ERTMS Formal Specs Workbench

Coding Guidelines, Version: 0.2

Moritz Dorka (MD)

12/11/2013

Revision History

Revision	Date	Author(s)	Description
0.1	31/10/2013	MD	initial draft
0.2	12/11/2013	MD	add "How to use this file", "Comments", "Glossary", "Bibliography"

Contents

How to read this file

This document lists rules to be followed when coding in the Domain Specific Language (DSL) of the ERTMS Formal Specs Workbench (EFS) tool. It is targeted at both model developers and test-case designers. ...

It will not cover the technical implementation of the underlying DSL or give general instructions on how to use the EFS-tool. For this refer to the Technical design [1] and the user guide [2], respectively.

TODO$_{MD}$

Typographical conventions

Rule texts

Each rule has a unique identifier composed of the current section and a running number. Words to be used verbatim within the rule texts are printed in *emphasized* letters. Rule texts conclude with the initials of the rule author and the date of the last change of the rule.

Code examples

Code of the DSL used for examples within the rules is printed in `typewriter` font with

- language keywords in **`bold`**,

- pre-defined functions <u>`underlined`</u>,

- placeholders *`emphasized`* and

- string literals in a sans-serif font.

For a documentation of the keywords and the pre-defined functions see [1].

1 Domain specific language

1.1 General

1.1.1. TODO$_{MD}$ **EMPTY** vs. **NOT** <u>Available</u>() vs. []

1.1.2. **SUM** *List* **USING X** shall be preferred over **REDUCE** *List* **USING X + RESULT**.

1.1.3. **THERE_IS_IN** *List | Condition* shall be preferred over **MAP** *List | Condition* ▮ **USING X**.

2 Model

2.1 General

2.1.1. All decisions made by the modeler which are not purely cosmetic (i.e. using a Function instead of a Procedure), but have an influence on the actual behavior of the model shall be documented in the section *Design Choices* of the Specification View and linked to the respective code.

<div align="right">MD, 12/11/2013</div>

2.2 Functions

2.2.1 Cases

2.2.1.1 General

2.2.1.1.1. `TBD`*MD* Cases shall only contain trivial expressions (i.e. all logic shall be defined inside pre-conditions).

<div align="right">MD, 31/10/2013</div>

2.2.1.1.2. `TBD`*MD* Complex expressions and pre-conditions shall not be used together.
this rule is mutually exclusive to 2.2.1.1.1.*MD*

<div align="right">MD, 31/10/2013</div>

2.2.1.2 Pre-conditions

2.2.1.2.1. Cases with no pre-condition assigned shall always come as the very last case.

<div align="right">MD, 31/10/2013</div>

2.2.1.2.2. Cases with no pre-condition assigned shall always be named *Otherwise*.

<div align="right">MD, 31/10/2013</div>

3 Tests

3.1 Structure

3.1.1. TBD*MD* A Sub-step shall either contain Actions or Expectations but not both.

MD, 12/11/2013

3.1.2. The first Sub-step of the first step of a test frame shall only contain the expression `Kernel.InitializeTestEnvironment()`.

MD, 31/10/2013

   3.1.2.1. There shall be no Expectation associated with this Sub-step.

MD, 31/10/2013

3.2 Naming conventions

3.2.1. Individual test steps shall be named *Step n - explanation*.
Explanation is a user-defined text to summarize the actions and expectations of the current step.

MD, 31/10/2013

4 Comments

4.1 General

4.1.1. Comments shall be used wherever the inner workings of a code are not directly obvious from the documentation provided by the linked requirements or design choices.

MD, 12/11/2013

4.1.2. Comments shall always be assigned to the narrowest possible scope to which they apply (i.e. to an individual test-step rather than to the entire test).

MD, 12/11/2013

4.1.3. Comments shall not be used to justify Design Choices. See rule 2.1.1. instead.

MD, 12/11/2013

4.1.4. Comments shall always end with *#<github_username>, yy/mm/dd*. Where *<github_username>* is to be replaced with the actual username of the comment-author and *yy/mm/dd* with the current date.

MD, 12/11/2013

5 Git

5.1 General

5.1.1. Own edits shall always be committed into a personal fork of the EFS repository and then transferred by issuing a pull-request.

MD, 31/10/2013

5.2 Commits

5.2.1 General

5.2.1.1. numerous small commits dealing with a single issue are preferable over few huge commits possibly dealing with many different issues.

MD, 31/10/2013

5.2.2 commit description texts

5.2.2.1. The text shall not contain line-breaks (i.e. consist of a single line only).

MD, 31/10/2013

5.2.2.2. Commits dealing with the modeling part of the tool shall begin with *EA_MODEL* followed by a reference to the sections of the subset the commit is related to and a descriptive text.

MD, 31/10/2013

5.2.2.3. Commits dealing with the testing part of the tool shall begin with *EA_TEST* followed by a reference to the sections of the subset the commit is related to and a descriptive text.

MD, 31/10/2013

Glossary

DSL . . . . Domain Specific Language

EFS . . . . ERTMS Formal Specs Workbench

Bibliography

[1] ERTMS Solutions, Brussels. *ERTMSFormalSpec Workbench Technical design*, 2013.

[2] ERTMS Solutions, Brussels. *ERTMSFormalSpec Workbench User Guide*, 2013.

List of Corrections

List of PDF comments