openETCS Toolchain WP Description of Work

September 4, 2012

This Work Package will provide the tool chain that is necessary to formalise the ETCS system specification. The formal specification will be used further for verification and code generation. The tool chain must support the following tasks:

1. Support the authoring of the formal ETCS system specification.
2. Support the creation of a formal ETCS software specification (while the ETCS specification describes the *problem*, the software specification describes the *solution*).
3. Support code generation from the software specification.
4. Support the verification and validation of the various artefacts, including the formal ETCS specification against the textual ETCS specification.

The tool chain definition will benefit from other R&D projects and off-the-shelves tools. The semantics of the modelling languages shall be carefully studied.

The first goal of this WP is to identify sets of consistent languages and tools enabling the design of the system. This will be done in close collaboration with WP2.

1 Tool Chain Architecture Analyses and Recommendations

The first subtask is the tool chain architecture analyses and recommendations. This tool chain will encompass all description levels of the system design from holistic viewpoint to code generation. Specific attention shall be paid to the semantics and the traceability of each part of the chain. This task is composed of the following activities:

1.1 Identify/define the potential modelling languages

Given that different levels of abstraction have to be addressed in the design phase, several languages may be necessary to handle all design phases. Here, the semantics of the languages is the key point; it shall be accurately adapted to the level of description required in each phase of the design.

(mj) I think there is a huge overlap with WP2. Their findings may reduce the possible target languages drastically, and by their requirement must take everything written here into account anyway.

Input:	WP2 State of the Art Analysis	Oct-12
	WP2 Requirements (incomplete)	Oct-12
Output:	Modeling Language(s) (and reasoning)	???

1.2 Decide which existing tool platform shall be used as the foundation

Based on the state of the art findings and requirements from WP2, an existing tool platform will be chosen for implementing the tool chain.

Input:	WP2 State of the Art Analysis	Oct-12
	WP2 Requirements (incomplete)	Oct-12
Output:	Tool Platform (and reasoning)	???

1.3 Identify the modelling tools architecture

(mj) I assume that the previous task will pretty much dictate the platform architecture.

These tools enable designers to model using the languages defined before. The interoperability of the tools is a key point to address traceability of the different models in the design process. Tools may already exist and may be used “off-the-shelves” or may be developed in the WP. As example, if a standard modelling language is selected such as UML/SysML, open tools already exists (Topcased); if a specific modelling language is needed, WP actors have to develop a grammar or meta model to develop such particular modelling tool.

1.4 Analyse requirement elicitation techniques

Analyse requirement elicitation techniques in order to define a strategy (mj: process?) to derive OpenETCS formal model requirements from ERTMS FRS/SRS

Input:	WP2 Requirements (incomplete)	Oct-12
Output:	Requirement Elicitation Techniques	???

1.5 Verification Tools

(mj) I think this is not on the cricital path.

Identify potential verification tools with regard to modelling techniques; verification techniques shall be investigated.

Comment Hardi Hungar:

Verification tools resp. techniques are very important in the development of a safety-critical system like the ETCS OBU. According to the relevant standards (most prominently the EN 50128 of the CENELEC family), every design step has to be verified. Assuming that models will constitute artifacts oft he development – and are not just used for explanatory purposes – it is necessary to be able to establish the correctness of each refinement step. Or, to put it differently, the tool chain needs a concept for seamless verification, preferably tool-based, to be fit for its purpose.

I think, this must be taken into account already early in the definition process. Therefore, while it might not be the first thing to consider (without modeling, there is no verification of models), it should definitely not be the last.

Comment Stanislas Pinte:

I agree with Hardi.

In my opinion, the model should include the tests of the model, so that it could be verifiable in a "model-in-the-loop" fashion.

It doesn't have to be model proving (that I think belons more to other WPs) than model testing.

Inside our http://www.ertmssolutions.com/ertms-formalspecs/ approach, we assume the following:

- Model tests are part of the model - Model should be 100% covered by tests (proved by model coverage reports) - Toolchain must support developing, executing and debugging tests

Model verification also includes:

- veryfing that the model corresponds to the orginial requirements specifications (in our case, UNISIG Subset-026 BL3). ERTMSFormalSpecs also supports marking model artifacts as "verified" against source requirements. - veryfing that 100% of source requirements are traced against one or more model artifacts (proven by traceability reports)

I would think that such verifications are indeed in the critical path...i.e. if not implemented we shall not be able to have a fully functional model.

Input:	WP2 Requirements	???
Output:	Verification tool choice	???

1.6 Code Generation Strategy

(mj) I think this is not on the cricital path.

Analyse the code generation strategy.

Input:	WP2 Requirements	???
Output:	Code Generation Strategy	???

1.7 Model Transformation

(mj) I think this is not on the cricital path.

Analyse model transformation techniques and tools in order to refine the specification from one description level to another.

Input:	WP2 Requirements	???
Output:	Model Transformation Strategy	???

1.8 Schedulability

(mj) I think this is not on the cricital path.

Analyse schedulability tools.

Input:	WP2 Requirements	???
Output:	Schedulability Strategy	???

1.9 Capture Additional Requirements

Capture wishes/requirements on how to support the designer in their activities.

Input:	Designer Wishes and Requirements	ongoing
Output:	Captured and organised designer requiremenets	???

2 Define and Develop Tool Chain

(mj) I see ”make” not as an option. Considering the resources available for this project, we will tailor (and extend) an exisiting tool platform.

The second subtask defines and develops the tool chain and the infrastructure enabling its evolution and maintenance. First of all, a ‘make or reuse’ decision about the components of the tool chain has to be made. Then a common development infrastructure has to be defined or chosen in order to integrate all the tools (Eclipse like infrastructure). Finally, the subtask achieves the development and the integration of the tools.

TODO: Further subtasks to be developed.