eDIANA
Embedded Systems for Energy Efficient Buildings

Grant agreement no.: *100012*

Requirements Management Plan

Author(s): Huascar Espinoza ESI

Issue Date	January 2010 (m012)
Deliverable Number	D1.2-D
WP Number	WP1
Status	Delivered

Dissemination level	
X	**PU** = Public
	PP = Restricted to other programme participants (including the JU)
	RE = Restricted to a group specified by the consortium (including the JU)
	CO = Confidential, only for members of the consortium (including the JU)

Document history			
V	Date	Author	Description
0.1	*23/10/2009*	*ESI*	*ToC definition*
0.2	*03/11/2009*	*ESI*	*First Draft*
1.0	*29/01/2010*	*ESI*	*Final version*

Disclaimer

The information in this document is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

The document reflects only the author's views and the Community is not liable for any use that may be made of the information contained therein.

Summary

The eDIANA Requirements Management Plan is a public document delivered in the context of WP1, task 1.2 with regard to the definition of a global plan to gather, specify, maintain and validate requirements related to the eDIANA Platform. This document defines a common set of requirements artefacts including documents, requirement types, requirement attributes and the traceability links between them.

Target Readers: *eDIANA Requirements Contributors*

Contents

Abbreviations

eDIANA	Embedded Systems for Energy Efficient Buildings

Table of Figures

Table of Tables

1. Introduction

This Requirement Management Plan describes the main decisions to manage requirements in the eDIANA project. This document specifies how requirements are created, organized, modified, and traced during the project lifecycle. It also describes all requirement types and their attributes used in the project.

The eDIANA project has defined a set of deliverables in WP1 (System requirements, scenario modelling and specification for energy efficiency). These deliverables provide a basis for managing requirements in eDIANA. However, to formalize these artefacts in the context of a requirements engineering process, as intended in Task 1.2 (Requirements Engineering on Energy Management Applications), we should define a detailed plan describing the main decisions on the structure of these deliverables, as well as the connection between them, to provide a coherent set of requirements driving the eDIANA architecture design. This document aims at providing this common plan.

This plan particularly focus in providing a coherent description of requirements types, their attributes and interdependency, as well as the relationship between the different artefacts (WP1 deliverables) proposed in eDIANA.

The document is organized in two main sections. The first section describes the global strategy to organize requirements and documents, and the second section details the proposed requirement artefacts and the requirement engineering process.

2. Requirements Management Strategy

This section describes the general decisions to manage requirements in eDIANA. The purpose of this section is to record the strategy for Requirements Management, including decisions on the related eDIANA deliverables, the required organisational structure and the needed tooling. This strategy consists among others of the definition of a number requirements types en a description how management based on attributes is realised.

2.1 Approach

Depending on the format, source, and project phase, the requirements can be split into different requirement artefacts. Examples of requirement artefacts are stakeholder needs, use cases and scenarios, functional and non-functional requirements.

As a first organization level, we propose to present these artefacts in the form of a pyramid, as shown in Figure 1.

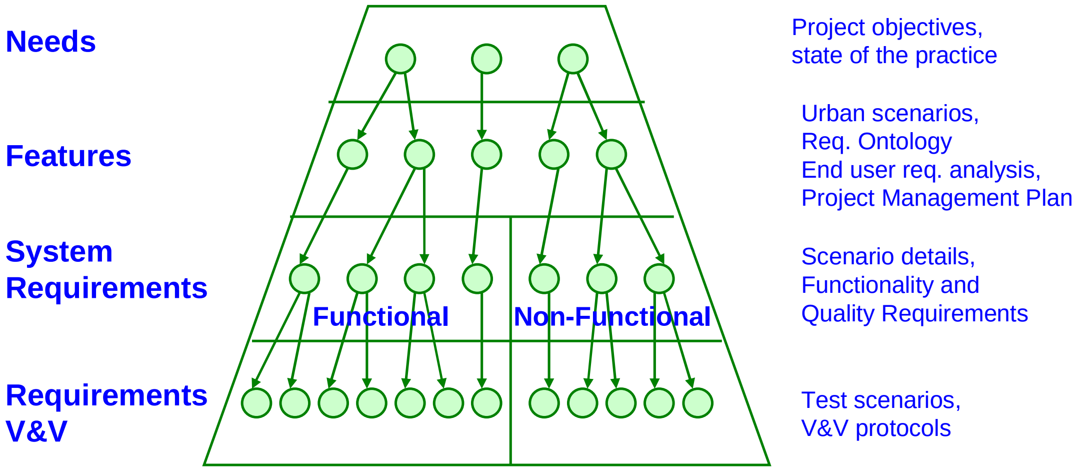
Needs
Features
System Requirements
Requirements V&V
Functional
Non-Functional
Project objectives, state of the practice
Urban scenarios, Req. Ontology End user req. analysis, Project Management Plan
Scenario details, Functionality and Quality Requirements
Test scenarios, V&V protocols

Figure 1 – eDIANA Requirements Pyramid

At the top level are stakeholder needs. On the lower levels are features, and system requirements. Quite often, at different levels of these requirements, different levels of detail are captured. The lower the level, the more detailed the requirement. However, it is up to each eDIANA responsible of deliverable to decide on the granularity of requirements at its level. Nothing is wrong with placing quite detailed requirements from stakeholders on the stakeholder needs level.

The main difference between needs and features is in the source of the requirement. Needs come from the eDIANA project objectives and the state of the art, and features are formulated by analysing the high-level usage scenarios and services provided by the eDIANA platform. Such scenarios can be presented as a sequence of steps to use the eDIANA platform. In the next level, system requirements refine features and classify them into functional and non-functional requirements. Functional requirements can be expressed as use cases that hold low-level scenarios. These low-level scenarios describe the flow of events for using a given functionality. At the bottom, the test cases and V&V protocols check if system requirements are implemented correctly.

We can also define many other requirement types, such as glossary terms and the present requirements management plan itself. They are not pure requirements conforming to the definition of requirement, but they are requirement artefacts that can help to manage requirements.

Table 1 summarizes the concrete artefacts the corresponding eDIANA deliverables associated to them, and the requirement types used in the different phases.

Table 1 – Requirement Artefacts and their Relation to eDIANA Deliverables

Levels	Artefacts	eDIANA Reference	Requirement Types
Needs	1. Preliminary Project Needs	Objectives from Technical Proposal	PJREQ – Project Requirements
	2. Baseline Analysis Result	D1.1-A	
	3. State of the Practice Report On Urban Environment Requirements	D1.2-A	
Features	4. Requirements Support Ontology	D1.2-B	FEAT – Features
	5. Customer and End User Requirements Analysis and Eval.	D.1.3-A,B,C	
	6. Reference (Urban) Scenarios (Release 1)	D1.4-A	
	7. Requirements Management Plan	D1.2-C *(Change Request)*	
System Requirements	8. Reference (Detailed) Scenario (Release 2)	D1.4-B	FUNCR – Functional Requirement NFUNCR – Functional Requirement
	9. Functional Requirements	D1.2-D *(Change Request)*	
	10. Non-Functional Requirements		
Requirements V&V	11. Questionnaires for V&V	Internal Document	REQVV – Requirement V&V
	12. V&V protocols for energy-aware artefacts	D6.2-C	
	13. Test Scenarios to validate Requirements	D6.2-B	

Requirements management is an interactive process. In a typical iteration, a full pass through the pyramid is performed. Even in the same iteration, we can go back a few steps and repeat the activity. For example, during the creation of functional and non-functional requirements, we can discover that some information is missing in the customer and end user requirements analysis, and we need more input from an eDIANA stakeholder, so we go back a step to enrich artefact 5. This will be reflected in the different deliverables of artefact 5 (Artefact 5: D.1.3-A, B and C).

For a detailed description of the Requirement Types, please see Section 3.

2.2 Tools and Infrastructure

Using a requirements management tool significantly facilitates the creation and maintenance of requirements. Usually tools provide traceability tracking. Various reports can easily be produced. These benefits come from the fact that this kind of tools is supported in a database that allow for filtering information and link database items with other kind of documents such as text processors.

In eDIANA, we decided to use RequisitePro, a tool from IBM Rational that has been provided by ATOS in its web version (RequisiteWeb).

RequisitePro, and hence RequisiteWeb (from IBM Rational), is a tool that facilitates requirements management. It allows input, updates, tracking, and review of requirements during the project lifecycle. You can create, organize, prioritize, and trace the requirements. RequisiteWeb integrates Microsoft Word and a database infrastructure. The tool allows detailed customization of documents, requirement types, and attributes, by means of project templates. Change management is facilitated by tracking traceability between the requirements.

A user manual (D1.2-C) has been provided for eDIANA users in order to ease its use by the project members.

3. Requirements Management Program

This section provides a detailed description of the requirement types, artefacts and the process to specify, analyse and evaluate requirements in eDIANA.

3.1 eDIANA Requirements Template

This section describes the requirement types and their attributes used in the eDIANA project. These are specified in what we call eDIANA Requirement Template. The template has been defined in the RequisiteWeb.

Table 2 and Table 3 summarize the attributes for the two types of requirements defined in eDIANA.

Table 2 – Attributes of the Functional Requirement Type

Attribute Name	**Type**	**List values**	**Default**
Priority	List	High Medium Low	Medium
Status	List	Initial Defined Authorized Developed Delivered Closed Cancelled	Initial
Criticity	List	Mandatory Important Useful Optional	Useful
eDIANA Level	List	Cell Macro Cell Whole eDIANA Platform	n/a
eDIANA Component	List	Macro Cell-Level Concentrator Macro-Cell Data Gathering Macro-Cell Contr. Strate. Manag. Cell Monitoring and Metering Cell Control and Actuation Cell User Interface Channels Cell Generation and Storage Cell-Level Concentrator	n/a
Category	List	COM – Communication FUN - Functions/Operations IS - Information Storage/Flow UI - User Interface DI - Device Interface MOD - Operational Mode	n/a
Information Direction	List	Required Provided Shared Undefined	n/a
Subtypes	List	Use Case Use Case Details	n/a
User Types	MultiSelect	End User Maintenance Macro-Cell Administrator	n/a
Affected Devices or Elements	Text		n/a
Traceability Nature Description	Text		n/a

Table 3 – Attributes of the Non-Functional Requirement Type

Label	**Type**	**List values**	**Default**
Priority	List	High Medium Low	Medium
Status	List	Initial Defined Authorized Developed Delivered Closed Cancelled	Initial
Criticity	List	Mandatory Important Useful Optional	Useful
eDIANA Level	List	Cell Macro Cell Whole eDIANA Platform	n/a
eDIANA Component	List	Macro Cell-Level Concentrator Macro-Cell Data Gathering Macro-Cell Contr. Strategy Man. Cell Monitoring and Metering Cell Control and Actuation Cell User Interface Channels Cell Generation and Storage Cell-Level Coordinator	n/a
Related Functional Category	List	FUN - Functions/Operations COM – Communication IS - Information Storage/Flow UI - User Interface DI - Device Interface MOD - Operational Mode	n/a
Category	List	Cost/Price Design Constraint Memory Storage Performance Physical Power Consumption Reliability Safety Security Standard Compliance Usability	n/a
Affected Devices or Elements	Text		n/a
Traceability Nature Description	Text		n/a
Quantifier	Text		n/a

The meaning of the two requirement types is the following:

a. **Functional Requirements**: A functional requirement defines a function of a system or their components. Functional requirements may be calculations, data manipulation and processing, communication functions, and other specific functionality that define what a system is supposed to accomplish.

b. **Non Functional Requirements**: A non-functional requirement is more concerned with specific criteria that can be used to judge the operation of a system. In the eDIANA context, this includes requirements such as performance, reliability, safety, power consumption, as well as other more generic aspects such as design constraints (e.g., technology, architecture) and compliance with standards.

Most of the requirement attributes are quite self-defined and others were originated from deliverable D1.2-B (Requirements Support Ontology). However some of them are somehow vague and need some clarifications:

Priority

- *High*: Must be implemented no later than in the first iteration of the Construction phase.
- *Medium*: Must be implemented no later than by the end of Construction.
- *Low*: May be implemented if time permits.

(Functional) Category

- *COM – Communication*: functionality related to communication between separated nodes.
- *FUN - Functions/Operations*: functionality covering computation/processing operations.
- *IS - Information Storage/Flow*: functionality related to information storage or transmission.
- *UI - User Interface:* functionality associated to user interfaces.
- *DI - Device Interface:* functionality related to interfaces of devices.
- *MOD - Operational Mode:* description of required operational modes. An operational mode identifies a segment within the system execution that is characterized by a given configuration.

Information Direction: Only applicable if the functional category is *Information Storage/Flow*.

- *Required:* the information is required by the component defined in the attribute *eDIANA Component*.
- *Provided:* the information is offered by the component defined in the attribute *eDIANA Component*.
- *Shared:* the information is shared (read/write) by multiple eDIANA components.
- *Undefined:* the information direction exists, but it is not defined at this stage of the requirement elicitation.

Functional Subtypes

- *Use Case:* The functional requirement is a Use Case. The use case may be documented by means of an attached Use Case document.
- *Use Case Details:* The functional requirement is a Use Case. The use case may be documented by means of an attached Use Case Details document.

User Types

- *End User:* This is a resident of a building.
- *Maintenance:* This is an actor that has access for setting and managing the whole system configuration.
- *Macro-Cell Administrator:* This is an actor that can manage the settings and system operation at macro-cell level.

Affected Devices or Elements: It describes a list the specific component, device, or element that holds the related requirement. The elements must be separated by a comma.

Traceability Nature Description: When having traceability relationships, this requirement attribute describes the meaning of the traceability link (e.g., how changes in one requirement affect on its dependent requirements).

Non-Functional Category

- *Cost/Price:* the requirement is related to the cost or price of an eDIANA component.
- *Design Constraint:* the requirement is related to constraints in the system design.
- *Memory Storage:* constraints in memory size.
- *Performance:* the requirement is a timing constraint.
- *Physical:* the requirement is related to a physical element on the environment of the eDIANA platform.
- *Power Consumption:* the requirement identifies a power consumption constraint.
- *Reliability:* any requirement concerning reliability such as for example MTBF (mean time between faults).
- *Safety:* any requirement concerning safety such as SIL.
- *Security:* requirement related to system security.
- *Standard Compliance: requirements of standards conformance.*
- *Usability:* any requirement concerning the easy of use of the system or its components.

Quantifier

This attribute makes possible to assign a quantifier and its measurement unit (if necessary) to the requirement. The field is text/string to simplify the definition of any kind of quantifier; the attribute can indicate the units (ºC, seconds, watts, etc.) if applicable.

3.2 Eliciting Needs

Most of eDIANA requirements come from project partners. For the purpose of requirements elicitation, we call these actors "stakeholders". An eDIANA stakeholder is an individual who is materially affected by the outcome of the project. EDIANA stakeholders act as both customers and users. Task T1.3 in eDIANA is in charge of defining the customer and user needs.

Besides customers and users, other stakeholders that provide input to the eDIANA requirements specification is people who will maintain the system, anyone participating in developing the eDIANA platform (WP2, W3, WP4, WP5), domain experts, people that will validate the system and the system requirements (WP6-V&V), providers of rules and regulations (WP6-Certification), other related projects in this domain.

During the eDIANA project these different stakeholder will provide the information for the activity of requirements elicitation, by using also specific elicitation techniques.

Some of the techniques to gather requirements that will be used in eDIANA are:

Interviews: individual dialog with experts (e,g, WP6 will gather the V&V needs in industry by performing interviews).

- **Questionnaires:** when the required information may cover a large technical domain, it may be useful to use well detailed questionnaires.
- **Workshops:** perhaps one of the most used techniques used in collaborative projects like eDIANA. During workshops, partners gather for an intensive, focused period.
- **Storyboarding:** in particular, scenarios of T1.4 must be collected as graphical storyboards describing the interaction of users with eDIANA devices, or eDIANA system behaviour in general.
- **Use cases:** interaction of users or other kind of actors with the eDIANA platform. Task T6.2 will provide the modelling basis to uses use case diagrams in order to gather functional requirements.
- **Analysis of existing documents or systems:** it is recommendable to use existing information of the state of the art and state of the practice projects. This technique will be used during the technical specification of the eDIANA component requirements.

One important aspect to take into account in requirements engineering, and this is not an exception in eDIANA, is the fact that requirements are gathered during the entire project. Having a requirements management tool (RequistePro in our case) helps to manage the process during the whole project life, in particular with a proper requirements change management.

3.3 Requirements Change Management

Traceability is the main feature of a good requirements change management process. Traceability helps analyze the impact on individual requirement change.

Traceability is a technique that provides a relationship between different levels of requirements in the system. This technique helps determine the origin of any requirement. Figure 1 illustrates how requirements are traced from the top level down in eDIANA. Every need usually maps to some features. Generally, it is a many-to-many relationship because one need can trace to many features, but one feature may be derived from many needs.

Every functional requirement maps to one or more scenarios, so a one-to-many relationship exists between functional requirements and scenarios. Scenarios map to test cases in a one-to-many relationship.

This traceability needs to be reflected during the different eDIANA tasks gathering requirements. Customer and user needs requirements specified in deliverables D1.3-A, B and C must have their counterparts into the list of functional and non-functional requirements define din T1.4. In the same way, the dependency between requirements of the same level must be traced to explicitly define

Traceability plays several important roles:

- Verifying that an implementation fulfils all requirements. In other words, we can verify that the customer requested was implemented.
- Verifying that the application does only what was requested. That means that we do not implement something that the customer never asked for.
- Impact analysis. What elements will be affected when we consider adding a new requirement or changing an existing one?
- Helping with change management: When some requirements change, we want to know which test cases should be redone to test this change.

A traceability item is a project element that needs to be traced from another element. In terms of RequisitePro, it is everything that is represented by an instance of the requirement type. Some examples of requirement types in RequisitePro are stakeholder needs, features, scenarios, actors, and glossary terms.

For an operational guide of traceability management, refer to deliverable D1.2-C.

3.4 Analysing Requirements

In order to refine requirements, there should be a careful process of requirements analysis, from needs to detailed scenarios and technical non-functional requirements.

There are some key activities that require special attention:

3.4.1 Scenario Refinement

The purpose of a scenario is to facilitate agreement between eDIANA architects and developers with stakeholders. Scenarios are refined from high level scenarios representing generic interactions between users or other system actors with the eDIANA platform.

High level scenarios must start by identifying initiating actors and by defining the basic sequence of interaction with the eDIANA platform.

Scenarios must be created properly from early phases. Hence, actor identification and interaction storyboards must be as complete as possible. It must be noted that an actor is someone or something that interacts with the system. This means that in addition to users, administrators, etc. external systems providing or requiring information from the eDIANA platform must be identified as actors.

On the other hand, to properly identify useful scenarios, the following questions should be answered:

- What functionality does each actor expect from the eDIANA platform?
- Do the actors need to be informed about the events occurring in the systems?
- What information the actors need to supply/receive to/from the system?

When refining scenarios, it is useful to split the interactions with the system in logical sequences of events. This may allow assigning detailed scenarios to specific parts of the eDIANA platform or to specific components. In the same way, optional or conditional paths within a sequence of events can be divided into separated detailed scenarios. In general the criteria should be to assign detailed scenarios coming from task T2.5 to other task of the eDIANA project dealing with concrete eDIANA aspects.

3.4.2 Analysing Non-Functional Requirements

In order to facilitate the specification of non-functional requirements, the eDIANA Template for Requirements provides a list of categories of non-functional aspects. These categories have been carefully chosen to match with eDIANA needs. Hence, we will find categories such as performance, safety, power consumption, cost, design constraints, etc. For instance, performance requirements are concerned with temporal aspects such as response times, throughput, recovery time, etc.

Many of the features specified in the customer and client requirements can become non-functional requirements. The third level of the pyramid (starting at the top) provides an opportunity to add more details and organize the requirements by inserting them in the appropriate eDIANA component. One approach to refine requirements is to go through all the features, identify which ones were not addressed in scenarios, and translate them into non-functional requirements.

3.5 Specifying System Requirements

In the eDIANA approach, we provide two options to specify requirements: either by using Word documents or using individual items of a RequisitePro project database. For tracking requirements attributes and traceability, it is not necessary to have them in documents. They may just reside in the database. However, also having them in documents offers some advantages:

- Easier access to the requirements by team members who do not have access to RequisitePro
- Opportunity to visually group and organize the requirements
- Presenting them in a more readable form
- Easy to add comments and explanations

An alternative to managing requirements in documents is the use of filters and queries in RequisitePro, with an organization scheme similar to that of the corresponding RequisitePro template.

Importing documents in RequisiteWeb can be done either by using predefined document formats (eDIANA Word Templates stored by default in the RequisitePro project) or by using the original document format. This feature allows for importing existing eDIANA deliverables without worring on formatting issues.

Additionally, some imported documents might already have requirement bookmarks (a requirement bookmark is a RequisiteWeb mark in the Word document pointing to requirements stored in a RequisteWeb database). To avoid encountering errors in a document that contains requirement bookmarks, you must select the option to remove the bookmarks while importing the document.

The following requirement types are usually stored in corresponding documents, not just in the database:

- Because of their descriptive nature, scenarios should be associated with the documents— one document per scenario.
- Features are included in the deliverables of T1.3.
- Needs documents (T1.1) can be better explained with diagrams and examples.

3.6 Validating Requirements

This section does not describe in detail how to create functional test cases from scenarios. Deliverables of task 6.2 will provide a close description of test cases.

In eDIANA, the importance of this step is significant. Often the testers are given a printout of a scenario specification and then perform ad hoc manual testing. However, if we neglect formal creation of test cases in eDIANA, we may end up with poor universality of the eDIANA platform.

Just as general information, we can mention that four steps are involved in test case creation:

1. Identify variables for each scenario step (input information, multiple selection options, etc.)
2. Identify significantly different options for each variable. Options are significantly different if they may trigger different system behaviour. These different options will trigger different processing flows.
3. Combine options to be tested into test cases. They must be combined in the sequence of test case steps. A technique is to use matrix mapping variables and test cases. For each variable, we must enter all the options that need to be tested for this variable.
4. Assign values to variables. This helps find any gaps in requirements early in the process

More details on test cases will be done in deliverables of T6.2.

Acknowledgements

The eDIANA Consortium would like to acknowledge the financial support of the European Commission and National Public Authorities from Spain, Netherlands, Germany, Finland and Italy under the ARTEMIS Joint Technology Initiative.

References

- Peter Zielczynski. Requirements Management Using IBM® Rational® RequisitePro. IBM Press. 2008.