

GoedelWorks and The ASIL project

Eric Verhulst

www.altreonic.com

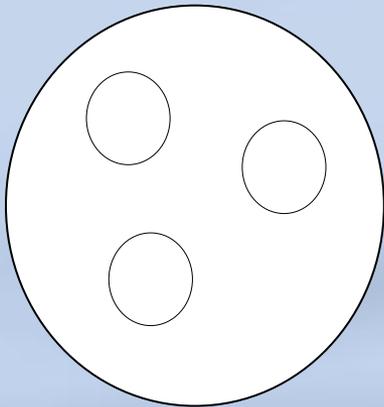
Some history

- R&D project of Open License Society:
 - Metamodel for systems engineering: “systems grammar”
 - OpenSpecs and OpenCookBook prototype web portal
- **EVOLVE** ITEA project
 - **E**volutionary **V**alidation, **V**erification and **C**ertification
- **ASIL**: Flanders Drive project on developing a common safety engineering methodology for automotive and related domains
- Currently commercialised and redeveloped by Altreonic under **GoedelWorks** by Altreonic

Refinement approach

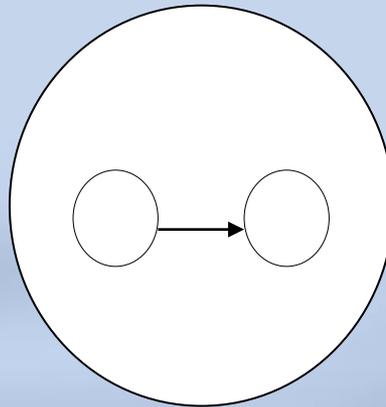
- Refinement by adding structure and properties
- Avoids overlapping in concepts

Domain



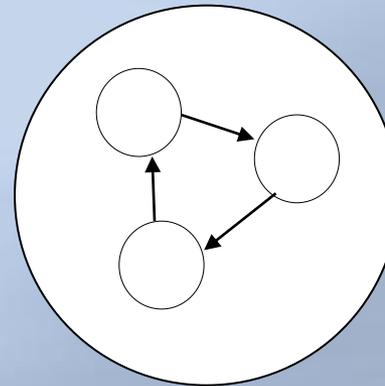
Entities

Metamodel



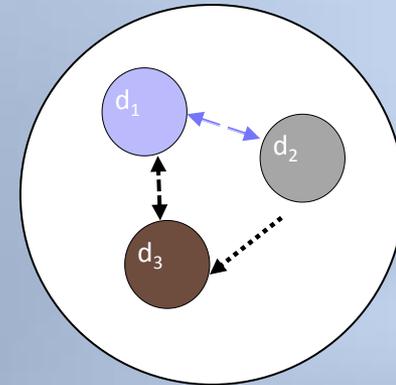
Entities and Interactions

Model



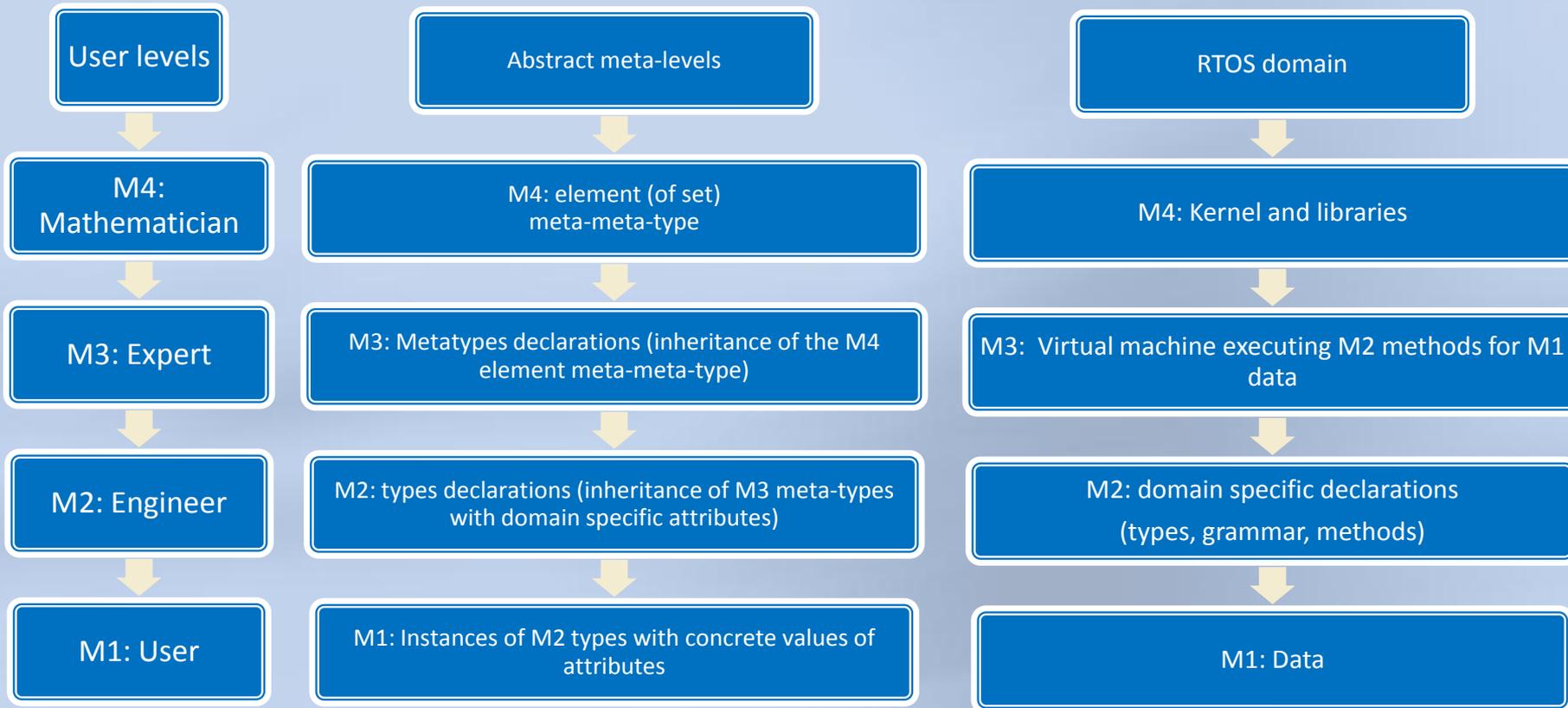
Structure & Architecture

Instance



System / Process

Meta-levels for different users and different application domains



The different views on a system

- View 1:

- System = Processes + Architecture
- or: the "right" System = "how" + "what"

- View 2:

- A process is a meta-system
- Has to be developed as well

- In practice different views correspond to complementary domains:

- Process, Engineering, Modeling, Simulation, Testing, Software, Hardware, Safety, ...

Systems engineering with just 16 meta-concepts

<u>System</u>	<u>Sub-systems</u>
<u>Project</u>	Sub-Project
<u>Process</u>	Sub-Process
<u>Reference</u>	
<u>Requirement</u>	Sub-Requirement
<u>Specification</u>	Sub-Specification
<u>Resource</u>	
<u>Work Package</u>	<u>Development, Verification, Test, Validation Task</u>
<u>Work Package Flow</u>	Work Package
<u>Work Product</u>	Process type (“evidence”) or development (“Model”)
<u>Model</u>	Sub-Models
<u>Entity</u>	Sub-Entities
<u>Change Request</u>	
<u>Issue</u>	

SYSTEM

PROCESS

PROJECT

Development Organisation Supporti ng Plan- ning

System under Development

*WPT-
template*

MODEL

WPT-
Process

WP

ENTITY

WP_Flow

Tasks

SPC

RES

WP

WP

DEV

REQ

Tasks

SPEC

RES

Tasks

SPC

RES

VER

REF

DEV

REQ

DEV

REQ

TST

VER

REF

VER

REF

VAT

TST

TST

VAT

VAT

GoedelWorks
Meta-Meta-concept

Relationships

- Dependency links:
 - E.g. a SPC depends on REQ (n)
 - etc.
- Precedence links:
 - A WP precedes a WPT (n)
 - etc.
- Structural links:
 - A WP is composed of Tasks (n)
 - A Model is composed of Entities (n)
 - etc.

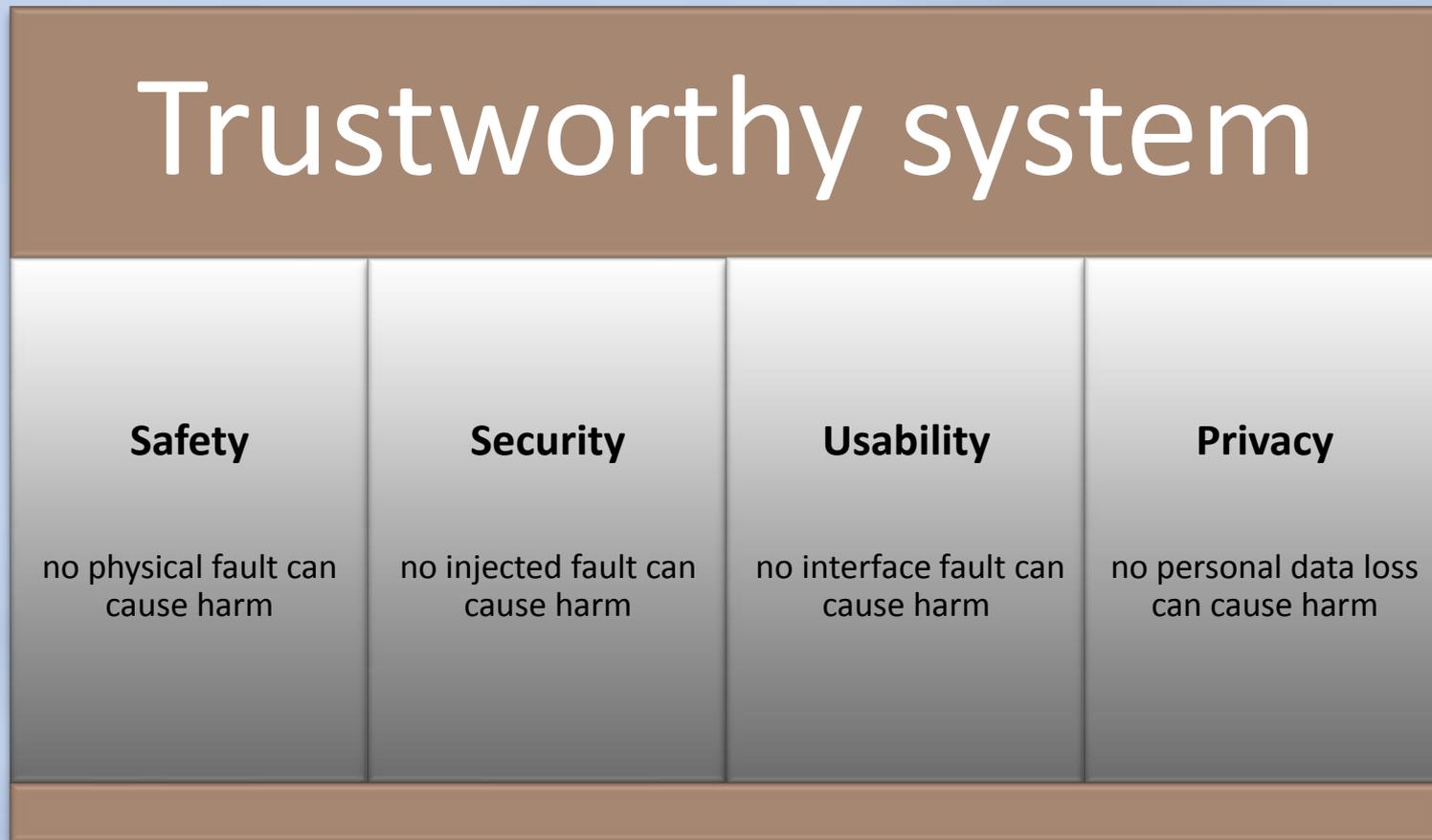
State Transitions in a Process

- During the life-time of a Project/Process entities go through states:
 - Defined => In Work => Frozen For Approval => Approved
- Dependency and structural relationships create a partial order for Approval
- **REF=>REQ=>SPC // RES // Tasks =>WP=>WPT (MOD)**
- A Project is a collection of Processes producing Work Products. Not one V-model but 100's.
- Overall Process follows from respecting states
- WorkProducts morphé (Resource at input is always result of previous Project)

Some differences

- Explicit difference between Requirements and Specifications
- Distinction Process (how) and Project (what)
- Verification = verifying the work done
- Testing = verifying the system meets specifications
- Validation = verifying it meets requirements (includes integration)
- Process/Project is not seen as flow but as a collection of steps producing WorkProducts
- System = Implementation model
- Safety case is seen as Specification-Fault case
- Domain agnostic

Trustworthiness as goal



Specification has subtypes:

Normal Case, Test Case, Fault Case

Safety and Security case are subtypes of Fault Case

Application and validation using a Safety Engineering process

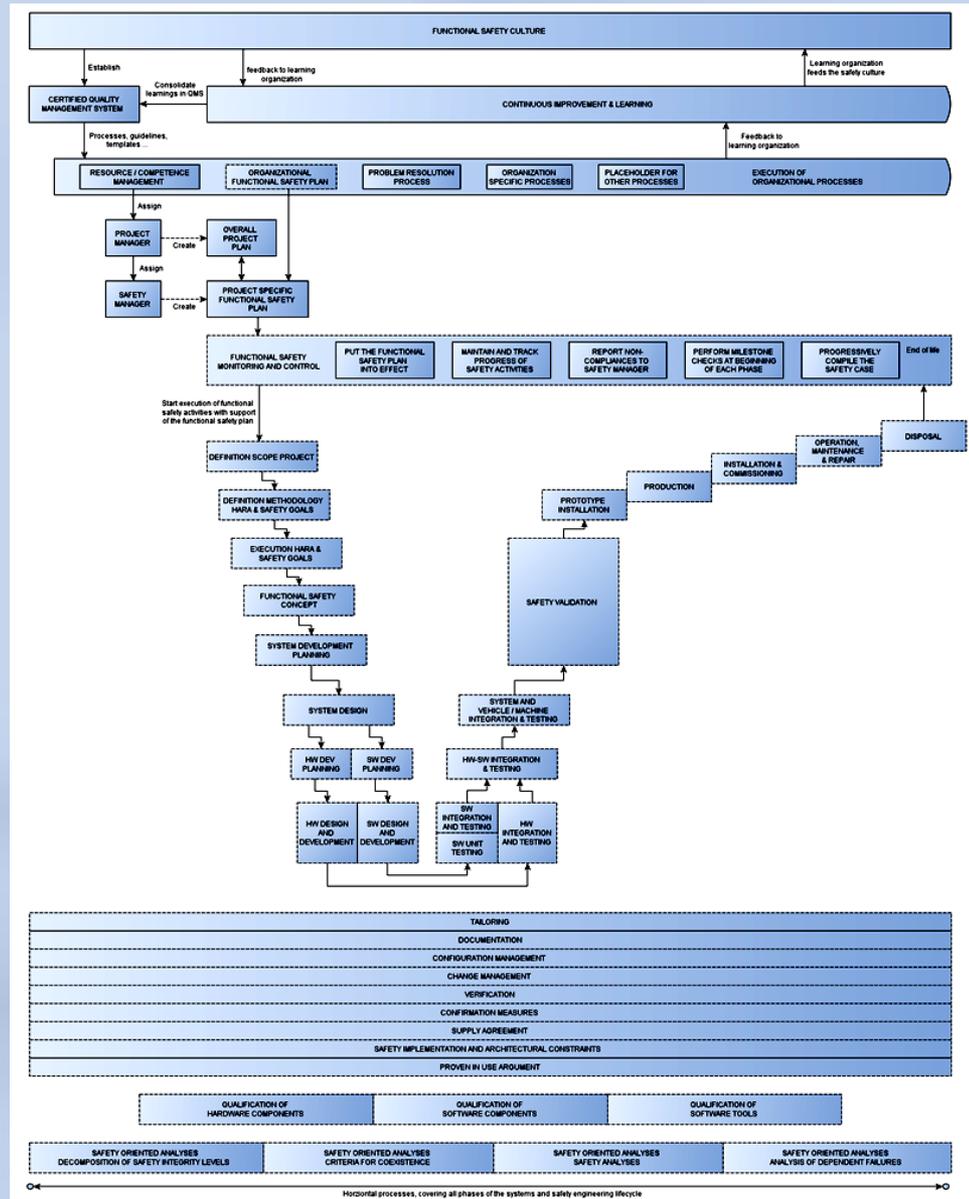
- Input: ASIL project of Flanders Drive
 - Automotive Safety Integrity Level
- Goal: develop common safety engineering process based on existing standards:
 - Automotive: off-highway, on-highway
 - Machinery
- IEC 61508, IEC 62061, ISO DIS 26262, ISO 13849, ISO DIS 25119 and ISO 15998
- Partners:
 - Altreonic, DANA, EIA, Flanders Drive, Punch Powertrain, Triphase, TÜV Nord

Process followed

- Acquiring general understanding of Safety and Systems Engineering standards.
- Development of ASIL process flow:
 - Dissecting standards in semi-atomic statements
 - Tagging according to activity domain
- Development of ASIL V-model with 3 Process domains:
 - Organisational Processes ("safety culture")
 - Supporting Processes
 - Safety and Engineering Development Processes.
- Completion
 - Identification of Work Products and RACI Roles
 - Development of templates for Work Products (.doc or .xls)
 - Development of Guidelines (e.g. HARA)
 - Development of Glossary

ASIL V-model

- Organisational
- Safety and Engineering/
Development
- Supporting



ASIL Results

- Effort: approx. 21000 personhours (over 3 years.)
- Semi-atomic process requirements extracted: 3800
- Work products defined: 98 => templates
- Types of roles identified: 17 => HR responsibility
- Guidelines developed: 34 => templates
- ASIL process flow has 355 steps
 - Organisational processes identified:19
 - Supporting processes identified: 75
 - Safety and Engineering processes identified: 261
- Work is not finished! (validation using use cases + organisation specific mapping) + iterative!

ASIL import (1)

GoedelWorks	ASIL
Process	Process
Flow	Flow
Work Package	Step with descriptive text
Tasks (DEV, VET, TST, VAT)	Not defined
Project	Not defined
Model/Entity	Not defined
Reference	Standards' requirements attached to Step
Requirement	Not defined, Step description
Specification	Not defined
Resource	Roles, Work Product template, Guidelines
Work Product	Work Product (input and output of Step)
Change Request	Not defined, but Change Management Step
Issue	Not defined, but Change Management Step
State	Not defined
Relationships	Not defined, except as WPT input and Roles

ASIL import (2)

- V-model respected by following order:
 - Steps become Work Packages
 - Dependencies and structural relationships inserted but left empty
 - State: most often “In Work” upon creation.
- Benefits from import:
 - All Process (and Project) Entities user-editable
 - Project entities and Process entities can be linked
 - Organisation specific instance of Processes can be created and new processes added
 - Dependency analysis and reporting

Example project (1)

ASi imported reference

The screenshot shows a web browser window displaying a project management interface. The left sidebar shows a tree view of project entities, with 'WP-206: 10_000_sysintegrationandtesting_001_sysinttestplanning' selected. The main content area shows the details for this work package, including its name, description, and a table of test methods.

Work Package: WP-206: 10_000_sysintegrationandtesting_001_sysinttestplanning (Finalize the system and vehicle/machine integration and testing plan)

Workflow: Status: In Work [Approve]

Edit Entity: WP-206

Name: 10_000_sysintegrationandtesting_001_sysinttestplanning [Finalize the system and vehicle/machine integration and testing plan]

Description Text:

During this step, the [system integration and testing plan](#) and vehicle/machine integration and testing plan is being finalized as basis for the system and vehicle/machine integration and testing activities. Besides, the [system testing](#) and [vehicle/machine testing specifications](#) are created during this step.

Note that the system and vehicle/machine integration and testing plan is already created during the [system design phase](#). Defining the integration and test strategy

The integration and test strategy is detailed based on the [system design specification](#), [functional safety concept](#) and the [system integration and testing plan](#). To ensure the appropriate specification of test cases for the selected system integration test methods, test cases will be derived using an appropriate combination of the methods listed in the table below:

Test methods for deriving test cases at system level	NR	O	R	MR
Analysis of requirements	-	-	-	ASIL A ASIL B ASIL C ASIL D
Analysis of external and internal interfaces	-	-	ASIL A	ASIL C ASIL D
Generation and analysis of equivalence classes for hardware software integration	-	-	ASIL A ASIL B	ASIL C ASIL D
Analysis of boundary values	-	-	ASIL B	ASIL D
Knowledge or experience based error guessing	-	-	ASIL A ASIL B	ASIL C ASIL D
Analysis of functional dependencies	-	-	ASIL A ASIL B	ASIL C ASIL D
Analysis of common limit conditions, sequences, and sources of common cause	-	-	ASIL A ASIL B	ASIL C ASIL D

Responsible: Project Manager

Consulted: Test Manager

Example project (3)

Dependency graph (Process)

The screenshot displays the GoedelWorks r480 web application interface. The browser window shows the URL <http://...> and the page title "GoedelWorks r480". The application is running in Mozilla Firefox. The main content area is titled "Requirement: REQ-119(09b_000_Verification of SW Architectural design (Verification of Software Architectural Design))". It features three tabs: "Changelog", "Dependency Tree", and "Precedence Tree". The "Dependency Tree" tab is active, showing a hierarchical tree view of requirements. The "Node View" tab is also active, displaying a dependency graph for the selected requirement. The graph consists of several nodes connected by arrows, representing dependencies. The nodes include: SPC-119 (09b_000_Verification of SW Architectural design), RES-119 (09b_000_Verification of SW Architectural design), WKP-119 (09b_000_Verification of SW Architectural design), WPT-42 (Software_verification_report.doc), DVT-119 (09b_000_Verification of SW Architectural design), VET-119 (09b_000_Verification of SW Architectural design), REF-1838 (ISO_CD_26262-6-6.4.20-0), and IEC_62061-N.A.-6.11.3.1.2-0. The graph shows a complex web of dependencies, with some nodes having multiple incoming and outgoing arrows. The interface also includes a left-hand navigation pane with a tree view of project entities, a top navigation bar with "Create" and "Generate Document" buttons, and a bottom status bar indicating "1 of 46 rows".

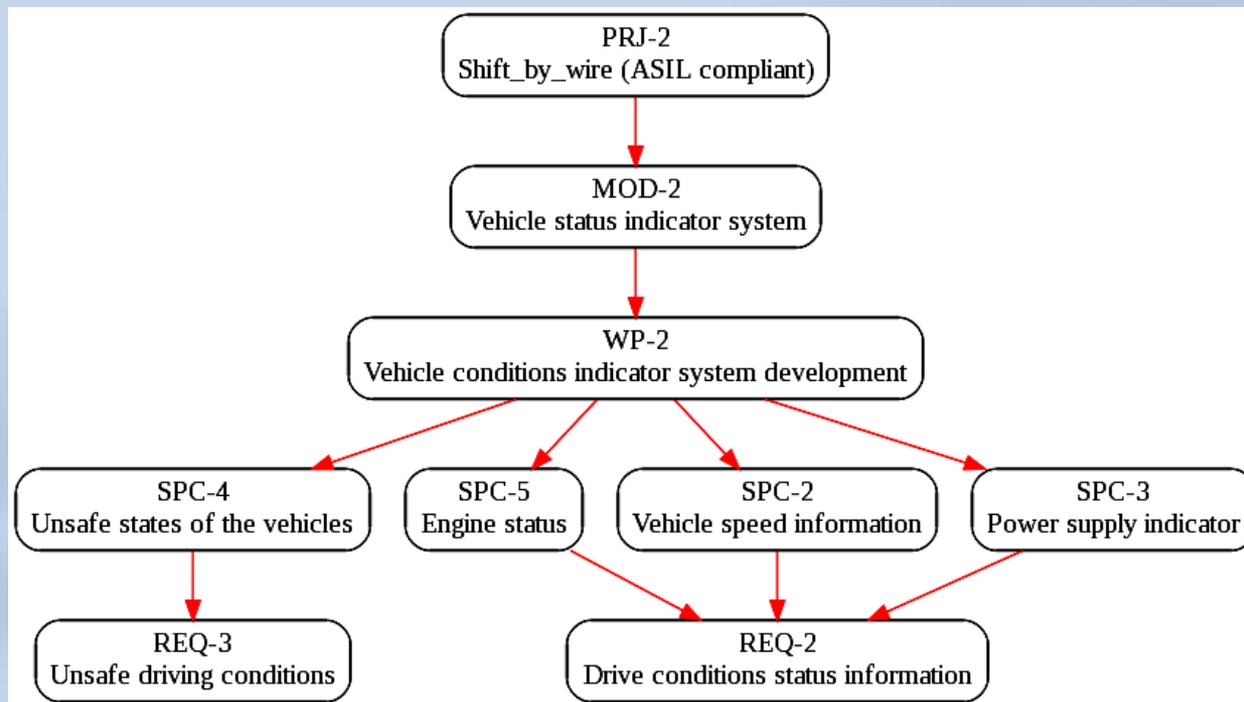
Example project (4)

- Example: shift-by-wire example

The screenshot displays a web browser window with a project management interface. The browser address bar shows 'http://'. The interface includes a navigation menu on the left with a tree view of project entities. The main content area shows details for 'Project: PRJ-1: Shift_by_wire (ASIL compliant)'. The status is 'In Work' with an 'Approve' button. The 'Edit Entity: PRJ-1' section shows the name 'Shift_by_wire (ASIL compliant)' and status 'In Work'. The description text reads: 'The scope of this project is the development of a shift-by-wire system for a CVT type transmission. The scope of the shift-by-wire system (also referred to as 'system') is the combination of the shifter, TCU and transmission (including a gear-change-lever actuator). Sub-systems are the shifter, TCU, transmission and'. Below the text is a diagram showing a 'Shifter' component connected to a 'TCU' component. A dashed vertical line is positioned below the shifter, and a horizontal line labeled 'A' connects the shifter to the TCU. A vertical line labeled 'B' connects the TCU to a transmission component below it.

Example project (5)

- Generated precedence graph



OpenCookBook2GoedelWorks

- Main lessons learned:
 - Bridging different domains: semantic differences
 - Safety engineering standards are subsets of systems engineering
 - Certification requires “evidence” (artifacts)
- Major problems:
 - Find a common language
 - Find a clean language: orthogonality
 - Usability aspects prime requirement for tool
 - Difficult in a web based environment
 - Standards’ license terms!

Conclusion

- Systems engineering process can be formalised using a common metamodel
- Booklet available from Altreonic website
- Challenges
 - Integration of different domains
 - Concepts, Architectural design, WorkFlow
 - System Engineering processes (“standards”) are heuristic
- Progress through formalisation
 - Reduction of design space give reliability
 - Modular architecture and unified semantics essential for incremental/evolutionary verification/validation/certification
 - Automated support is feasible
- Work will continue in OPENCROSS FP7 project
 - (cover avionics, railway, automotive)
 - Focus on re-use of certification evidence

8/28/2013



AL TREONIC
"FROM
DEEP
SPACE TO
DEEP SEA"

TRUSTWORTHY SYSTEMS ENGINEERING WITH GOEDELWORKS

First publication in the
Gödel Series:

SYSTEMS
ENGINEERING FOR
SMARTIES



More info at
www.altreonic.com