<b>Source:</b> General		
<b>Reference process:</b> Domain Independent Pattern Integration process		
<b>Rational:</b> To guarantee the correct transformation of the pattern into the pattern integration process.		
<b>Type:</b> <i>Mandatory</i>	<b>Validation criteria:</b> <i>Inspection</i>	<b>WP Relevance:</b> 3
<b>Date:</b> 27/09/10		<b>Version:</b> 1.0

## 6.2 Domain Specific Pattern Integration

### 6.2.1 Editable Conceptual Level

#### 6.2.1.1 Project's Organizational Model

<b>REQ-IPT-009</b>		<b>Project's organizational model</b>
<b>Description:</b> The process should provide the information (documentation) related to the pattern following the organizational model (based on standards such as ISO 9000-1) defined for the specific project.		
<b>Source:</b> General		
<b>Reference process:</b> Domain Specific Pattern Integration process		
<b>Rational:</b> To guarantee quality assurance		
<b>Type:</b> <i>Optional</i>	<b>Validation criteria:</b> <i>Inspection</i>	<b>WP Relevance:</b> 2
<b>Date:</b> 27/05/10		<b>Version:</b> 1.0

### 6.2.1.2 Specialization of Artefact Instances

<b>REQ-IPT-010</b>		<b>Specialization of artefact instances</b>
<b>Description:</b> The process must provide a mechanism to specialize the artefact instances for domain specific needs.		
<b>Source:</b> General		
<b>Reference process:</b> Domain Specific Pattern Integration process		
<b>Rational:</b> To characterize the artefacts to the domain specific needs.		
<b>Type:</b> <i>Mandatory</i>	<b>Validation criteria:</b> <i>Inspection</i>	<b>WP Relevance:</b> 2
<b>Date:</b> 27/05/10		<b>Version:</b> 1.0

## 6.2.2 Editable Modelling

### 6.2.2.1 Using Coding Standards

<b>REQ-IPT-011</b>		<b>Using coding standards</b>
<b>Description:</b> In the implementation of the pattern, the process should generate the pattern code following the strict syntactic coding rules defined for the application.		
<b>Source:</b> General		
<b>Reference process:</b> Domain Specific Pattern Integration process		
<b>Rational:</b> To respect quality restrictions		
<b>Type:</b> <i>Optional</i>	<b>Validation criteria:</b> <i>Analysis</i>	<b>WP Relevance:</b> 2
<b>Date:</b> 27/05/10		<b>Version:</b> 1.0

### 6.2.2.2 Adaptation of Pattern to Resource Constraints

<b>REQ-IPT-012</b>		<b>Adaptation of pattern to resource constraints</b>	
<b>Description:</b> The actions related to the pattern's integration into the application are linked to a set of transformations in order to consider several constraints in its use. The process must consider the resource constraints during the pattern adaptation.			
<b>Source:</b> General			
<b>Reference process:</b> Domain Specific Pattern Integration process			
<b>Rational:</b> To take into account the system resource constraints in the pattern integration process			
<b>Type:</b> <i>Mandatory</i>	<b>Validation criteria:</b> <i>Inspection</i>	<b>WP Relevance:</b> 2	
<b>Date:</b> 07/10/10		<b>Version:</b> 1.0	

### 6.2.3 Editable Transformations

#### 6.2.3.1 Guidelines for Easy Pattern Integration (Domain Specific Actions)

<b>REQ-IPT-013</b>		<b>Guidelines for easy pattern integration (domain specific actions)</b>	
<b>Description:</b> The set of domain specific actions (procedures) required for integrating the pattern in a specific project must be defined.			
<b>Source:</b> General			
<b>Reference process:</b> Domain Specific Pattern Integration process			
<b>Rational:</b> To ease the identification and implementation of the domain specific actions in the pattern integration.			
<b>Type:</b> <i>Mandatory</i>	<b>Validation criteria:</b> <i>Inspection</i>	<b>WP Relevance:</b> 2	
<b>Date:</b> 07/06/10		<b>Version:</b> 1.0	

### 6.2.3.2 Adaptation of Pattern to Domain Specific Constraints

<b>REQ-IPT-014</b>		<b>Adaptation of pattern to domain specific constraints</b>
<p><b>Description:</b> The actions related to the pattern's integration into the application are linked to a set of transformations in order to consider several constraints in its use. The process must consider the domain specific constraints during the pattern adaptation.</p>		
<p><b>Source:</b> General</p>		
<p><b>Reference process:</b> Domain Specific Pattern Integration process</p>		
<p><b>Rational:</b> To take into accounts the domain specific constraints in the pattern integration process.</p>		
<p><b>Type:</b> <i>Mandatory</i></p>	<p><b>Validation criteria:</b> <i>Inspection</i></p>	<p><b>WP Relevance:</b> 2</p>
<p><b>Date:</b> 07/10/10</p>		<p><b>Version:</b> 1.0</p>

### 6.2.3.3 Certification

<b>REQ-IPT-015</b>		<b>Certification</b>
<p><b>Description:</b> Where necessary, the formal validation must follow a domain specific certification criterion.</p>		
<p><b>Source:</b> General</p>		
<p><b>Reference process:</b> Formal Validation Process</p>		
<p><b>Rational:</b> To take into account specific criteria in the application domains that can be relevant in the validation process.</p>		
<p><b>Type:</b> <i>Optional</i></p>	<p><b>Validation criteria:</b> <i>Inspection</i></p>	<p><b>WP Relevance:</b> 2</p>
<p><b>Date:</b> 27/05/10</p>		<p><b>Version:</b> 1.0</p>

#### 6.2.3.4 Integration of the Formal Validation

<b>REQ-IPT-016</b>		<b>Integration of the formal validation</b>
<p><b>Description:</b> Once the formal validation of the application has been performed, the results of this evaluation have to be available in order to analyze them. For that reason, the results of the formal validation must be distributed to the participants of the project in a format consistent with the engineering process.</p>		
<p><b>Source:</b> General</p>		
<p><b>Reference process:</b> Formal Validation Process</p>		
<p><b>Rational:</b> The pattern repository and integration process must foresee possibilities to inspect the S&amp;D proof.</p>		
<p><b>Type:</b> <i>Mandatory</i></p>	<p><b>Validation criteria:</b> <i>Inspection</i></p>	<p><b>WP Relevance:</b> 2</p>
<p><b>Date:</b> 27/05/10</p>		<p><b>Version:</b> 1.0</p>

## 7 Requirements on Repository Structure

In this section, we will identify and detail all requirements related to the repository structure. This section is divided in two subsections. The first one, called “Conceptual Level”, is relevant for the task 4.1 and represents a part of the deliverable D4.1. The second one, titled “Modelling Level” corresponds to the deliverable D2.2.

### 7.1 Conceptual Level

#### 7.1.1 Artefacts

##### 7.1.1.1 Domain Independent Repository Construction

<b>REQ-RS-001</b>		<b>Domain independent repository construction</b>	
<b>Description:</b> The repository must support the acquisition and categorization of the set of domain independent artefacts required for the repository construction.			
<b>Source:</b> General			
<b>Reference process:</b> Repository Structure			
<b>Rational:</b> To have all the required domain independent artefacts identified for pattern implementation in the repository.			
<b>Type:</b> <i>Mandatory</i>	<b>Validation criteria:</b> <i>Inspection</i>	<b>WP Relevance:</b> 4	
<b>Date:</b> 07/06/10		<b>Version:</b> 1.0	

##### 7.1.1.2 Domain Specific Repository Construction

<b>REQ-RS-002</b>		<b>Domain specific repository construction</b>	
<b>Description:</b> The repository must support the acquisition and categorization of the set of domain specific artefacts required for the repository construction.			
<b>Source:</b> General			
<b>Reference process:</b> Repository Structure			
<b>Rational:</b> To have all the required domain specific artefacts identified for pattern implementation in the repository.			
<b>Type:</b> <i>Mandatory</i>	<b>Validation criteria:</b> <i>Inspection</i>	<b>WP Relevance:</b> 4	
<b>Date:</b> 07/06/10		<b>Version:</b> 1.0	

### 7.1.1.3 Domain Independent Repository Organization

<b>REQ-RS-003</b>		<b>Domain independent repository organization</b>	
<b>Description:</b> The repository must support the acquisition and categorization of the set of domain independent artefacts required for the repository organization.			
<b>Source:</b> General			
<b>Reference process:</b> Repository Structure			
<b>Rational:</b> To have all the required domain independent artefacts for the patterns indexing.			
<b>Type:</b> <i>Mandatory</i>	<b>Validation criteria:</b> <i>Inspection</i>	<b>WP Relevance:</b> 4	
<b>Date:</b> 07/06/10		<b>Version:</b> 1.0	

### 7.1.1.4 Domain Specific Repository Organization

<b>REQ-RS-004</b>		<b>Domain specific repository organization</b>	
<b>Description:</b> The repository must support the acquisition and categorization of the set of domain specific artefacts required for the repository organization.			
<b>Source:</b> General			
<b>Reference process:</b> Repository Structure			
<b>Rational:</b> To have all the required domain specific artefacts for the patterns indexing.			
<b>Type:</b> <i>Mandatory</i>	<b>Validation criteria:</b> <i>Inspection</i>	<b>WP Relevance:</b> 4	
<b>Date:</b> 07/06/10		<b>Version:</b> 1.0	

### 7.1.1.5 Domain Independent Pattern Retrieval

<b>REQ-RS-005</b>		<b>Domain independent patterns retrieval</b>
<b>Description:</b> The repository must support the acquisition and categorization of the set of domain independent artefacts which are required for the implementation of the patterns retrieval policies (i.e., techniques of searching and selection of patterns).		
<b>Source:</b> General		
<b>Reference process:</b> Repository Structure		
<b>Rational:</b> To have all the required domain independent tools which mechanize the identification process.		
<b>Type:</b> <i>Mandatory</i>	<b>Validation criteria:</b> <i>Inspection</i>	<b>WP Relevance:</b> 4
<b>Date:</b> 07/06/10		<b>Version:</b> 1.0

### 7.1.1.6 Domain Specific Patterns Retrieval

<b>REQ-RS-006</b>		<b>Domain specific patterns retrieval</b>
<b>Description:</b> The repository must support the acquisition and categorization of the set of domain specific artefacts required for the implementation of the patterns retrieval policies.		
<b>Source:</b> General		
<b>Reference process:</b> Repository Structure		
<b>Rational:</b> To have all the required domain specific artefacts which mechanize the identification process.		
<b>Type:</b> <i>Mandatory</i>	<b>Validation criteria:</b> <i>Inspection</i>	<b>WP Relevance:</b> 4
<b>Date:</b> 07/06/10		<b>Version:</b> 1.0



**7.1.1.7 Visualisation of a Pattern**

<b>REQ-RS-007</b>		<b>Visualisation of a pattern</b>
<b>Description:</b> The repository must support the acquisition and categorization of the set of graphical artefacts required for the visualisation of a pattern		
<b>Source:</b> General		
<b>Reference process:</b> Repository Structure		
<b>Rational:</b> To have all the required graphical artefacts identified for pattern visualisation in the repository.		
<b>Type:</b> <i>Mandatory</i>	<b>Validation criteria:</b> <i>Inspection</i>	<b>WP Relevance:</b> 4
<b>Date:</b> 28/10/10		<b>Version:</b> 1.0

**7.1.1.8 Role Dependent Visualisation of the Repository**

<b>REQ-RS-009</b>		<b>Role Dependent Visualisation of the Repository</b>
<b>Description:</b> The repository can be viewed in accordance to the role of the user. Its role specific representation must be used.		
<b>Source:</b> General		
<b>Reference process:</b> Repository Structure		
<b>Rational:</b> To have all the required graphical artefacts identified for the visualisation of the repository according to the role user.		
<b>Type:</b> <i>Mandatory</i>	<b>Validation criteria:</b> <i>Inspection</i>	<b>WP Relevance:</b> 4
<b>Date:</b> 28/10/10		<b>Version:</b> 1.0

## 7.1.2 Modelling

### 7.1.2.1 Guidelines on Interactions with the Repository

<b>REQ-RS-010</b>		<b>Guidelines for an easy interaction with the repository</b>
<b>Description:</b> The set of domain specific actions (procedures) required to interact with the repository must be defined. The process must provide guidelines.		
<b>Source:</b> General		
<b>Reference process:</b> Repository Structure		
<b>Rational:</b> To ease the use of the repository.		
<b>Type:</b> <i>Mandatory</i>	<b>Validation criteria:</b> <i>Inspection</i>	<b>WP Relevance:</b> 4
<b>Date:</b> 07/07/10		<b>Version:</b> 1.0

### 7.1.2.2 Multiview Support for Domain and Actor Specific Adaptation

<b>REQ-RS-011</b>		<b>Multiview support for domain and actor specific adaptation</b>
<b>Description:</b> Depending on the application domain, a pattern can be implemented through different ways according to specific concern focuses. For this purpose, the process has to support different views of the pattern depending on the role of the pattern developer or user.		
<b>Source:</b> General		
<b>Reference process:</b> Repository Structure		
<b>Rational:</b> To ease the interaction and the understanding of the pattern for different stakeholders of the project.		
<b>Type:</b> <i>Mandatory</i>	<b>Validation criteria:</b> <i>Inspection</i>	<b>WP Relevance:</b> 4
<b>Date:</b> 07/06/10		<b>Version:</b> 1.0

**7.1.2.3 Repository Compliant Description of the Pattern**

<b>REQ-RS-014</b>		<b>Repository compliant description of the pattern</b>
<p><b>Description:</b> The repository must provide a mechanism to collect and separate all the information generated throughout the pattern definition and generation. Moreover the repository must support the extended information generated through the adaptation of the pattern to the application and provide this information with existent information of the pattern.</p>		
<p><b>Source:</b> General</p>		
<p><b>Reference process:</b> Repository Structure</p>		
<p><b>Rational:</b> To ease organization of the pattern information and to increase the understanding of the pattern.</p>		
<b>Type:</b> <i>Mandatory</i>	<b>Validation criteria:</b> <i>Inspection</i>	<b>WP Relevance:</b> 4
<b>Date:</b> 27/05/10		<b>Version:</b> 1.0

**7.1.2.4 Domain and Role Based Pattern Searching**

<b>REQ-RS-015</b>		<b>Domain and role based pattern searching</b>
<p><b>Description:</b> The patterns may have different ways of implementation, or their use may be focused on certain phases of application design where only a few roles can work with this. This information can be used to ease the search for patterns in the repository. The pattern search mechanism of the process should take into account the application domain where the pattern is used and the actors to which the pattern is in focus.</p>		
<p><b>Source:</b> General</p>		
<p><b>Reference process:</b> Repository Structure</p>		
<p><b>Rational:</b> To ease the search of the pattern in the repository.</p>		
<b>Type:</b> <i>Optional</i>	<b>Validation criteria:</b> <i>Inspection</i>	<b>WP Relevance:</b> 4
<b>Date:</b> 27/05/10		<b>Version:</b> 1.0

### 7.1.2.5 Support Dependencies between Patterns

<b>REQ-RS-016</b>		<b>Support dependencies between patterns</b>	
<p><b>Description:</b> In order to reduce the pattern development time and to reduce the source of errors in the pattern definition, the repository must provide functionalities that allow the pattern designer to create new dependencies between patterns and to easy the interaction with the existent ones.</p>			
<p><b>Source:</b> General</p>			
<p><b>Reference process:</b> Repository Structure</p>			
<p><b>Rational:</b> To ease the definition of new patterns.</p>			
<p><b>Type:</b> <i>Optional</i></p>		<p><b>Validation criteria:</b> <i>Inspection</i></p>	<p><b>WP Relevance:</b> 4</p>
<p><b>Date:</b> 27/05/10</p>		<p><b>Version:</b> 1.0</p>	

### 7.1.2.6 Guideline for Defining Pattern Dependencies

<b>REQ-RS-017</b>		<b>Guideline for defining pattern dependencies</b>	
<p><b>Description:</b> Generating/ Adding a pattern in the repository sometimes requires the use of an existing pattern stored in the repository. For that reason new dependencies have to be generated to establish the relation between these patterns. The repository must provide a guideline to define the new dependencies and the way to add them in the repository.</p>			
<p><b>Source:</b> General</p>			
<p><b>Reference process:</b> Repository Structure</p>			
<p><b>Rational:</b> To ease the definition of new dependencies in the repository.</p>			
<p><b>Type:</b> <i>Optional</i></p>		<p><b>Validation criteria:</b> <i>Inspection</i></p>	<p><b>WP Relevance:</b> 4</p>
<p><b>Date:</b> 27/05/10</p>		<p><b>Version:</b> 1.0</p>	

### 7.1.2.7 Support the Visualisation of the Pattern Dependencies

<b>REQ-RS-019</b>		<b>Visualisation of the pattern dependencies</b>
<b>Description:</b> The repository must provide the possibility to visualise a graph of pattern dependencies.		
<b>Source:</b> General		
<b>Reference process:</b> Repository Structure		
<b>Rational:</b> To ease the user to know the dependencies that exist between a given pattern and all the other patterns which are dependent from itself.		
<b>Type:</b> <i>Optional</i>	<b>Validation criteria:</b> <i>Inspection</i>	<b>WP Relevance:</b> 4
<b>Date:</b> 28/10/10		<b>Version:</b> 1.0

### 7.1.3 Transformation

#### 7.1.3.1 Adaptation of the visualisation according to the user role

<b>REQ-RS-021</b>		<b>Adaptation of the visualisation according to the user role</b>
<b>Description:</b> In the repository, a pattern can be viewed in accordance the concern of the pattern user. The visualisation of the pattern should be adapted according to the user role. A tool must implement this function.		
<b>Source:</b> General		
<b>Reference process:</b> Repository Structure		
<b>Rational:</b> To ease the selection and understanding of the pattern.		
<b>Type:</b> <i>Optional</i>	<b>Validation criteria:</b> <i>Inspection</i>	<b>WP Relevance:</b> 4
<b>Date:</b> 28/10/10		<b>Version:</b> 1.0

### 7.1.3.2 Data Structure to Implement the Repository

<b>REQ-RS-022</b>		<b>Data structure to implement the Repository</b>
<b>Description:</b> The data structure that implement the repository internal structure must be chosen by taking into account the following elements: Visualisation – Modules of interaction – Repository Internal Structure		
<b>Source:</b> General		
<b>Reference process:</b> Repository Structure		
<b>Rational:</b> To identify the main elements to consider in the definition of the repository internal structure and subsequently to ease the search of the pattern in the repository.		
<b>Type:</b> <i>Optional</i>	<b>Validation criteria:</b> <i>Inspection</i>	<b>WP Relevance:</b> 4
<b>Date:</b> 28/10/10		<b>Version:</b> 1.0

## 7.2 Modelling Level

### 7.2.1 Editable Artefacts

#### 7.2.1.1 Edit Repository Search Keys

<b>REQ-RS-023</b>		<b>Edit Repository Search Keys</b>
<b>Description:</b> The repository must support the definition of keywords that support the search for a specific pattern.		
<b>Source:</b> General		
<b>Reference process:</b> Repository structure		
<b>Rational:</b> To ease the search for patterns.		
<b>Type:</b> <i>Mandatory</i>	<b>Validation criteria:</b> <i>Inspection</i>	<b>WP Relevance:</b> 2
<b>Date:</b> 27/05/10		<b>Version:</b> 1.0

**7.2.1.2 Edit Visualisation of a Pattern**

<b>REQ-RS-024</b>		<b>Edit Visualisation of a pattern</b>	
<b>Description:</b> The repository must support the definition of artefacts that can help in the visualisation of the pattern.			
<b>Source:</b> General			
<b>Reference process:</b> Repository structure			
<b>Rational:</b> To ease the visualisation of the patterns.			
<b>Type:</b> <i>Mandatory</i>	<b>Validation criteria:</b> <i>Inspection</i>	<b>WP Relevance:</b> 2	
<b>Date:</b> 28/10/10		<b>Version:</b> 1.0	

**7.2.1.3 Edit Visualisation of a Pattern Dependencies**

<b>REQ-RS-025</b>		<b>Edit Visualisation of the pattern dependencies</b>	
<b>Description:</b> The repository must support the definition of artefacts that can help in the visualisation of the pattern dependencies.			
<b>Source:</b> General			
<b>Reference process:</b> Repository structure			
<b>Rational:</b> To ease the visualisation of the pattern dependencies.			
<b>Type:</b> <i>Mandatory</i>	<b>Validation criteria:</b> <i>Inspection</i>	<b>WP Relevance:</b> 2	
<b>Date:</b> 28/10/10		<b>Version:</b> 1.0	

### 7.2.1.4 Edit User Role View

<b>REQ-RS-026</b>		<b>Edit User Role View</b>	
<b>Description:</b> The user role view can be modified.			
<b>Source:</b> General			
<b>Reference process:</b> Repository structure			
<b>Rational:</b> To adapt the visualisation to new needs.			
<b>Type:</b> <i>Mandatory</i>	<b>Validation criteria:</b> <i>Inspection</i>	<b>WP Relevance:</b> 2	
<b>Date:</b> 28/10/10		<b>Version:</b> 1.0	

## 7.2.2 Editable Models

### 7.2.2.1 Support Pattern Search Function According to Level

<b>REQ-RS-027</b>		<b>Support Pattern Search Function according to level</b>	
<b>Description:</b> In some situations, it is necessary to look for a pattern already defined in the repository in order to carry out an action on it, e.g. modification/correction on an existing pattern. The process must provide a function to search the pattern in the repository according to a level.			
<b>Source:</b> General			
<b>Reference process:</b> Repository structure			
<b>Rational:</b> To ease the search of the patterns.			
<b>Type:</b> <i>Mandatory</i>	<b>Validation criteria:</b> <i>Inspection</i>	<b>WP Relevance:</b> 2	
<b>Date:</b> 27/05/10		<b>Version:</b> 1.0	



**7.2.2.2 Support Pattern Search Function According to Role**

<b>REQ-RS-028</b>		<b>Support Pattern Search Function according to Role</b>	
<b>Description:</b> In some situations, it is necessary to look for a pattern already defined in the repository in order to carry out an action on it, e.g. modification/correction on an existing pattern. The process must provide a function to search the pattern in the repository according to a role.			
<b>Source:</b> General			
<b>Reference process:</b> Repository structure			
<b>Rational:</b> To ease the search of the patterns.			
<b>Type:</b> <i>Mandatory</i>	<b>Validation criteria:</b> <i>Inspection</i>	<b>WP Relevance:</b> 2	
<b>Date:</b> 27/05/10		<b>Version:</b> 1.0	

**7.2.2.3 Edit the Functionality Relating the Visualisation of a Pattern**

<b>REQ-RS-029</b>		<b>Edit the functionality relating the visualisation of a pattern</b>	
<b>Description:</b> The pattern user can use the functionality relating the visualisation of a pattern in order to understand its meaning. In addition, this functionality can be used in collaboration with the functionality concerning the pattern search. Indeed, depending of the user analyse, he can decide to keep this pattern or to search another (in the case where the present pattern does not meet his requirement).			
<b>Source:</b> General			
<b>Reference process:</b> Repository Structure			
<b>Rational:</b> To ease the selection of a pattern.			
<b>Type:</b> <i>Mandatory</i>	<b>Validation criteria:</b> <i>Inspection</i>	<b>WP Relevance:</b> 4	
<b>Date:</b> 28/10/10		<b>Version:</b> 1.0	

### 7.2.2.4 Edit the Functionality Relating the Visualisation of the Pattern Dependencies

<b>REQ-RS-030</b>		<b>Edit the functionality relating the visualisation of the pattern dependencies</b>
<b>Description:</b> The pattern user can use the functionality relating the visualisation of the pattern dependencies in order to see a part of the complexity of the pattern (complexity in term of dependencies). Once the pattern user knows this complexity, he can decide to use the pattern or to search another one.		
<b>Source:</b> General		
<b>Reference process:</b> Repository Structure		
<b>Rational:</b> To ease the selection of a pattern.		
<b>Type:</b> <i>Mandatory</i>	<b>Validation criteria:</b> <i>Inspection</i>	<b>WP Relevance:</b> 2
<b>Date:</b> 28/10/10		<b>Version:</b> 1.0

### 7.2.3 Editable Transformations

#### 7.2.3.1 Edit the Data Structure to Implement the Repository

<b>REQ-RS-033</b>		<b>Edit data structure to implement the Repository</b>
<b>Description:</b> The repository must support the definition of artefacts that can help to create the repository implementation		
<b>Source:</b> General		
<b>Reference process:</b> Repository structure		
<b>Rational:</b> To ease the repository implementation		
<b>Type:</b> <i>Mandatory</i>	<b>Validation criteria:</b> <i>Inspection</i>	<b>WP Relevance:</b> 2
<b>Date:</b> 28/10/10		<b>Version:</b> 1.0

## 8 Conclusion

This document represents the deliverables D2.2, D3.1 & D4.1 which address requirements. The list of the requirements and their description represents our current understanding of the TERESA scope. This document will be modified during the project in order to meet our new needs.

This document represents an entry point for other tasks and deliverables such as D3.2 and D4.2. A state of the art on related work will be done in the next deliverables in order to highlight the TERESA contributions and how the requirements of the present document will be take into account and use in the TERESA project.

## 9 Bibliography

1	OMG, "A UML profile for MARTE: Modelling and analysis of real-time embedded systems," pct/2008-06-09, vol. beta 2, p. 676 pages, June 2008.
2	F. Buschmann, K. Henney, and D. Schmidt, "Pattern-oriented software architecture: On patterns and pattern languages," John Wiley & Sons, p. 490 pages, 2007.
3	P. E. Verissimo, N. F. Neves, and M. P. Correia. "Intrusion-Tolerant Architectures: Concepts and Design". Architecting Dependable Systems, pages vol.2 677. pp 3–36, 2003.
4	D. Powell and R. Stroud. "Conceptual model and architecture of MAFIA". Project IST-1999-11583. Deliverable D 21, January 2003.
5	A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr. "Basic Concepts and Taxonomy of Dependable and Secure Computing". IEEE Transactions on Dependable and Secure Computing, Vol.1, pp. 11–33., january-march 2004.
6	J. L. Fernandez, R. Alonzo, F. Gómez, C. Jouvray, Y. Rouxel, and A. Pérez. "Requirements Engineering of Trusted Embedded Systems". INCOSE, 2010.
7	A. Pfitzmann and M. Hansen. "A terminology for talking about privacy by data minimization: Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management". Apr. 2010. v0.33.
8	S. Borgo and L. Vieu. Artifacts in formal ontology. In A. MEIJERS (ed.), "Handbook of the Philosophy of the Technical Sciences", volume 2: Artifact ontology and artefact epistemology, Elsevier. 2008.