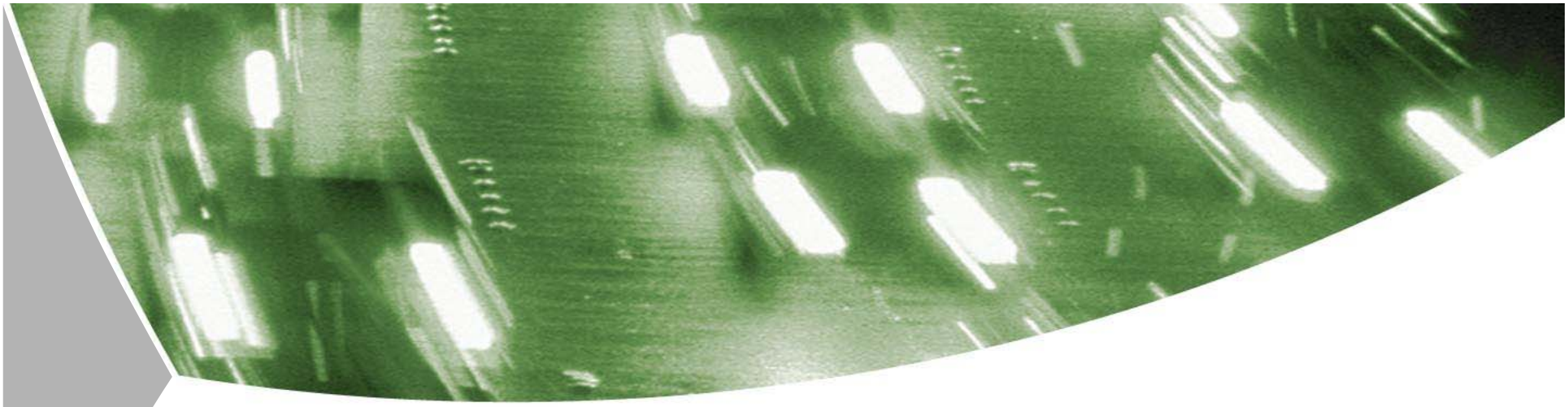


openETCS WP2 Workshop – V&V workshop

„Process/ Methodology and V&V strategy“

23.11.2012, Paris

Stephan Jagusch AEbt; Jan Welte, TU-BS; Rico Kaseroni, DB; Renaud De Landtsheer, Alstom; Hardi Hungar, DLR; Marc Behrens, DLR

Deutsches Zentrum
für Luft- und Raumfahrt e.V.
in der Helmholtz-Gemeinschaft

Content

- openETCS Challenges
- WP4 Key Aspects of openETCS
- Outside of WP4
- Open Proof Approach
- WP4 Requirement Structure
  - General
  - Exchange formats
  - V&V Model/ Code
  - Procedure and Safety
- Way Ahead on V&V Requirements & Workshop Results

openETCS Challenges

- The specification languages and modelling need to be adapted to the specific problems of **embedded systems**, that is to say, **real-time, fault tolerance, safety, security**, etc.
- The test methods, formal verification and related tools need to be tailored to networked embedded systems to deal with **safety** as well as other **non-functional** aspects (i.e. security, performance, real-time...).
- Methods and techniques are needed for **reducing the vulnerability** of embedded systems to external possibly 0-day attacks (viruses, intrusions...).

WP4 Key Aspects of openETCS

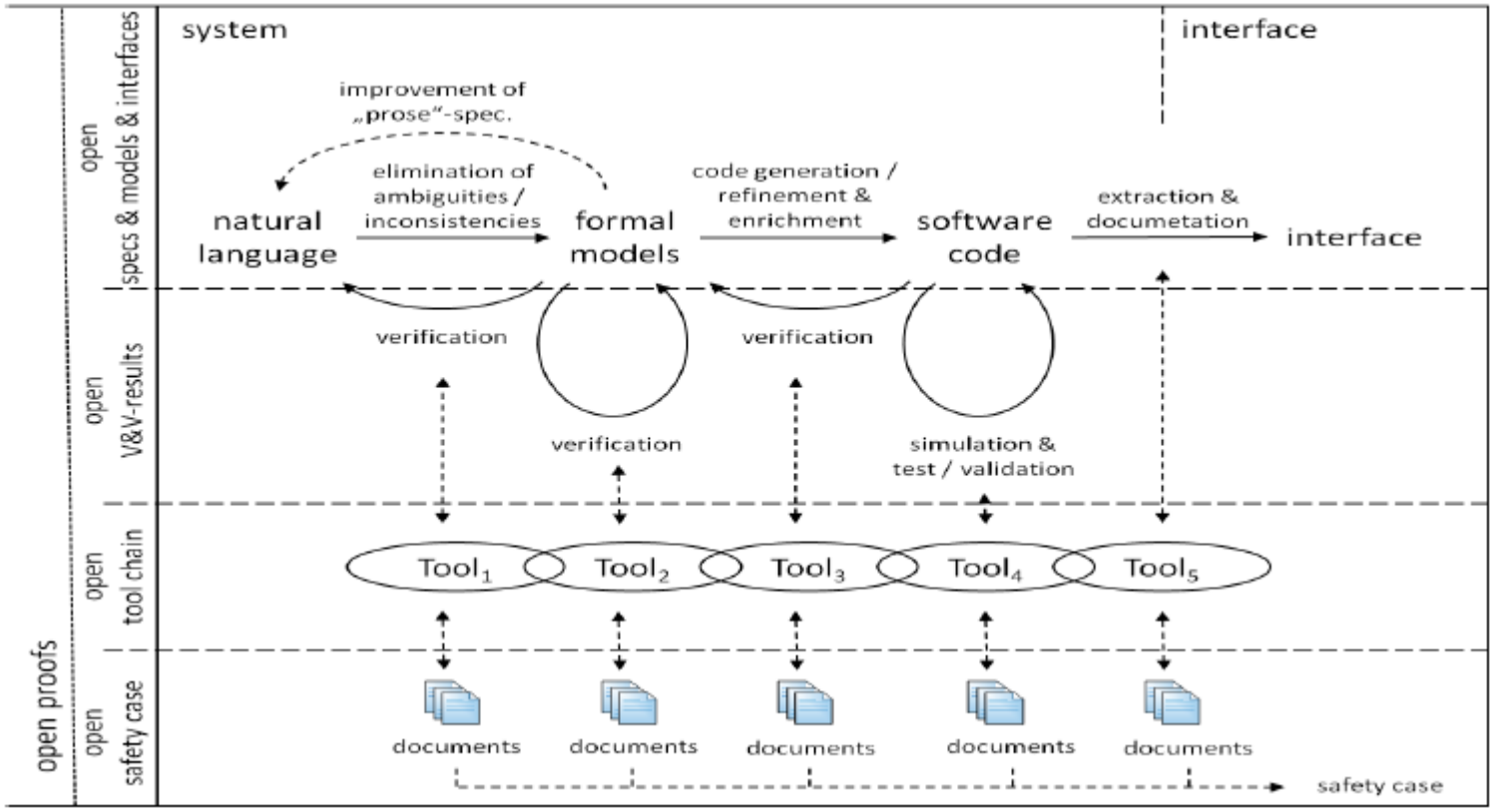
- Strategy describing …
  - concept how the **consistency, coherence of the model** as well as the coverage of system requirements will be transparently verified.
    - How the correctness of requirements will be assessed?
      How will risk analysis/FMEA be included in the formal development process?
      - Have a look at Goal-oriented requirements engineering
  - how to **validate the model** for reliability and acceptance
  - **code consistency** with the model
  - generating **feedback to modelling** process
  - how to **measure quality and maturity** of the model
- Strategy for
  - … verification and validation of the implementation/code …
- Evaluate and choose **techniques and tools for V&V**
- Validation and verification **plan**
- Verification of **tools** and **processes** (Safety case, hazard log, …)

Outside of WP4

- Concerning Model and Tool development: (→ WP3)
  - the implementation of the **core functionalities** of the ETCS onboard unit,
  - semiformal and formal models & code

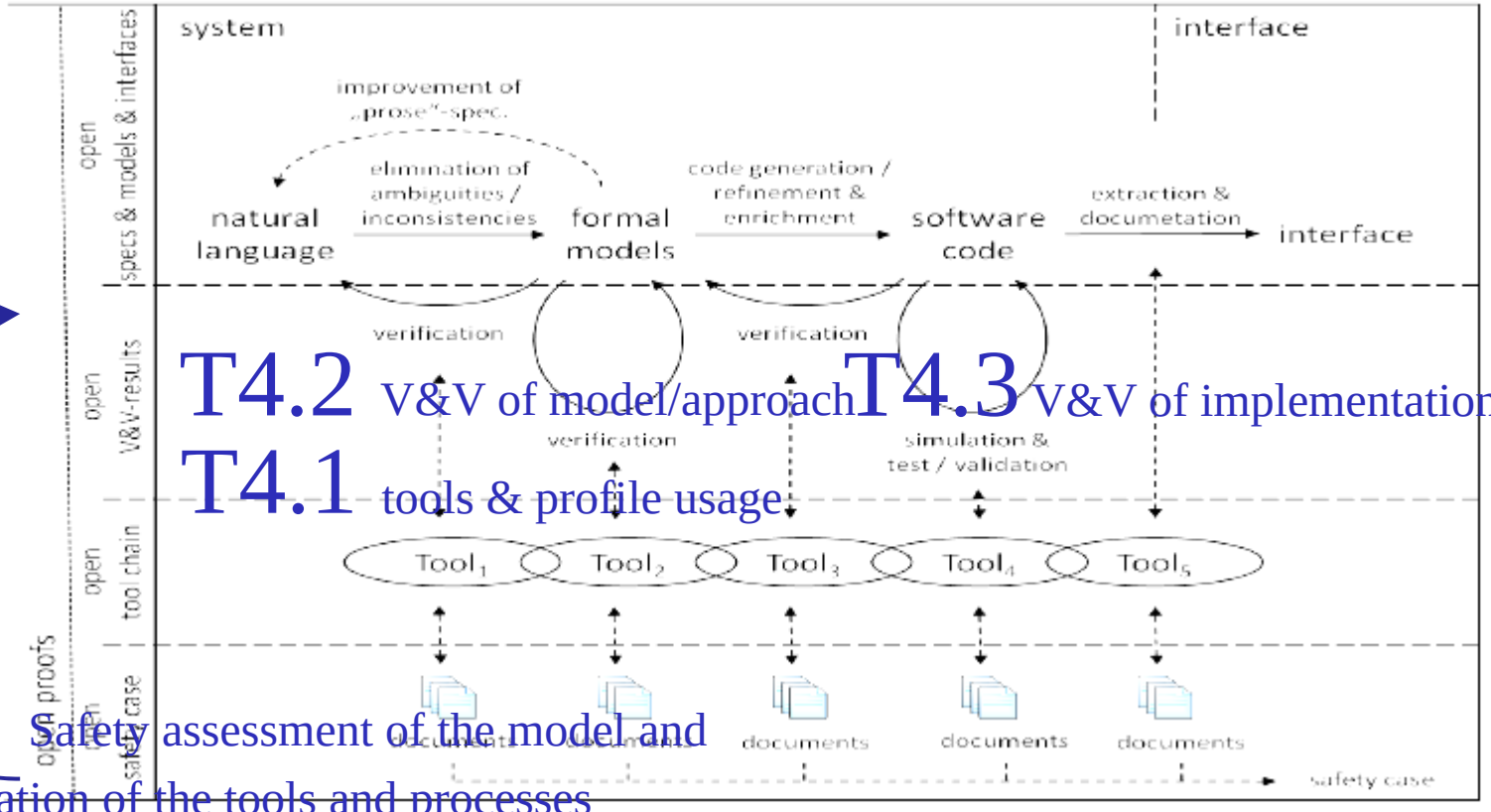
- Concerning the **onboard simulator**: (→ WP 5)
  - Defining test scenarios
  - data preparation
  - execution of scenarios
  - data exploration
  - execution of industrial tests

Open Proof Approach



form:: 11025_openETCS_Full_Project_Proposal_Document_FPP_MERGED.pdf

Open Proof Approach – WP4 Task Matching



T4.2 V&V of model/approach

T4.3 V&V of implementation/code

T4.1 tools & profile usage

T4.4 Safety assessment of the model and verification of the tools and processes

form:: 11025_openETCS_Full_Project_Proposal_Document_FPP_MERGED.pdf

WP4 Requirement Structure: General

- How to use scrum?
- Define requirements for requirements tracing
- Interface to other WPs
  - Which information is exchanged?
    - Model & CMB (Common Model Backlog)
    - Code & CCB (Common Code Backlog)
    - Tool & CTB (Common Tools Backlog)
    - Process & CPB (Common Process Backlog)
- Collect use cases of the API in V&V (potentially additional features required)
- Collect and discuss V&V Tools
- Define criteria for open proof
- Test plan
  - Simplified thanks to the use of formal methods

WP4 Requirement Structure: Exchange Formats

- **Exchange formats**
  - CMB – common model backlog (Info about model)
    - (at least) 2 Model levels: SRS, Architecture (Design)
  - CCB – Common code backlog
  - CTB – common tool backlog (info about tool)

- **WP4 reporting formats**
  - CMB assembled
  - CCB assembled
  - CTB assembled
  - CPB common process backlog / all Artifacts

WP4 Requirement Structure: V&V Model/ Code

- **Define requirements for verifiable model**
  - Respect requirement trace information from the model
  - Packed based interfaces for API or translatable
  - Evaluate CMB with the model
  - Assemble CMB for reporting

- **Define requirements for verifiable code**
  - Respect requirement trace information from the code
  - Packed based interfaces for API or translatable
  - Evaluate CCB with the model
  - Assemble CCB for reporting

WP4 Requirement Structure: Process and Safety

- Define requirements for process, tools and safety
  - Define conformity criteria (e.g. licence, open proof, standards …)
- **Tools** to collect and evaluate for conformity
- **Artifacts** to collect and evaluate for conformity
- **Procedures** to collect and evaluate for conformity
  - Seamlessness
- Approve tools in general (non functional)
- Define criteria for ecosystem release
- Define abstract steps in process
- Clearing of software-rights
- personal competence
- certification -> technical report: concept how to become certifiable

Way Ahead on V&V Requirements: Discussion

- Identify standards (CENELEC, TSI Subsets (26, 36 …), …)
- **Matching formats** needed for language and process definition as well as tool chain
  - Discussion and agreement about the **interfaces**
- Design and implementation of procedures for maintaining the backlogs
- Requirements of **V&V**: Identify key aspects for V&V concerning
  - Model (e.g. human readable)
  - Code
  - Proof (format, tools consistent)
- Requirements on **process/ tools**
  - Artifacts
  - Process (e.g. absence of media breach, tracing)
  - safety case (documents, SIL4, …)

Way Ahead on V&V Requirements: Discussion

- Clear view: Do we do it or do we describe it?
  - Testing
    - What can be replaced by e.g. proof, verifying the test suite
  - Formalising
  - Define which API functionality do we need for automation of testing
- How far do we go with the assessment?

Workshop Results: Standards & Reference Projects

- Standards
  - CENELEC EN50128, EN50129, EN50126 (EN50159)
    - Formalisation of process
    - Compliance to Artifacts
  - TSI (http://www.era.europa.eu/Document-Register/Pages/New-Annex-A-for-ETCS-Baseline-3-and-GSM-R-Baseline-0.aspx)
    - Subset-026 SRS
    - Subset-036 FFFIS for Eurobalise
    - …
  - Idea: Input from Project “deploy” on V&V (www.fm4industry.org)
  - Idea: Input from “Feasibility Study for the formal specification of ETCS functions” (http://www.era.europa.eu/Core-Activities/ERTMS/Pages/Feasibility-Study.aspx)

Workshop Results: Model and Code

- Model
  - Should we transfer the requirements into behavioural requirements?
    - Define a method to refine requirements, e.g. cut?
    - What if the model is only partially integrated?
  - How do we differentiate between model and code?
  - Ensure refinement from higher level model to low level model
  - What are the language-, tool-, model- related constraints?
  - Verdict about constraints on the model
    - Analysis on language/model/tool choices and their consequences for V&V
- Code
  - Does the source code have to comply with limitations to the hardware/ architecture? (Runtime, overflow…)
  - Ensure refinement of model to code

Workshop Results: Process

- At which point in the process do we have to validate the tools?
- Refine the development cycle (EN50128)
  - V-process down: 1. informal SRS/ FRS → formal model → code/ compiler
    - → Code & Compiler?
      - Mandatory: defined interfaces of code within architecture
  - V- Process up: Code Test → Integration test → HW/ SW Integration & code generation → (-> 1.) validation
- We need a model based test environment! (test system)
- Ensure that steps left out are redundant (e.g. covered by proof)
- How to ensure that the process is used?
- How to prove that the process is integrated?
- How to integrate proof (e.g. goal refinement) within functional modelling (Text ←→ Formal representation & verification)

Workshop Results: Process

- SRS ←→ Refine system requirements ←→ formal representation
- Define requirements for traceability
- Define requirements for WP3 (model/ code) to be met to be certifiable