**Data Model for OpenETCS project**

This document contents the list of items to take into account during the modelling activities, how to describe them and how there are linked.

**List of items:**

* Type
* Requirement
* Function
* Functional Block
* Variable
* Constant
* Packet

All the items are identified by:

* an ident: this ident shall be unique and allow to identify exactly the item. How to be defined the ident depend of the tools to manage the repository of items
* a name: the name is defined according naming convention (see first proposal <https://github.com/openETCS/validation/wiki/Verification-Artifacts-Styleguide>) and shall allow to identify the class of items (requirement, variable, function,...)

In the case that uniqueness is ensured on the names, and depending on the tools selected, we can avoid to use “ident”.

**Type:**

* Boolean
* Integer
* Floating point
* T\_text: textual description
* T\_ident: based on a textual description
* T\_Version = {3.0.0, 3.3.0}
* T\_VariableNature = {Acceleration, Distance, Gradient, Length, Miscellaneous, Number, ClassNumber, IdentityNumber, Qualifier, TimeDate, Speed, Text}
* T\_SourceDocument = {subset-26, subset-34}
* T\_Definition : a textual and graphical description (all kind of picture, table and diagram are allowed)

**Requirement: T\_Requirement**

It is identified by the following attributes:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Occurrence** | **Type** | **Description** |
| Definition | 1 | T\_Definition | Textual and graphical description of the requirement |
| Nature | 1 | * Structural * Functional * Definition |  |
| Source | 1 | T\_SourceDocument | the document where the requirement is defined the first time (SRS, FIS, SSRS,...) |
| Discussion | 1 (Optional) | T\_Text | Discussion or comment to clarify or justify the requirement |
|  |  |  |  |

Requirements are links together by mother-child relations:

* a requirement can be refined (description can be clarified) in a child requirement
* a requirement can be split in several child requirements
* a requirement can be derived from several mother requirements

**Variable: T\_Variable**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Occurrence** | **Type** | **Description** |
| Definition | 1 | T\_Definition | Textual and graphical description of the variable |
| Source | 1 | T\_SourceDocument | the document where the variable is defined the first time (SRS, FIS, SSRS,...) |
| Nature | 1 | T\_VariableNature | Nature of the Variable (Acceleration, Speed, Ident,...) |
| MinimalValue | 0..1 | T\_Text |  |
| MaximalValue | 0..1 | T\_Text |  |
| SpecialValue | 0..n | T\_Text |  |
| Allocation | 1 | * Interface * Packet * Internal |  |
| *Requirement* | *1..n* | *T\_Requirement* | *The set of requirement which defined the variable (at least one)* |
|  |  |  |  |

**Constant: T\_Constant**

*Do we need to separate Variables and constants ?*

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Occurrence** | **Type** | **Description** |
| Definition | 1 | T\_Definition | Textual and graphical description of the variable |
| Source | 1 | T\_SourceDocument | the document where the variable is defined the first time (SRS, FIS, SSRS,...) |
| Nature | 1 | T\_VariableNature | Nature of the Variable (Acceleration, Speed, Ident,...) |
| Value | 1 | T\_Text |  |
| Allocation | 1 | * Interface * Packet * Internal |  |
| *Requirement* | *1..n* | *T\_Requirement* | *The set of requirement which defined the constant (at least one)* |
|  |  |  |  |

**Function: T\_Function**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Occurrence** | **Type** | **Description** |
| Definition | 1 | T\_Definition | Textual and graphical description of the function |
| *Input* | *0..n* | *T\_Variable* | *Input variables of the function* |
| *Output* | *0..n* | *T\_Variable* | *Output variables of the function* |
| *Local* | *0..n* | *T\_Variable* | *Main internal variables of the function used to describe it* |
| *Parameter* | *0..n* | *T\_Constant* | *Main constants and parameters used to describe the function* |
| *Requirement* | *0..n* | *T\_Requirement* | *Requirements allocated to the function* |
| *Block* | *1 (optional)* | *T\_FunctionalBlock* |  |
|  |  |  |  |

Functions are linked together by several kind of relation:

* mother-child relation: a function is split in several sub-functions
* sequential relation: a function shall occur before an another

**Functional Block: T\_FunctionalBlock**

*Are functional blocks really necessary ?*

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Occurrence** | **Type** | **Description** |
| Definition | 1 | T\_Definition | Textual and graphical description of the function |
| *Function* | *1..n* | *T\_Function* | *Set of function of this block* |
|  |  |  |  |

**Packet: T\_Packet**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Occurrence** | **Type** | **Description** |
| Definition | 1 | T\_Definition | Textual and graphical description of the packets |
| *Data* | *0..n* | *T\_Variable* | *Set of external variable transmitted* |
| Sender | 1 |  |  |
| Receiver | 1 |  |  |
| *Requirement* | *1..n* | *T\_Requirement* | *The set of requirement which defined the packet (at least one)* |
|  |  |  |  |

**Verification Artifacts Styleguide**

(Proposal of 2013-10-14)

Interface Variables must be standardised.

Please use semantic versioning explained here: http://semver.org/spec/v2.0.0.html

Variables Namings are separated between internal (inside the EVC- Kernel) and external variables (used for communication between the EVC - Kernel and the interfaces). (see D2.4) All external variables are described inside the following documents:

1. openETCS API documentation
2. Subset-026-7
3. to be completed

External and internal variables are announced to the data dictionary. (Role and way to be defined)

Only variables accepted by the data dictionary are allowed to be used as variables inside the model.

The mapping between external and internal variables is defined inside the data dictionary.

**Data Dictionary**

The data dictionary will evolve over time and thus the need to have a synchronised set of variables has been identified a crucial part in a distributed development.

The proposal is to have a common document inside github with the following properties:

1. The name of the document is: "openETCSDataDictionary.xml"
2. The document is encoded with Extensible Markup Language (XML) see Extensible Markup Language Definition
3. All contributions to the document are launched as a pull request
4. Each variable will be represented within one xml- block consisting of
   1. Variable name (see naming converntions within D2.4)
   2. Variable type (see SysML Version 1.3 definition June 2012 and ISO/IEC 9899:TC2 specification)
   3. Variable source (the name of the function or name of the interface the variable is derived from)
   4. Requirements link either the use of the variable is defined or, if available, the variable itself is defined(see naming convention within D2.4)
   5. Risk: possible ambiguity used for refinement of data dictionary, proposals e.g. to replace one variable by differentiating two aspects can be filed here (free text, string for later reuse)
5. The data dictionary will reside inside the SSRS Repository under the responsibility of the SSRS task leader.
6. Merging the block into the repository will be done according to the rules set within the SSRS task
7. Merging of the block related to the variable into the XML file master branch means that the variable has been
   1. Checked for redundancy within the data dictionary
   2. Checked for naming conventions and restrictions are met (defined within D2.4)
   3. Checked that all information related to the variable is given
   4. Approved and accepted by the SSRS team

**Internal Functions**

Naming of internal functions:

Internal functions are named using an active form consisting of at least a combination of a verb and an object

Example:

1. initiateTerminatingASession

**Internal variables**

Naming of internal variables

The naming of the variables should consist of at least a combination of a verb and an object using the passive form

Examples:

1. orderReceivedFromTrackside for a structure
2. endOfMissionIsExecuted for a boolean

Property of internal Variables

1. Can be correlated to external variables
2. Used to create internal functions
3. The source the internal variable is derived from is given in the name of the function
4. The variable type is defined within SysML Version 1.3 definition June 2012 and ISO/IEC 9899:TC2 specification)

**Links :**

