An Unified Meta-Model for Trustworthy Systems Engineering

Eric Verhulst

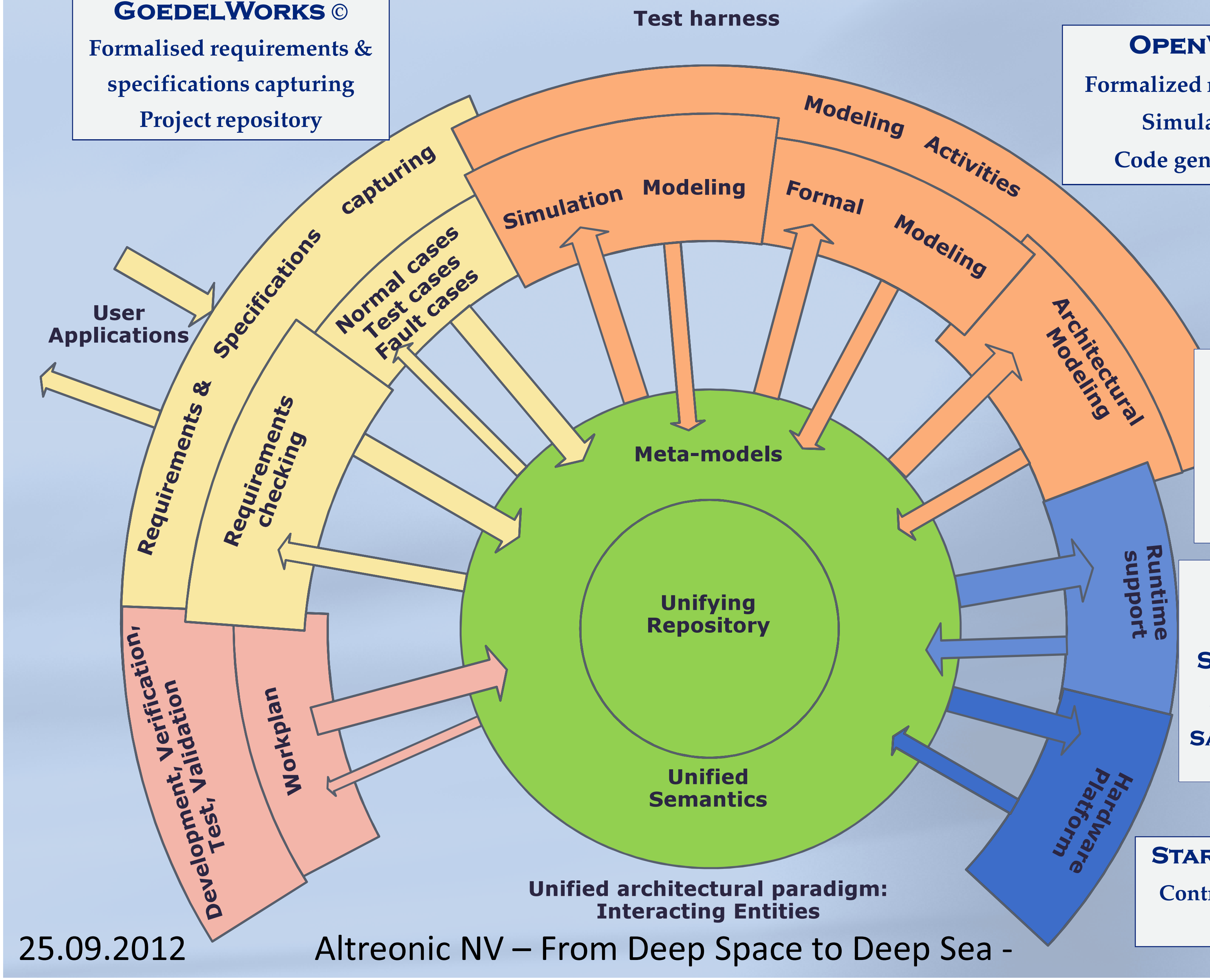
eric.verhulst@altreonic.com

http://www.altreonic.com

Altreonic

Systems Engineering Context

GOEDELWORKS ©
Formalised requirements &
specifications capturing
Project repository



Unified architectural paradigm:
Interacting Entities

OPENVE ©
Formalized modelling
Simulation
Code generation

OPENCOMRTOS ©
Formally developed
Runtime support for
concurrency and
communication

OPENTRACER ©
Visual Tracing
SYSTEM DEBUGGER©
System Level Debugger
SAFEVIRTUALMACHINE ©

STARFISH CONTROLLER ©
Control & processing platform

Models and systems engineering

- **Wittgenstein** (in “Philosophical Grammar” 65 years ago):
  - *“A **blueprint** serves as a **picture** of the object which the workman is to make from it.*
  - *… for the builder or the engineer, the blueprint is used as an **instruction or rule dictating how** he should construct the building or machine. And if what he makes deviates from the blueprint, then he has erred, built incorrectly en must try again.”*
  - *… What we may call **‘picture’ is the blueprint together with the method of its application”.***

=> Wittgenstein defined Systems Engineering

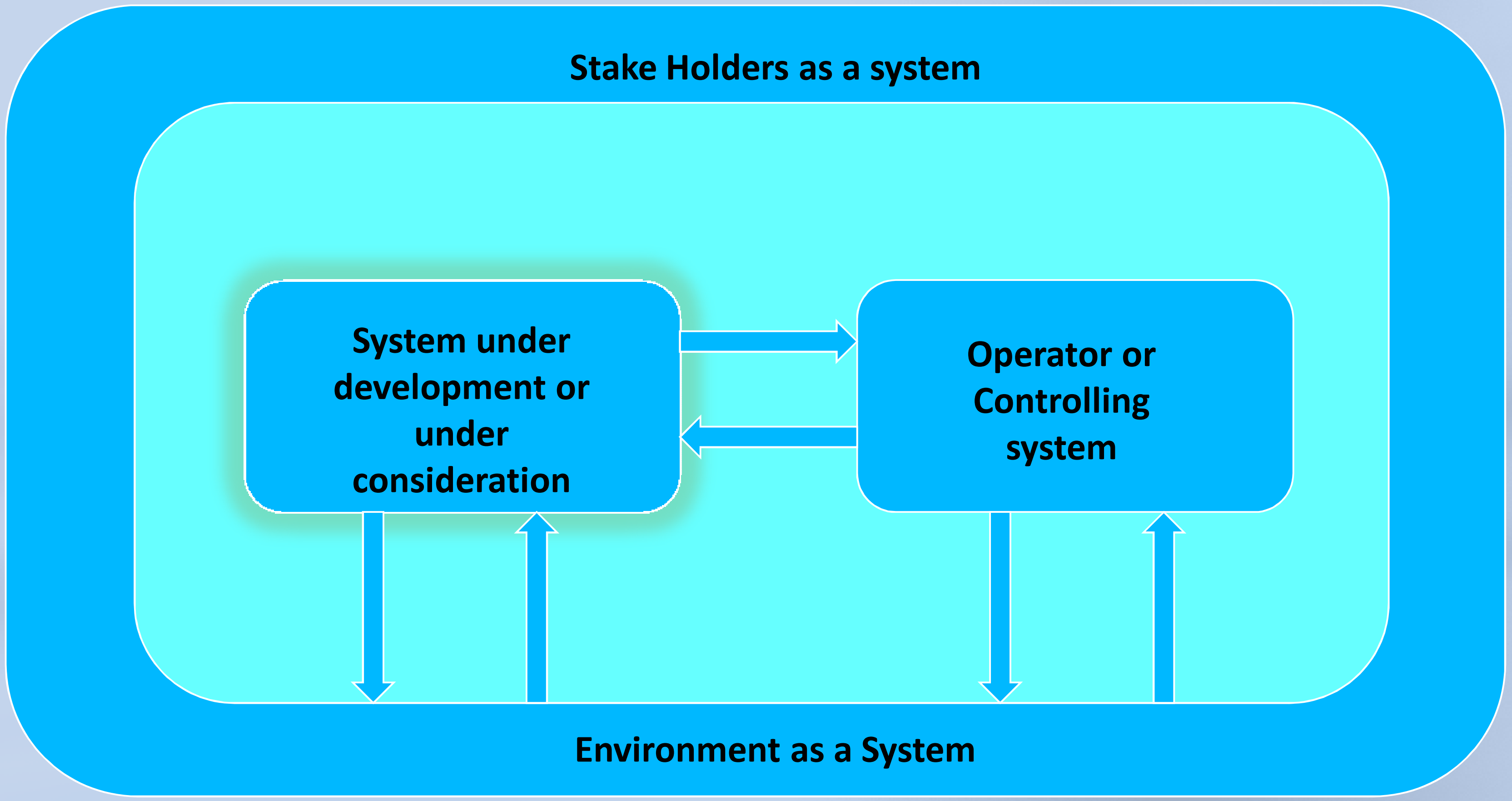
before the term even existed.

What did Wittgenstein really say?

- A **model** is a **projected view**
- A model assumes a **methodology**
  - We can only faithfully make the transition from model to system, if the model is complete
  - also the **implementation is a model**
- We can only faithfully generate the implementation, if we have a completely defined mapping between the model(s) as a set of projected views and the selected implementation.
- Therefore: **Models and Processes are strongly linked**.
- The issue: both are actually very large state spaces!
- **Mastering the complexity** is the challenge!

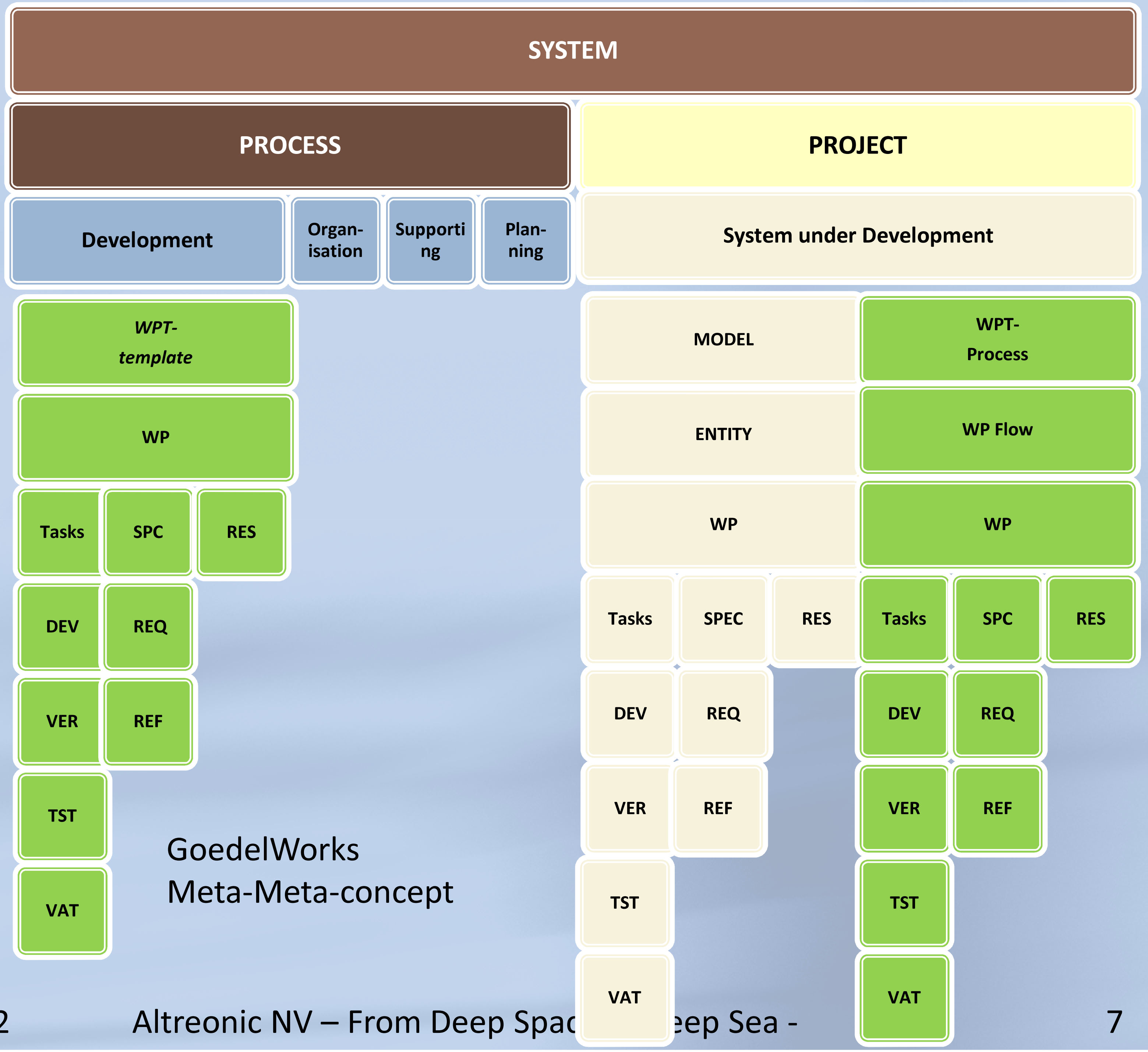
What system?

- Any system is part of a larger system


Stake Holders as a system
System under development or under consideration
Operator or Controlling system
Environment as a System

Systems engineering with just 16 meta-concepts

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



SYSTEM

PROCESS

PROJECT

Development

Organ-isation

Supporting

Plan-ning

System under Development

WPT-template

WP

Tasks

SPC

RES

DEV

REQ

VER

REF

TST

VAT

MODEL

ENTITY

WP

Tasks

SPEC

RES

DEV

REQ

VER

REF

TST

VAT

WPT-Process

WP Flow

WP

Tasks

SPC

RES

DEV

REQ

VER

REF

TST

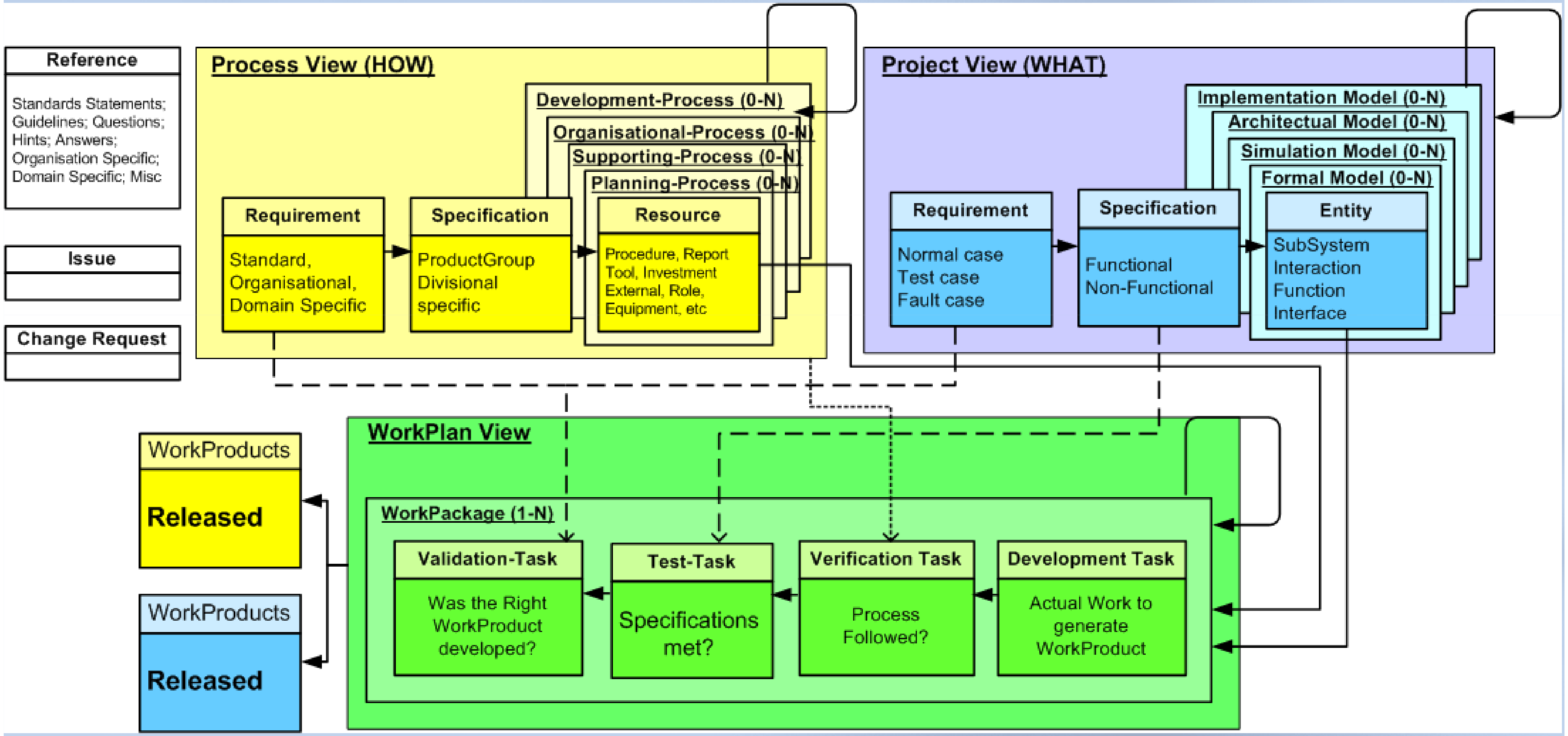
VAT

GoedelWorks
Meta-Meta-concept

Relationships

- Association links:
  - Dependency links:
    - E.g. a SPC depends on REQ (n)
  - Precedence links:
    - A Verification Task preceeds a Test Task
- Structural links:
  - A WP is composed of Tasks (n)
  - A Model is composed of Entities (n)
- Navigation links:
  - Navigation tree for easy access
  - Flow links (next-previous) for defining flows

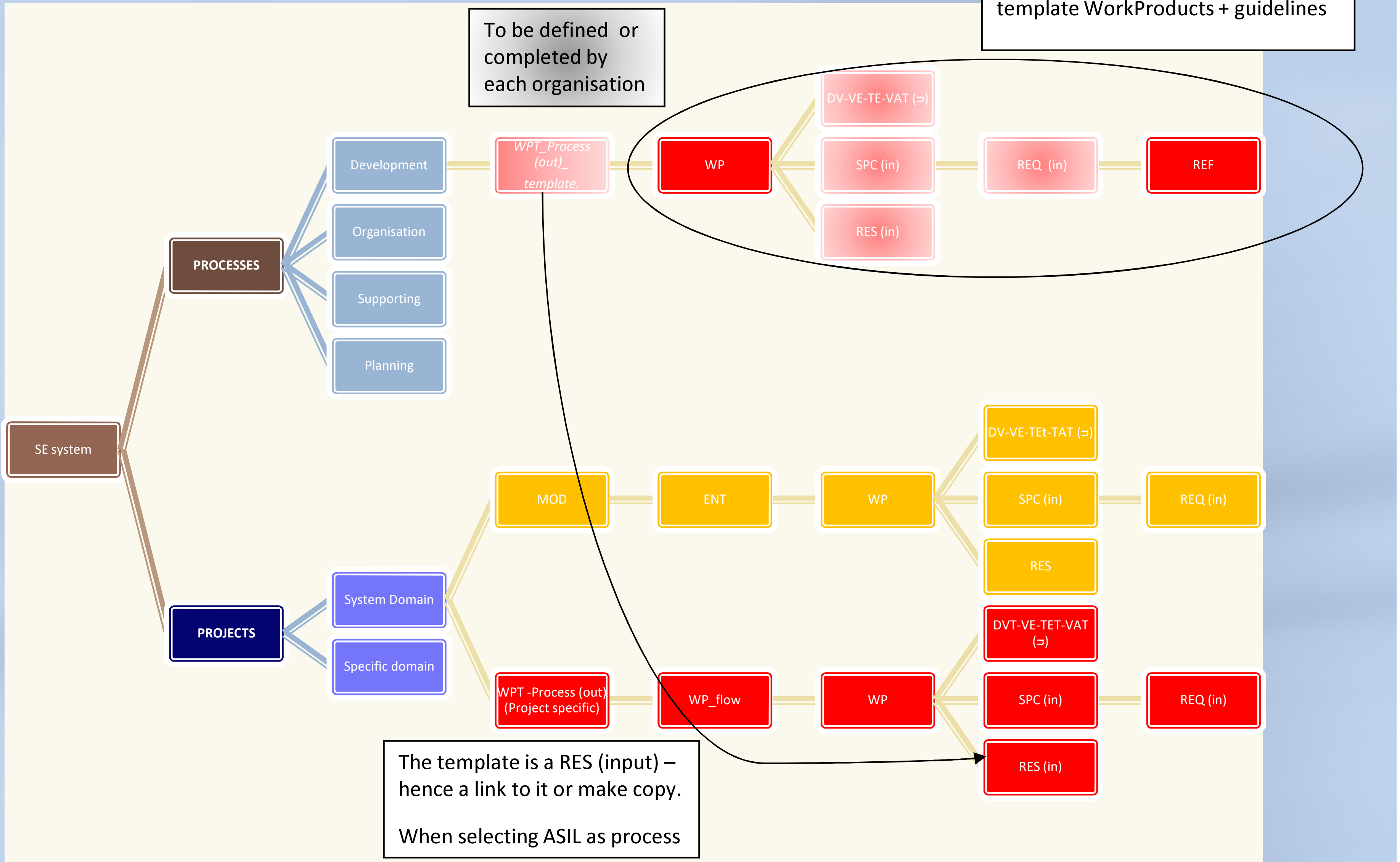
GoedelWorks' (simplified) meta-model


Reference
Standards Statements; Guidelines; Questions; Hints; Answers; Organisation Specific; Domain Specific; Misc
Issue
Change Request
Process View (HOW)
Development-Process (0-N)
Organisational-Process (0-N)
Supporting-Process (0-N)
Planning-Process (0-N)
Requirement
Standard, Organisational, Domain Specific
Specification
ProductGroup Divisional specific
Resource
Procedure, Report Tool, Investment External, Role, Equipment, etc
Project View (WHAT)
Implementation Model (0-N)
Architectual Model (0-N)
Simulation Model (0-N)
Formal Model (0-N)
Requirement
Normal case Test case Fault case
Specification
Functional Non-Functional
Entity
SubSystem Interaction Function Interface
WorkProducts
Released
WorkProducts
Released
WorkPlan View
WorkPackage (1-N)
Validation-Task
Was the Right WorkProduct developed?
Test-Task
Specifications met?
Verification Task
Process Followed?
Development Task
Actual Work to generate WorkProduct

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 a single V-model but 100’s.
- Overall Process follows from respecting states
- WorkProducts morph:
  - Resource at input is always result of previous Project
  - Work Product template => Work Product specific Project
  - System in Project A => component in Project B

From work products to resource



To be defined or completed by each organisation

ASIL developed template flow + template WorkProducts + guidelines

The template is a RES (input) – hence a link to it or make copy.

When selecting ASIL as process

Application and validation importing the ASIL Safety Engineering process

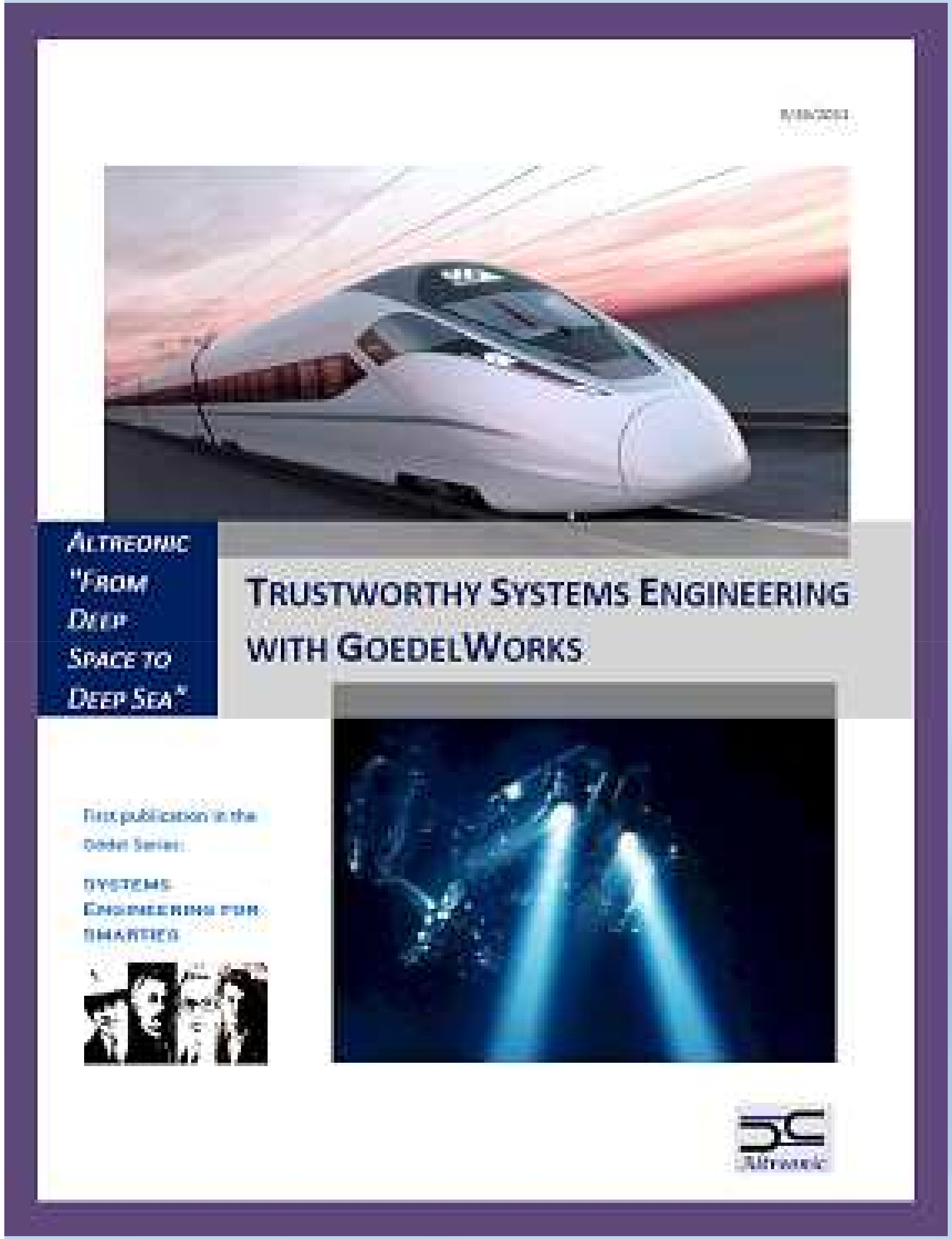
- Input: ASIL project of Flanders Drive
  - **A**utomotive **S**afety **I**ntegrity **L**evel
- Goal: develop a common safety engineering process based on existing standards
- IEC 61508, IEC 62061, ISO DIS 26262, ISO 13849, ISO DIS 25119, ISO 15998, CMMI, Automotive Spice
- Partners:
  - Altreonic, DANA, EIA, Flanders Drive, Punch Powertrain, Triphase, TüV Nord
- DO-178C, DO-254, ARP4761: Altreonic

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
- Flawlessly imported in GoedelWorks portal using meta-model

Conclusion

- Systems engineering process can be formalised using a common metamodel
- Challenges: Integration of different domains
    - Concepts, Architectural design, WorkFlow
    - System Engineering 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 OPENCOSS FP7 project
    - (cover avionics, railway, automotive)
    - DO-178C and DO-254
    - Focus on re-use



[illegible]

ALTREONIC "FROM DEEP SPACE TO DEEP SEA"

TRUSTWORTHY SYSTEMS ENGINEERING WITH GOEDELWORKS

First publication in the Gödel Series:

SYSTEMS ENGINEERING FOR SMARTIES

Altreonic

More info at www.altreonic.com

http://www.altreonic.com/sites/default/files/Systems%20Engineering%20with%20GoedelWorks.pdf