Modelling Large Infrastructure Systems





Who am i:

- Jan de Liefde
- Systems engineer in the infrastructure and industry domain
- Focus on Model Based (Systems) Engineering
- Jan.de.liefde@thecollective.si



















(Large) infrasystems: some characteristics

- High political and public interest (always in the spotlight)
- Long life-cycle (>80 years)
- Civil, Mechanical, Electrical, Industrial automation, etc.
- Direct or indirect impact on the environment (pollution, view, fine dust, etc.)
- Impact from the environment (salt water, wind, temperature, etc.)
- Strong focus on safety (zero casualties) and availability (24/7)
- Many different suppliers







					Management.	System	
Safety	Mechanical	Electrical	Civil	Automation			Etc.
						(T (THE COLLECTIVE the systems integration company

Document centric











Design

Cument Template

ea de nature oficia document; s Sotiviare Development other econtracionifies to

KLARITI.COM

Today issues

- Information duplication
 - In multiple documents same kind of information
 - Information loss during changes
- Access to information
 - Searching within documents
 - Exchange of information by using exports and copy/paste
- Lack of traceability
- Different not connected data sources
- Lack insights in the Impact of changes
- Discipline based



Document centric (hym Stokebule

Model centric

Weggebruiker Grom Stakeholder Rolleri

Desktopmonito ports

block Toetsenbord parts nbord Output



Modelling: Today issues

- High level design
 - Focus mainly on producing documents
 - Focus mainly on functions
- Objects partly modeled (no constraints, properties, etc.)
- Lack of traceability
- No distinction between problem and solution domain (usage is not the same as functionality/solution)
- Language not readable for domain engineers



What do we need?



(TC) THE COLLECTIVE the systems integration company

How: A model and a multi-user workspace!





Model

- Every model should have a goal:
 - The goal of the System model is to be a "single source of truth"
 - The goal of the Library model is to support the reuse of of signals, value types, etc.
 - The goal of the management model is to support the project management processes (for example, planning informatie- and configuratiemanagement

Management

- The goal of the operational model is to define the intended usage of the system
- The goal of the functional model is to define the necessary functionality and systemstructure that is needed to support the intended usage
- The goal of the component model is to define the technical solution



Library

Operational mode

Functional model

Component mode

Model



Library

- Use Standards
- Define Glossary
- Define Acronyms



«GlossaryEntry» architecture view	«GlossaryEntry» anomaly	«GlossaryEntry» agreement	«GlossaryEntry» acquisition	«GlossaryEntry» acceptance testing
work product expressing the architecture of a system from the	Anything observed in the documentation or operation of a system that deviates	mutual acknowledgement of terms and conditions under	process of obtaining a system, product or service	Testing conducted determine whether system satisfies its acceptance criteria
perspective of specific system concerns	from expectations based on previously verified system,	which a working relationship is conducted	«GlossaryEntry» acquirer	and to enable the customer to determine whethe
«GlossaryEntry» architecture	software, or hardware products or reference documents.	EXAMPLE Contract, memorandum of agreement.	Stakeholder that acquires or procures a product or service	to accept the system. (B) Formal testing conducted enable a user, customer, or other
<system> fundamental concepts or properties of a</system>	«GiossaryEntry» architecture viewpoint		from a supplier. Other terms commonly used for an acquirer are buyer, customer	authorized entity t determine whethe to accept a system component.
system in its environment Tra	ceability		▼ 1	×
embodied in its 60° elements, relationships, a the principles o design and evol	architecture	atura description		
«GlossaryEntr	P 100 12010 Picilite	cure description		





Library

- Define Engineering units
- Define Quantity kinds
- Define Value types

ge] Space and Time [Space and T	[ime]
metre	second
quantityKind = symbol = m description = ISO 80000-3 definitionURI =	quantityKind = symbol = s description = ISO 80000-3 definitionURI =
square metre	minute
quantityKind = symbol = m2 description = ISO 80000-3 definitionURI =	quantityKind = symbol = min description = ISO 80000-3 definitionURI =
cubic metre	hour
quantityKind = symbol = m3 description = ISO 80000-3 definitionURI =	quantityKind = symbol = h description = ISO 80000-3 definitionURI =
radian	day
quantityKind = symbol = rad description = ISO 80000-3 definitionURI =	quantityKind = symbol = d description = ISO 80000-3 definitionURI =
degree angle	radian per second
quantityKind = symbol = ° description = definitionURI =	quantityKind = symbol = rad/s description = ISO 80000- definitionURI =
metre per second	radian per second squared
quantityKind = symbol = m/s description = ISO 80000-3 definitionURI =	quantityKind = symbol = rad/s2 description = ISO 80000-3 definitionURI =
metre per second squared	hertz
quantityKind = symbol = m/s2 description = ISO 80000-3 definitionURI =	quantityKind = symbol = Hz description = ISO 80000-



«valueType» MTBF	«valueType» Distance
quantityKind = time unit = hour	quantityKind = distance unit = metre
«valueType» MTTR	
quantityKind = time unit = hour	
«valueType» length	
quantityKind = length unit = metre	
«valueType» breadth	
quantityKind = breadth unit = metre	
«valueType» height	
quantityKind = height unit = metre	



Library

• Define reusable set of Signals

bdd[package] Control interface [Control interface] / «signal» «signal» «signal» «signal» Undervoltage trip **Checkback Not** Manual On/Open Manual Stop Off/Closed command «signal» «signal» «signal» Status discrepance **Checkback Not** Manual Off/Close alarm Closed > Open/ On/Open Off > On «signal» «signal» «signal» Checkback Off/Closed Fault Status discrepance alarm Open > Closed/ On > Off «signal» «signal» «signal» Checkback ON/Open Contact supply voltage Test status











Management information

- Organisation
- Processes and activities
- Products
- Planning
- Project risks



(Engineering) products





Management: Process



Management: RASCI

Target Source	1 Identificeer de belangri	2 Identificeer de Gebrui	3 Identificeer Operatione	4 Identificeer operationel	5 Identificeer Safety Haz	6 Identificeer security be	7 Stel Pre-Operationeel
1 Business analist	responsible >	responsible >	responsible >	responsible >			responsible >
2 Omgevingsmanagemer	consulted >	support >					
з Project managemer	accountable	accountable	accountable	accountable	accountable	accountable	accountable >
4 Safety analist					responsible >		
5 Security analist						responsible >	
6 Systeem architect		support >			support >		support >











Phases and the Systemmodel

- Design phases
 - preliminary design
 - Final design
 - implementation design





Operational model: Process

- Identify Stakeholders
- Identify Stakeholder goals
- Identify System boundary
- Define Capabilities
- Define operational usage
 - Operational scenario's
 - System interaction
- Identify hazards
 - Mitigation
- Identify and define design constraints
 - Safety, usability, environment, maintainability, etc.
- Define acceptance criteria
- Traceability





the systems integration company

Operational model: Stakeholders, Goals & Systems Boundary





Operational model: Operational scenario's

Operational model: Hazards

Operational model: Constraints

- Environmental
- Regulations
- Standards & Guidelines

ProRail

Asse

Ontwerpvoorschrift Tekenvoorschrift OBE- bladen Overzicht - symbolen

> Beherende instantie: AM Architectuur en Techniek Inhoud verantwoordelijke: Manager AM Treinbeveiligingssystemen Status: Definitief

Operational model: Validation

- Define validation criteria
 - Define for every goal a validation criteria
 - Define for every capability a validation criteria

Operational Model: Review

🙆 🖋 📋

R

 Review Operational Model with relevant Stakeholders

Review Dashboards in Prolaborate

Archived

Jan de Liefde

: 😭 1

1

: Incomplete

Author

Items

Contributors

My Status

	# ② 🛢 Demo ∽		🔁 Dia	igrams - Search	् 🛛 🌲 🜆		
	Review Verificatie & Validatie Archived			🚳 Dashboa	ard III Items 🕈 Action 👻		
	🚯 Moderator Dashboard (Default				֎ Moderator Dash ▼		
	Author: 🎬 Jan de Liefde			Ma	oderator		
iel with	Total Items 1 Total Items 1 Total Discussions 1 Description Review	Approvers	Reviewers	Review Dates : ☆ 01-0- Approval Dates : ☆ 06-0: Moderators : ☆ 1	4-2020 - ☆0-04-2020 5-2020 - ☆08-05-2020		
	Status of Contributors			Search:			
	El Diagrams - Search Q Q	TR.			Previous 1 Next		
		Closed Discussions	Review Done	Approved	Rejected		
		0	0	0	0		
My Reviews Inc	lude Archived Reviews Status All 🗸 🎹	0	U	U	U		
æ 🖉	æ / i						
Review Verificatie & Validatie plan	Review Verificatie & Validatie				Desidence of Next		
Author : 🔛 Jan de Liefde	Author : 🔙 Jan de Liefde				Previous 1 Next		
Contributors : 📽 1	Contributors : 📽 3	Closed Discussions	Review Done	Approved	Rejected		
Items : 1 My Status : Incomplete	Items : 🖺 1	U	U	U	0		
Archived	Archived						

Review Kwaliteit Management Plan

O Not Yet Started

📓 Jan de Liefde

: Not Yet Started

: 🗈 1

🕋 🕘 🛢 Demo 🗸

🚳 🖋 📋

TC] Collective

a.

 $\langle \rangle$

868

Manage Reviews

Author

Items

My Status

List of Reviews

56 Days To Start

Contributors : 📽 1

Operational model

CL01	.TEC.02 Stakeholder Needs and Requirements Evaluation
	Correctness:
	Verify and validate that the requirements satisfy the stakeholder needs for the system.
	Verify that the requirements comply with standards, references, regulations, policies, physical laws, and business rule
	Validate that the requirements define the intended interaction of the system with its operating environment and oth interfacing systems.
	Validate the suitability of the preferred candidate solution(s) to meet the stakeholder requirements.
	Consistency:
	Verify that all terms, concepts, and requirements are documented consistently in accordance with accepted syntax a structure (e.g., style guides and requirements modeling structure).
	Verify that there is consistency between assumptions, requirements, and between groups of requirements.
	Completeness:
	Validate that performance criteria and functionality are described in the requirements, within the assumptions and constraints of the operating environment and system boundaries.
	Verify that all stakeholders, or stakeholder classes (used here as groupings of stakeholders), are identified.
	Verify that the stakeholder requirements satisfy specified configuration management procedures. Verify stakeholder requirements are in a form suitable for requirements management throughout the life cycle.
	Readability:
	Verify that the documentation is legible, understandable, and unambiguous to the intended audience.
	Verify that the documentation defines all acronyms, mnemonics, abbreviations, terms, and symbols.
	Testability:
	Verify that objective acceptance criteria can be developed to validate the requirements.

Traceabil	ity	construct Standaards
	or ⊘	ISO 29148 Requirement engineering
66° 📑 🔛 ⊿ 🖹 CL01.T ⊿ 🕶 de	IEC.02 Stakeholder Needs ar epends on	Systems and Software engineering - Life cycle pro Requirement engineering
⊳ 🔳	1012-2016 IEEE Standard fo	r System
	TEC.02 Stakeholder require	Software, systems and enterprise — Architecture description
гU		ТМАР
		EIA-IEEE J-Std 016-1995 Software Development Acquirer-Supplier Agreement free .pdf
		1012-2016 IEEE Standard for System, Software, and Hardware Verification and Validation

Operational model

- Stakeholder identified
- Goals clear V
- System capabilities defined
- Operational scenario's defined V
- System interaction with users and external systems identified
- Hazards identified
- Constraints identified
- Design decisions documented
- Acceptance criteria defined
- Traceability guaranteed

Functional model: Process

- Define system architecture
 - Define functional and physical boundary
 - Define functions
 - Define system elements
 - Define internal interfaces
 - Define interface protocols
 - Allocate functions to system elements
 - Allocate information to detailed interfaces
 - Define system element parametrics (power consumption, availability, heat dissipation, stability, etc)
 - Define constraints
 - Define testcases
 - Traceability

Functional model

System elements

Interfaces deployment Standard and information object mapping of PUC05 «Data Model .. Modbus bdd[package] Waterkering [OSMA_ARCH Logisch Systeem BWK] «Information Object» Grid Schedule «Information Object Flow» 0 WSE)Mayside Equipment «block» (S&C) «block» Signalling & Control Veiligheidssensor DMS Scada «Information Object» Dispatch request «Information Object» Island Interruption Alarm «Information Object Flow» «Information Object Flow» «Information Object Flow» «Information Object» Charging/ «Information Object» Switch Position «Information Object Flow» Command Acknowledgement, «Information Discharging Schedule, «Information Object» Charging / Discharging Schedule Object» Switch Position Command Acknowledgement «Information Object» Charging/ Discharging Schedule «defines» «block» «Data Model . Waterkering Information Object Flow IEC 61968 namespace «defines «block» nformation Object Flow» Kering «Information Object» Charging/ «block) Discharging Schedule namespace Besturing «Information Object Flow» Acknowledgement «defines» «Information Object» Switch namespace «block» «block» Position Command «Information Object Grendel Sensor «block» «Information .. Acknowledgement «Information Object» Switch Charging/ Overbruggingsschakelaar Grid Position Command Discharging Schedule «Information Object Flow» Schedule Acknowledgement TIT «Information ... BWK: CBI Charging/ Switchgear Module Discharging Schedule «Data Model. IEC 60870-5-104 «defines» \Box «defines» «defines» «Information Object «Information ... Switch Position Switch Command Position Acknowledgement Command «block» Automation system <u>A</u>O-O **TC** S&C autorisatie: Autorisatie

Communication

Integration

Constraints from

- Implementation
- Testing
- Transition
- Maintenance
- Logistic
- etc

Failure Mode & Effect Analysis (FMEA)

Verification

- Review functional model
- Define Testcases and Testcriteria

Review

🕋 🕘 🛢 Demo 🗸

🚳 🖋 📋

Author

Items

My Status

Contributors : 😭 1

 Review Operational Model with the Stakeholders

Review Dashboards in Prolaborate

Archived

Jan de Liefde

1

: Incomplete

🚳 🖋 📋

Review Verificatie & Va

*

1

: Incomplete

Jan de Liefde

Author

Items

R

Contributors

My Status

				agrams - Search	Q 🔮 🐥
	Review Verificatie & Validatie Archived			🙆 Dashboa	rd 🎫 Items 🛉 Action
	noderator Dashboard (Default)				🚳 Moderator Dash 👻
т Г.	Author: 🌃 Jan de Liefde			Mo	derator
th	Total Items	Approvers	1 Reviewers	Review Dates : 101-0. Approval Dates : 106-09 Moderators : 111	4-2020 - ☆ 30-04-2020 5-2020 - ☆ 08-05-2020
	Description Review				
	Status of Contributors				
	Show 10 ¢ entries	_		Search:	Previous 1 Next
	🗈 Diagrams - Search 🔍 🕄 🌲 🌆	Closed Discussions	Review Done	Approved	Rejected
	+ Create Review	0	0	0	0
		0	0	0	0
My Reviews	clude Archived Reviews Status All 🗸 🎫				
My Reviews In	Dude Archived Reviews Status All				
My Reviews In	Review Verificatie & Validatie				Previous 1 Next
My Reviews In	Review Verificație 8 Validație Author :	Closed Discussions	Review Done	Annovad	Previous 1 Next
My Reviews In	Clude Archived Reviews Status All	Closed Discussions	Review Done 0	Approved 0	Previous 1 Next Rejocted 0
My Reviews In	Contributors : 2 3 Items : 1	Closed Discussions 0	Review Done 0	Approved 0	Previous 1 Next Rejected 0

Review Kwaliteit Management Plan

O Not Yet Started

📓 Jan de Liefde

: Not Yet Started

1

TC] (TC)

a.

0

 $\langle \rangle$

868

Manage Reviews

List of Reviews

56 Days To Start

Contributors : 😭 1

Author

Items

My Status

Review checklists

bdd[package] TEC.04 Architecture Definition V&V [TEC.04 Architecture Definition V&V]

CL01.TEC.04 Architecture Evaluation

Correctness:

- 1) Verify the characteristics, attributes, constraints, and functional and performance requirements of the selected architecture(s) correctly implement the system requirements.
- 2) Verify the selected architecture(s) complies with standards, regulations, policies, physical laws, and business rules.
- 3) Validate the product solution(s) defined by the system architecture(s) satisfies the stakeholder needs.
- 4) Validate that the selected architecture(s) and its element interactions do not result in unnecessary, unintended, or deleterious consequences.
- 5) Validate the selected architecture(s) defines the intended interaction of the system with its operating environment.
- Consistency:
- 1) Verify the selected architecture(s) conforms to the architectural guidance, principles, and tenets of the organization processes and procedures (e.g., serviceoriented architecture, modular open-systems architecture).
- 2) Verify the architectural principles, characteristics, and rules established for the system are being applied across the selected architecture(s).
- Completeness:
- 1) Verify the system functions are allocated to the elements of the selected architecture(s).
- 2) Verify all system requirements are included in the selected architecture(s).

CL01.TEC.03 Requirements Evaluation

Correctness:

Urerify and validate that the required characteristics, attributes, constraints (e.g., mechanical, electrical, mass, thermal, data, procedural flows), and functional and performance requirements for a product solution are correct (e.g., searity, ergonomics, human-machine interface, safety, reliability, maintainability, response time). Verify and validate that the system requirements satisfy the stakeholder requirements. Verify that the system requirements comply with standards, references, regulations, policies, physical laws, and business rules. Validate that the system requirements define the intended interaction of the system with its operating environment and other interfacing systems. Consistency: Verify that all terms, concepts, and requirements are documented consistently in accordance with accepted syntax and structure (e.g., style guides and requirements modeling structure). Verify that there is consistency among the requirements, groups of requirements (functional interaction), and assumptions. Completeness: Validate that all stakeholder needs are satisfied by the set of system requirements. Validate that performance criteria and functionality are described in the requirements, within the assumptions and constraints of the operating environment and system boundaries. Verify that the System requirements satisfy specified configuration management procedures. Verify that system requirements are in a form suitable for requirements management throughout the life cycle. Readability: Verify that the documentation is legible, understandable, and unambiguous to the intended audience. Verify that the documentation defines all acronyms, mnemonics, abbreviations, terms, and symbols. Testability: Verify that objective acceptance criteria can be developed to validate the requirements.

CL02.TEC.04 Interface Analysis

system elements.

1) Verify the system architecture satisfies the system interface requirements between

2) Validate the system architecture satisfies the system interface requirements with

1) Verify the architecture describes all internal and external interfaces.

external systems (external system boundaries).

Correctness:

Completeness:

Functional model

- Functions defined
- System structure defined
- Interfaces including information defined
- Communication protocols defined
- Testcases defined
- Hazards identified and mitigated
- Constraints identified and implemented
- Parametrics defined and verified
- Design decisions documented
- Traceability guaranteed
 - Within the System model
 - Other sources (drawings, simulations results, etc.)

Component model: Process

- Define instances of Systemelements
- Define relations between instances

Traceability

66 🚰 韓 影 - 🥘 - A instance of 🖌 🗐 Lift type 1 ✓ ➡ subtype of ⊿ 🗐 Lift

> 🕨 🖛 supertype of subtype of • embeds p 🖛 owns Association from Association to

▲ ← composed of

* I

Component model

Instances are defined as part of the model

- All Locations (buildings, rooms, etc.)
- All Equipment (Pumps, ventilators, etc.)
- All Automation (sw blocks, interlocks, inputs/outputs, etc.)
- All Relations between locations, equipment, sw blocks, test-/inspections
- All Cable & Pipes
- All Civil parts
- Cable and piping routing is defined
- All Integrations plans
- All Test-/Inspection plans & reports
- Traceability guaranteed
 - Within the model
 - To other data sources (CAD drawings, 3D visualisations, etc)

Implementation design

- Detail civil drawings
- Cabinet drawings
- Cable routing
- Piping routing
- Software design
- etc

Guarantee the traceability between the implementation design en the componentmodel

