openETCS/WP4/D4.1

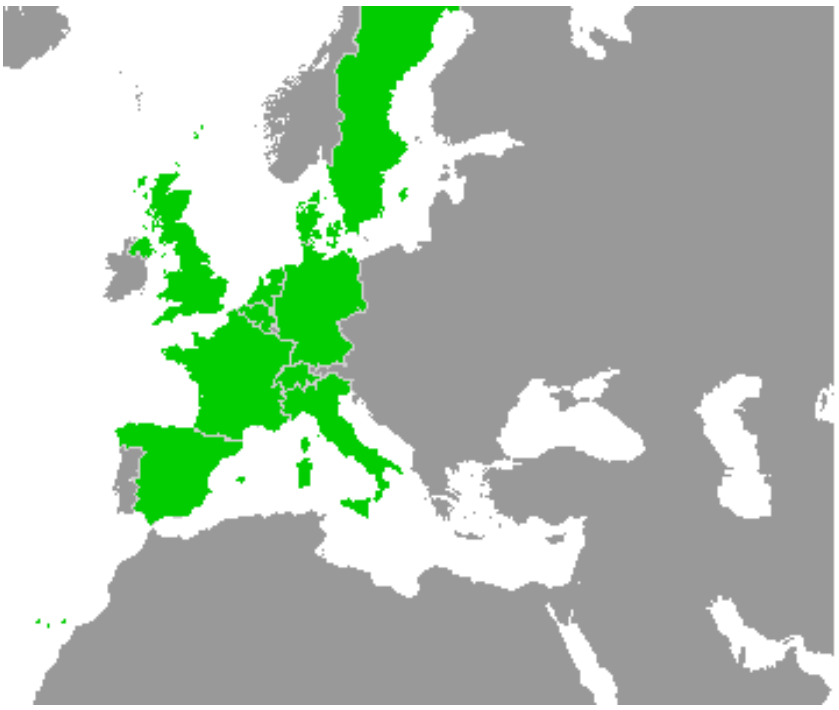
ITEA2 Project
2012 – 2015

Work-Package 4: "Requirements for Open Proofs"

openETCS D4.1: Test Cases Definition – Draft

The first draft of definition of test cases

Huu-Nghia Nguyen July 2014



openETCS

This page is intentionally left blank

openETCS D4.1: Test Cases Definition – Draft

The first draft of definition of test cases

Huu-Nghia Nguyen

Institut Telecom

Preliminary Report

Prepared for ITEA2 openETCS consortium
Europa

Abstract: This report presents the first draft of definition of test cases. Particularly, it defines a format of test cases to be later run against API/ Demonstrator.

Table of Contents

Figures and Tables

Figures

Tables

1 Introduction

The objective of this document is to provide a format to describe test cases to be later run against API/Demonstrator.

Let's start with the basic definitions. What is a test case?

- IEEE Standard Computer Dictionary 610 IEEE [1990] defines a test case as

  > A set of test inputs, execution conditions, and expected results developed for a particular objective, such as to exercise a particular program path or to verify compliance with a specific requirement.

- IEEE Standard 829-2008 IEEE [2008] adopted from the definition above to define a test case as

  > Documentation specifying inputs, predicted results, and a set of execution conditions for a test item.

The definition of test case format proposed in this document is guided by the definitions above.

The rest of the document consists of two chapters. The first chapter describes a template of test case specifications while the second one give a meta model using Ecore of the template.

2 Test Case Specification Template

The purpose of the test cases is to define the information needed as it pertains to inputs to and outputs from the being tested. A test case specification includes all test case(s) identified by the associated segment of the level test design (if there is one). A example of a test case outline is shown below. Section Introduction is specified once per document. The Test Case details in Sections 2 are documented once per Test Case.

1. Introduction (once per document)
    1.1. Document identifier
    1.2. Document change procedures and history
    1.3. Glossary
    1.4. Scope
    1.5. References
    1.6. Context
    1.7. Notation for description
2. Details (once per test case)
    2.1. Test case identifier
    2.2. Objective
    2.3. Inputs
    2.4. Outcome(s)
    2.5. Environmental needs
    2.6. Special procedural requirements
    2.7. Intercase dependencies

2.1 Introduction

Introduce the following subordinate subsections. This section identifies the issuing organization and the details of issuance. It includes required approvals and status (DRAFT/FINAL) of the document. It is here that the scope is described and references identified.

2.1.1 Document identifier

Uniquely identify a version of the document by including information such as the date of issue, the issuing organization, the author(s), and the status/version (e.g., draft, reviewed, corrected, or final).

It contains basically the following information:

- Unique "short" name
- Version date and version number
- Organization
- Author(s)
- Status

2.1.2 Document change procedures and history

Specify the means for identifying, approving, implementing, and recording changes to the test case specification. This may be recorded in an overall configuration management system that is documented in a Configuration Management Plan that is referenced here. The change procedures need to include a log of all of the changes that have occurred since the inception of the MTP. This may include a Document ID (every testing document should have a unique ID connected to the system project), version number (sequential starting with first approved version), description of document changes, reason for changes (e.g., audit comments, team review, system changes), name of person making changes, and role of person to document (e.g., document author, project manager, system owner). This information is commonly put on an early page in the document (after the title page and before Section 1).

2.1.3 Glossary

Provide an alphabetical list of terms that may require definition for the users with their corresponding definitions. This includes acronyms. There may also be a reference to a project glossary.

2.1.4 Scope

Summarize the software product or system items and features to be tested by this particular level of test. The need for each item and its history may be included.

2.1.5 References

List all of the applicable reference documents. The references are separated into “external" references that are imposed external to the project and “internal" references that are imposed from within to the project.

The external references list references to the relevant policies or laws that give rise to the need for this plan, e.g.: a) Laws b) Government regulations c) Standards (e.g., governmental and/or consensus) d) Policies. The reference to this standard includes how and if it has been tailored for this project, an overview of the level(s) of documentation expected, and their contents (or a reference to an organizational standard or document that delineates the expected test documentation details).

The internal references list references to documents such as other plans or task descriptions that supplement this plan, e.g.: a) Project plan b) Quality assurance plan.

2.1.6 Context

Provide any required context that is not already covered by other sections of this document (e.g., third- party party testing via the Internet).

2.1.7 Notion for description

Define any numbering schemes, e.g., for scenarios and test cases. The intent of this section is to explain any such schema.

2.2 Details

2.2.1 Test Case Specification Identifier

Describe the unique identifier needed by each test case so that it can be distinguished from all other test cases.

2.2.2 Objective

Identify the items or features to be tested by this test case. Keep in mind the level for which this test case is written and describe the items/features accordingly. The item description and definition can be referenced from any one of several sources, depending on the level of the test case specification. In such a case, it should reference the source document as well.

2.2.3 Input Specifications

Identify all inputs required to execute the test case. Again these may vary based on the level the case is written for. Be sure to identify all required inputs not just data elements and values. Some inputs will be specified by value (with tolerances where appropriate), whereas others, such as constant tables or transaction files, will be specified by name. Identify all appropriate files, terminal messages, memory resident areas, and values passed by the operating system.

- Data
  - Values
  - Ranges
  - Sets
- Table
- Human actions
- Conditions
  - States
  - Initial
  - Intermediate
  - Final
- Files
  - Control files
  - Transaction files
- Relationships
  - Timing

It is also acceptable to simplify the documentation process by using tables for elements and values. It is even conceivable to create a test case that can be used a multiple levels of testing (Unit, Integration, etc.). Notes can be used to document common rules and processes for elements that have shared characteristics.

Specify all required relationships between inputs (e.g., timing).

2.2.4 Output Specifications

Identify all outputs required to verify the test case. Again these may vary based on the level the case is written for. Be sure to identify all required outputs not just data elements and values.

- Data

  - Values
  - Ranges
  - Sets
- Tables
- Human actions
- Conditions
  - States
  - Initial
  - Intermediate
  - Final
- Files
  - Control files
  - Transaction files
- Relationships
- Timing
  - Response times
  - Duration

Outputs can also be simplified using tables as noted above and may even be included in the same table as the associated input to further simplify the documentation and improve its usefulness.

2.2.5 Environmental needs

Describe the test environment needed for test setup, execution, and results recording. This section is commonly documented per scenario or group of scenarios. It may be illustrated with one or more figures showing all of the components and where they interact.

- Hardware
  - Configurations
  - Limitations
- Software
  - Operating systems
  - Compilers
  - Tools
  - Other Application
- Other
  - Facilities
  - Training

Hardware. Specify the characteristics and configuration(s) of the hardware required to execute this test case.

Software. Specify all software configuration(s) required to execute this test case. This may include system software such as operating systems, compilers, simulators, and test tools. In addition, the test item may interact with application software.

Other. Specify any other requirements not yet included, e.g., unique facility needs, specially trained personnel,

2.2.6 Special Procedural Requirements

Identify any special constraints on the test case(s). If the test case specification covers more than one test case this may be the common procedure for several sets of tests or there may be more than one set of steps or external procedures identified. Focus on key elements such as:

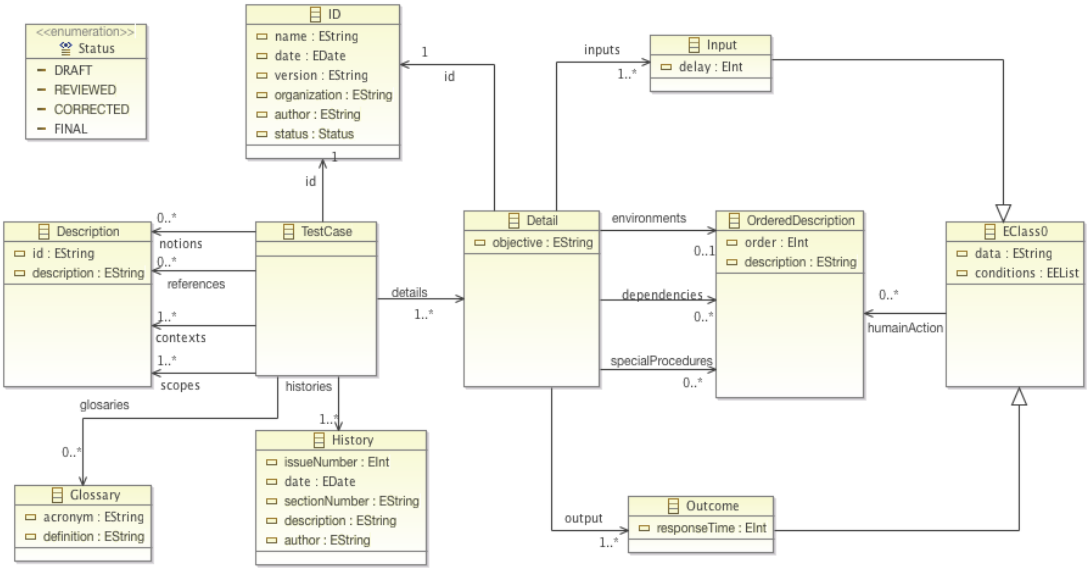
- Special Setup
- Operations intervention
- Output location and identification
- Special wrap-up

2.2.7 Inter-case Dependencies

Identify any prerequisite test cases. It is also recommended that the relationship of test cases be documented at both ends of the relationship. The precursor should identify any follow-on test cases and the post cases identify all prerequisites.

3 Meta Model of Test Cases Specifications

Ecore Diagram using Eclipse Modeling Framework



References

IEEE. *IEEE Standard Computer Dictionary*. 1990. ISBN 1559370793.

IEEE. *IEEE Standard for Software and System Test Documentation*, volume 2008. 2008. ISBN 9780738157467.

