Verification Report for Architecture and Design of Train Positioning Version 0.1

Marc Behrens (DLR), Bernd Gonska (DLR),
Jens Gerlach (Fraunhofer), Bernd Hekele (DB),
Jan Welte (TU-BS)

Oct 29, 2014

Abstract

This verification report presents the verification results for the architecture, interfaces and design artifacts for the component "Train Positioning" in the overall openETCS Kernel architecture.

1 Roles

- Fausto Cochetti - Design (PM)
- Jens Gerlach - Verification of SW/Implementation
- Uwe Steinke - Design/Implementation
- Jan Welvaarts - Design
- Vincent Nuhaan - Simulation/Design
- Bernd Gonska - Verification
- Marc Behrens - Verification
- Jan Welte - Verification
- Bernd Hekele - Verification

2 Verification Object

2.1 Identity of Verification Object

The component `Train Positioning` provides the central functionality to determine the train position based on detected balises and their measured and announced distances.

The component is defined in the `openETCS System Architecture and Design Specification` [3] as F.2.2 Calculate Train Position. Here it is defined that the function shall cover the requirements defined in the SRS [2] Chapter 3.6.

Concepts for `Train Positioning` have been defined in two parallel work streams, which developed independently two documents and simulations to clarify their approach. To achieve one implementation in context of the overall openETCS architecture both concepts have to be integrated. Respectively, this verification report is concerned with the following two groups of documents, which are handled as one verification object. All artifacts for verification and verification activities are documented in the github issue `https://github.com/openETCS/validation/issues/227`.

`Train Positioning` **concept 1**

reviewed document	Train Position and Locations (concept document)
	Train Position and Location Tester (Lab View Simulation)
github location	`https://github.com/openETCS/SRS-Analysis/commit/edc8a3238e59ad4d2a2440f39e4c791cf6bbf7bd`
comitted	10. July 2014
designer	Jan Welvaarts (concept)
	Vincent Nuhaan (simulation)
covered specification	SUBSET-26(SRS) chapter 3.6 version 3.3.0

`Train Positioning` **concept 2**

reviewed document	openETCS Determine Train Location Procedure (concept document)
github location	`https://github.com/openETCS/SRS-Analysis/commit/153e793955b38c986dad3bfd8d3fbfe8d5ced77e`
comitted	21. January 2014
reviewed document	CalculateTrainPosition (SCADE report)
github location	`https://github.com/openETCS/modeling/commit/5cea2abbf67da49b324aa8307129b5c47aba2188#diff-9fd0c16adf976a82850cd9184a13f0f5`
comitted	1. September 2014
reviewed document	openETCS CalculateTrainPosition implementation Test Scenario 3_linked_2_unlinkedBG (Test Scenarios)
github location	`https://github.com/openETCS/modeling/commit/3934d76f1fcba7cb28da7e8131f201182cc7d220`
comitted	22. October 2014
designer	Uwe Steinke (concept, SCADE implementation, Test Scenarios)
covered specification	SUBSET-26(SRS) chapter 3.6 version 3.3.0

This verification reports only address the documents in the committed versions named above.

The verification mainly concentrated on the concept papers and took the simulation and the SCADE model only in consideration to specify the approach presented in the concept documents.

2.2 Configuration related Components

The functionality `Train Positioning` has been defined with respect to the SUBSET-26 System Requirements Specification (SRS) [2] chapter 3.6 version 3.3.0 on the legal basis of the Technical Specification for Interoperability Control Command Signalling (TSI-CCS) [4] and amended by [5].

The component limits have to be defined by the `openETCS System Architecture and Design Specification`, which has to present the interfaces either with respect to data from balise (and radio) messages defined in the SUBSET-26(SRS) chapters 7 and 8 or as in consistence with inputs and outputs of related functions.

As the `openETCS System Architecture and Design Specification` have not been available at the beginning of the verification and not all functions interacting with `Train Positioning` are covered completely, the verification results have to accept assumptions made by the designers. Additionally, no further requirements for the interfaces have been assessed.

3 Software Architecture, Interface and Design Verification

This section presents all verification results concerning the verification object `Train Positioning`.

The following subsection present all different verification aspects in accordance with EN 50128 7.3.4.42 [1].

3.1 Internal Consistency

by Jan Welte and Marc Behrens

Contant:

- relations
- historical development
- claim of same approach
- of naming between documents consistence

Are the internal functional allocation and all related input and output consistent?

3.2 Adequacy to fulfill Software Requirements

by Bernd Gonska

3.2.1 Description of the developed train positioning system

The train positioning system clearly deviates from the SRS on purpose. This is justified by the intended operational performance.

It basically implements the following concepts:

- All distances are given as the triple of safe distances (minimum,nominal,maximal)
- The estimated position, (also named nominal position) is calculated to be the middle of the maximum safe position and the minimum safe position.

- Each Balise Group (BG) has a an own accuracy and position, relative to the Last Relevant Balise Group (LRBG) and the LRBG accuracy. Locations do not change their reference BG.

- Linking distances and accuracies are used to improve the accuracy when ever possible.

- Accuracy of a distance is calculated taking the worst possible case into account. For example: The accuracy of the distance between two ends of a linking chain includes the first and the last BG accuracy. The distance between two BG without linking is the odometry measured distance. The accuracy is the odometry acuracy during that distance. This inaccuracy is not reseted later.

- The confidence interval of an announced location never increases when a new BG is accepted. Always use the most accurate information. The odometry error estimation is trustworthy enough to optimize linking accuracy and distances.

3.2.2 Deviations

Within this paragraph the deviations to the specification is described by first mentioning the number of the Subset-026 paragraph [2] and then stating how the design deviates:

3.6.4.1 REMARK: There are several confidence intervals: They depend on the announced location.

3.6.4.2: In addition, the odometry inaccuracy of older track areas and older linking accuracy can be taken into account to widen a the confidence interval for safety reasons. The location accuracy of the LRBG is shortened on if the detected Balise group position is extreme. An old confidence interval can be taken instead of a larger new one.// 3.6.4.2.2: An odometer inaccuracy may not be reset at the new LRBG.

3.6.4.3: Even if the linking chain is not complete, linked parts replace odometry distances if they provide higher accuracy.

In the "sandwich problem" where two linked BGs enclose an unlinked BG (linked BG1 → unlinked BG2 → linked BG3) the distance and the accuracy between BG2 and BG3 can be calculated involving the linking accuracy of BG1and BG3 and the linking distance between BG1 and BG3.

The estimated distance may differ from the linking distance and from the odometry measured distance. The estimated distance is set to the middle of the maximum and minimum safe distance.

3.6.4.3.1: The train takes responsibility, it does not reset inaccuracies if this could lead to unsafe behavior.

Figure 13 a,b,c: There is more than one confidence interval.

The confidence interval is calculated differently.

The estimated distance can be different since it is the middle of the maximum and minimum safe distance.

Linking distance is not used if it leads to a less accurate distances.

3.6.4.4: The estimated distance is the middle of the maximum and the minimum safe position with respect to the possible LRBG position. It may differ from the measured traveled distance.

3.6.4.4.1: analogue for the rear end position.

3.6.4.7.1: The unlinked BG confidence interval is not reset at the next LRBG.

3.6.4.7.2 The unlinked BG confidence interval is not reset at the next unlinked BG. In some cases the estimated traveled distance between two unlinked BG is calculated by using other rules.

3.2.3 Readability and Traceability

by Marc Behrens

Content

- traceability of requirements
- unique references

Are all related system and software requirements uniquely referenced and is the relationship to other documents clearly defined? Are all parts of the architecture and inputs and outputs referenced to the related requirements. Are the elements referred to in the same way in all documents?

- 10' item: look for findings inside the two verification reports

- 10' structure: two chapters, one for each report - 20' 3 documents: readability: statistics: wordlength + syllable over sentence length - 20' look in 2.x: traceability to openetcs req - 10' only 3.6 available traceability to SRS requirements - high level - traceability to TSI requirements - 20' are there more req in TSI to positioning? - 10' what is missing - 15' design reasoning: what is needed for performance resons - 5' what is the least cycle time - 5' are realtime requirements defined on architecture level? - 10' traceability to higher level artifacts - 10' high level requirements were defined during workshop as RO US who will be response - 10' application of glossary - 5' subset-023 - 5' openETCS glossary - 5' make jan's document openETCS licensed + upload

3.2.4 Consideration of hardware and software constraints

Hardware design is out of scope of the openETCS project. No hardware assumptions have been formulated so far.
Software constraints encompass:

1. Constraints by the software design method. The design should rely on
    - modelling
    - a modular approach
    - defensive programming
2. Restrictions implied by the coding standards. The coding standard should include
    - a coding style guide
    - restrict the use of pointers, dynamic objects, recursion and global variables
3. Constraints on timing, performance, or memory of individual modules.
4. Any constraints implied by the interfacing system (e.g. decoder and encoder functions).
5. Constraints of the operating system.

References

[1] Comité Européen de Normalisation Electrotechnique. Railway applications - Communication, signalling and processing systems - Software for railway control and protection systems, EN 50128. *EUROPEAN STANDARD*, 6 2011.

[2] ERA, UNISIG, and EEIG ERTMS USERS GROUP. ERTMS/ ETCS System Requirements Specification, SUBSET-026. *European Railway Agency Document Register*, 3 2012. `http://www.era.europa.eu/Document-Register/Documents/Set-2-Index004-SUBSET-026%20v330.zip`.

[3] Bernd Hekele, Paymann Farhangi Peter Mahlmann, Uwe Steinke, Christian Stahl, and David Mentré. openetcs system architecture and design specification. *openETCS modeling repository*, 10 2014. `https://github.com/openETCS/modeling/blob/18cd45f54932b46dde7fea404f21aa5b27bcc516/openETCS%20ArchitectureAndDesign/FirstIteration/openETCSArchitectureAndDesignSpecification.pdf?raw=true`.

[4] European Union. Commission Decision of 25 January 2012 on the technical specification for interoperability relating to the control-command and signalling subsystems of the trans-European rail system. *Official Journal of the European Union*, pages L51/1–L51/65, 2012.

[5] European Union. Commission Decision of 6 November 2012 amending Decision 2012/88/EU on the technical specifications for interoperability relating to the control-command and signalling subsystems of the trans-European rail system. *Official Journal of the European Union*, pages L311/3–L311/13, 2012.