

Arcadia and Capella: Augmenting requirements with models to improve the articulation between systems engineering levels and optimize V&V practices

INCOSE SYMPOSIUM 2019
MBSE BEST PAPER AWARD

Stéphane Bonnet, Jean-Luc Voirin, Juan Navas
THALES



1. Preamble: Arcadia method and Capella workbench

2. Model elements ARE requirements

3. Contracts between engineering levels: workflow

4. (Happy) consequences on V&V and incremental development strategy

5. Instantiated workflow



1. Arcadia and Capella

Methodology and
high level concepts
and viewpoints



Purpose-built to
provide the
notation and
diagrams fitting the
Arcadia approach

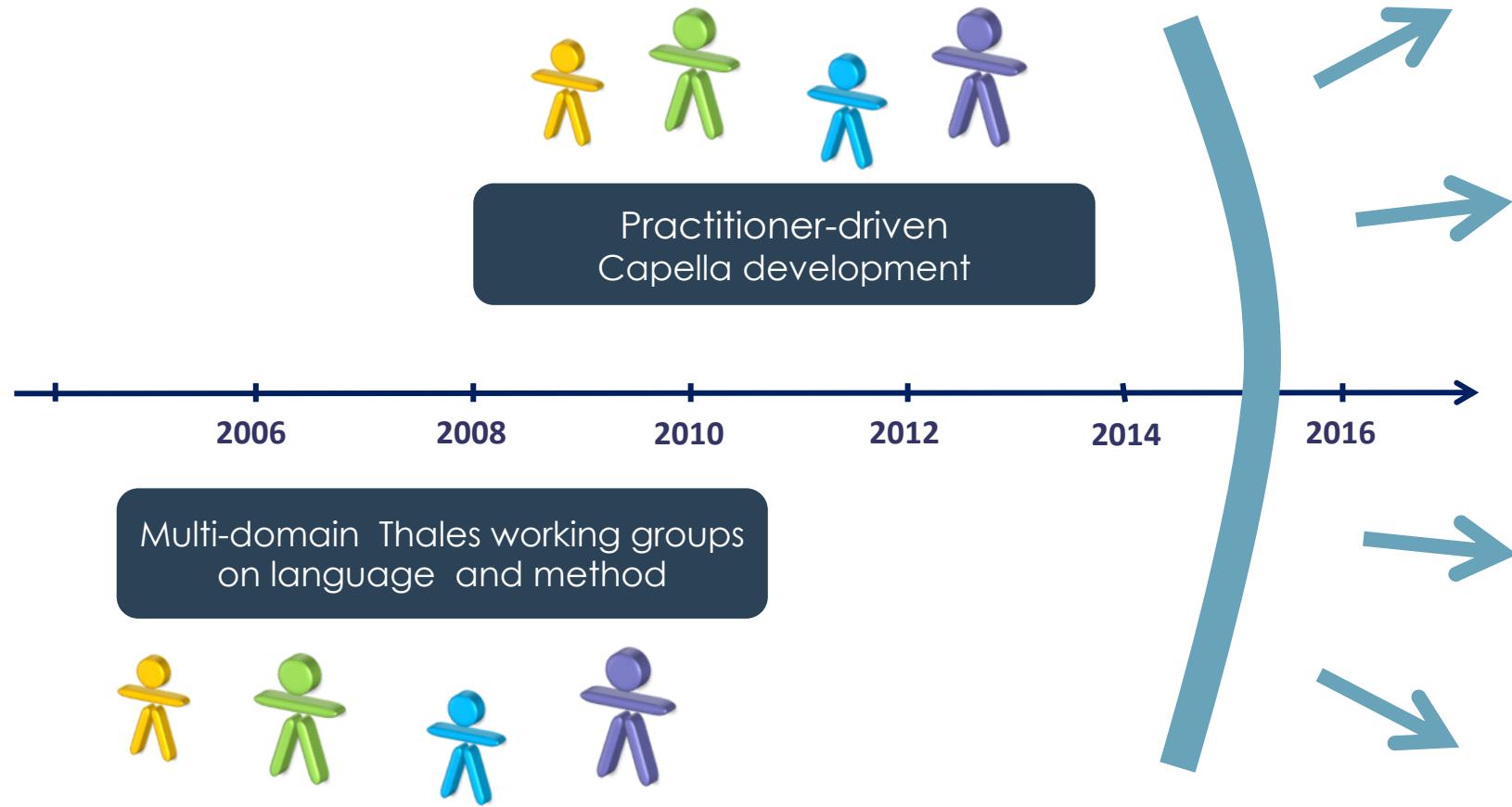


METHOD STEPS	TASKS	SAMPLE MODEL	CONCEPTS	DESCRIPTION MEANS
Customer Operational Need Analysis <small>What the users of the system need to accomplish</small>	<ul style="list-style-type: none"> ✓ Define operational capabilities ✓ Perform an operational need analysis 		<ul style="list-style-type: none"> - Operational capabilities - Actors, operational entities - Actor activities - Interactions between activities & actors - Information used in activities & interactions - Operational processes chaining activities - Scenarios for dynamic behaviour 	
System/ SW/HW Need Analysis <small>What the system has to accomplish for the Users</small>	<ul style="list-style-type: none"> ✓ Perform a capability trade-off analysis ✓ Perform a functional and non-functional analysis ✓ Formalise and consolidate requirements 		<ul style="list-style-type: none"> - Actors and system, capabilities - Functions of system & actors - Dataflow exchanges between functions - Functional chains traversing dataflow - Information used in functions & exchanges, data model - Scenarios for dynamic behaviour - Modes & states 	
Logical Architecture Design <small>How the system will work so as to fulfill expectations</small>	<ul style="list-style-type: none"> ✓ Define architecture drivers and viewpoints ✓ Build candidate architectural breakdowns in components ✓ Select best compromise architecture 		<p>SAME CONCEPTS, PLUS :</p> <ul style="list-style-type: none"> - Components - Component ports and interfaces - Exchanges between components - Function allocation to components - Component Interface Justification by functional exchanges allocation 	
Physical Architecture Design <small>How the system will be developed & built</small>	<ul style="list-style-type: none"> ✓ Define architectural patterns ✓ Consider reuse of existing assets design a physical ✓ Design a physical reference architecture ✓ Validate and check it 		<p>SAME CONCEPTS, PLUS :</p> <ul style="list-style-type: none"> - Behavioural components refining logical ones, and implementing functional behaviour - Implementation components supplying resources for behavioural components - Physical links between implementation components 	
Development Contracts <small>What is expected from each designer/ sub-contractor</small>	<ul style="list-style-type: none"> ✓ Define a components IVVQ strategy ✓ Define & enforce a PBS and component integration contract 		<ul style="list-style-type: none"> - Configuration items tree - Parts numbers, quantities - Development contract (expected behaviour, interfaces, scenarios, resource consumption, non-functional properties...) 	

METHOD STEPS	TASKS	SAMPLE MODEL	CONCEPTS	DESCRIPTION MEANS
Customer Operational Need Analysis What the users of the system need to accomplish	<ul style="list-style-type: none"> ✓ Define operational capabilities ✓ Perform an operational need analysis 		<ul style="list-style-type: none"> - Operational capabilities - Actors, operational entities - Actor activities - Interactions between activities & actors - Information used in activities & exchanged between them - Operational processes - Functionalities & dynamic behaviour 	 Dataflow: functions, op. activities interactions & exchanges
System/ SW/HW Need Analysis What the system has to accomplish for the Users	<ul style="list-style-type: none"> ✓ Perform a capability trade-off analysis ✓ Perform a functional and non-functional analysis ✓ Formalise and consolidate requirements 		<ul style="list-style-type: none"> - Functions or system & actors - Dataflow exchanges between functions - Functional chains traversing dataflow - Information used in functions & exchanges, data model - Scenarios for dynamic behaviour - Modes & states 	 Scenarios: actors, system, components interactions & exchanges
Logical Architecture Design How the system will work so as to fulfill expectations	<ul style="list-style-type: none"> ✓ Define architecture drivers and viewpoints ✓ Build candidate architectural breakdowns in components ✓ Select best compromise architecture 		<p>SAME CONCEPTS, PLUS :</p> <ul style="list-style-type: none"> - Components - Component ports and interfaces - Exchanges between components - Function allocation to components - Component Interface Justification by functional exchanges allocation 	 Functional chains, operational processes through functions & op. activities
Physical Architecture Design How the system will be developed & built	<ul style="list-style-type: none"> ✓ Define architectural patterns ✓ Consider reuse of existing assets design a physical ✓ Design a physical reference architecture ✓ Validate and check it 		<p>SAME CONCEPTS, PLUS :</p> <ul style="list-style-type: none"> - Behavioural components refining logical ones and implementing functional behaviour - Implementation components supplying resources for behavioural components - Physical links between implementation components 	 Component wiring: all kinds of components
Development Contracts What is expected from each designer/ sub-contractor	<ul style="list-style-type: none"> ✓ Define a components IVVQ strategy ✓ Define & enforce a PBS and component integration contract 		<ul style="list-style-type: none"> - Configuration items tree - Parts numbers, quantities - Development contract (expected behaviour, interfaces, scenarios, resource consumption, non-functional properties...) 	 Allocation of op.activities to actors, of functions to components, of behav.components to impl.components, of dataflows to interfaces, of elements to configuration items

NEED SOLUTION

A practitioner-driven journey started in Thales...



... and now continuing beyond

AIRBUS

THALES

INVAP

ThalesAlenia
Space
a Thales / Leonardo company

{PRFC}

ingeliance

CS

V

BOMBARDIER

arianeGROUP

esa

AIRBUS
DEFENCE & SPACE

KTH
VETENSKAP
OCH KONST

IRT
SAINT-EIUPERY+

LGM
INGENIERIE

elements

ZENITH

cnes
CENTRE NATIONAL D'ÉTUDES SPATIALES

indra

SAFRAN

HaCon

OLD DOMINION
UNIVERSITY

pure-systems

tecnalia
Inspiring business

SpaceWorks

ThalesRaytheonSystems

HARRIS
TechnicAtome

Concordia
UNIVERSITÉ
CONCORDIA

Shift2Rail

SAMARES
ENGINEERING
Accelerate Systems Design

ALL4TEC

CNXMOTION

ROLLS
ROYCE

Grus

SystemX
INSTITUT DE RECHERCHE
TECHNIQUE

SUTA
SINGAPORE UNIVERSITY OF
TECHNOLOGY AND DESIGN

Inria

ISAE
École Supérieure de l'Aéronautique et de l'Espace

iav
automotive engineering

Continental
The Future in Motion

alTRAN

SIEMENS

assystem

Oortal
Technologies

SUT

ENSMA

PUBLIC
TRANSPORT
VICTORIA
PT>

TEKFER

EASY
MILE

dyson

Orbital
Technology

U
IVERSITÉ
PARIS-SACLAY

OBELO

OIP
Sensor Systems

exprivia | ITALTEL

DIGINEXT
be visionary

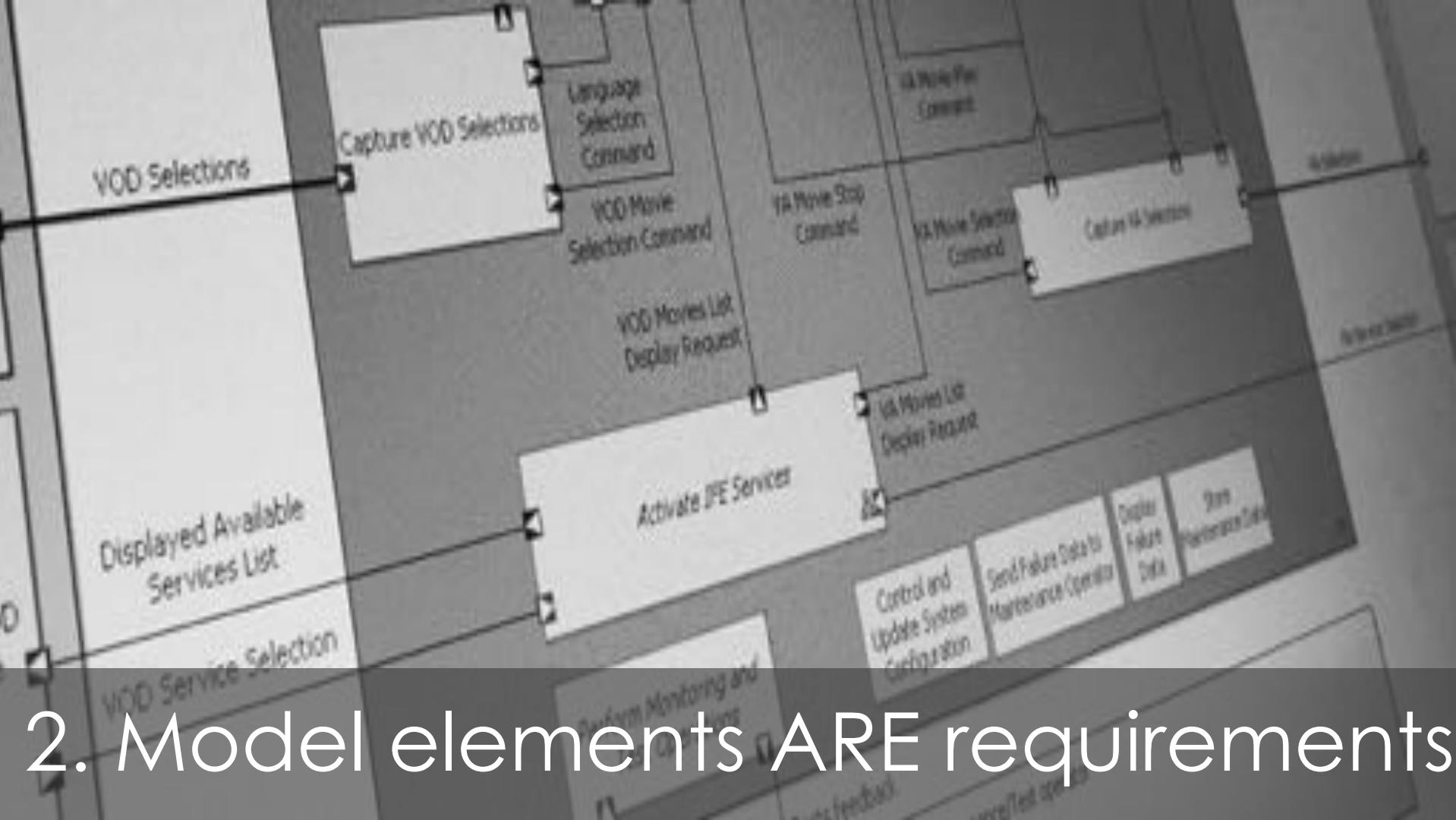
SyEntiveGroup

Escuela Tecnológica
Instituto Técnico Central
Establecimiento Politécnico de Educación Superior

SPHEREA

iid
Institut für Bahntechnik GmbH

THALES



2. Model elements ARE requirements

Need model

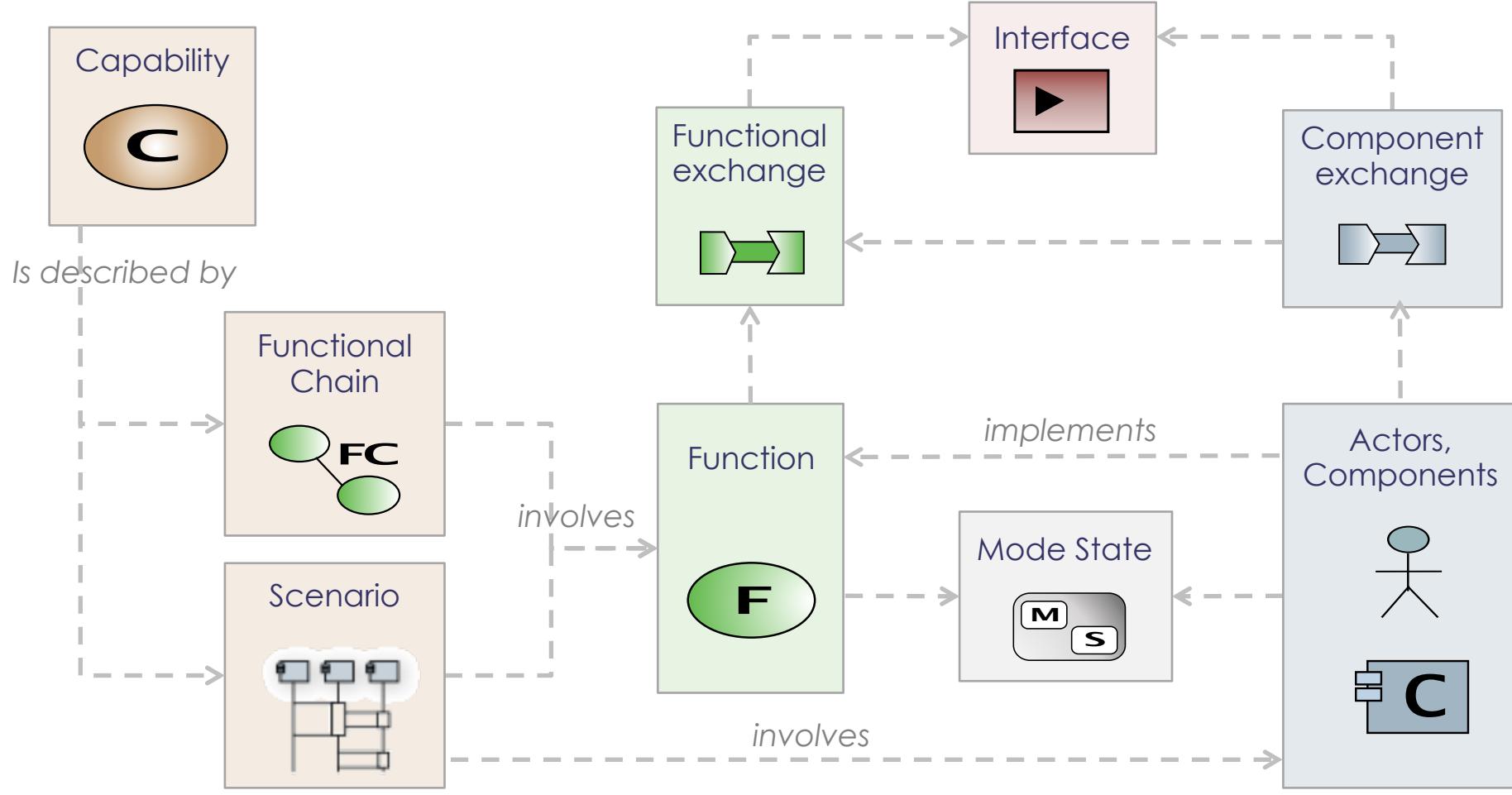
helps formalize and
consolidate
customer and
system requirements

Textual requirements

are at the heart of
the current
engineering
practices

Solution model

helps validate
feasibility,
elicit/justify new
requirements for the
system/subsystems

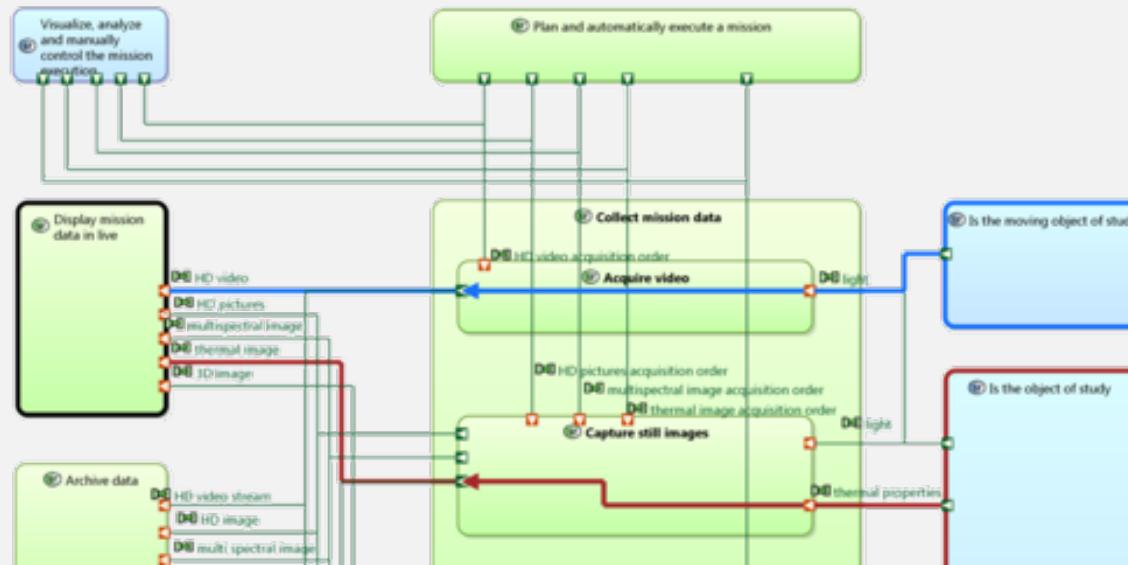


C

Visualize data in live during flight



- Display acquired HD video in live ■
- Display multi-spectral image in live
- Display thermal image in live ■
- Visualize all collected mission data
- Visualize substance level in live

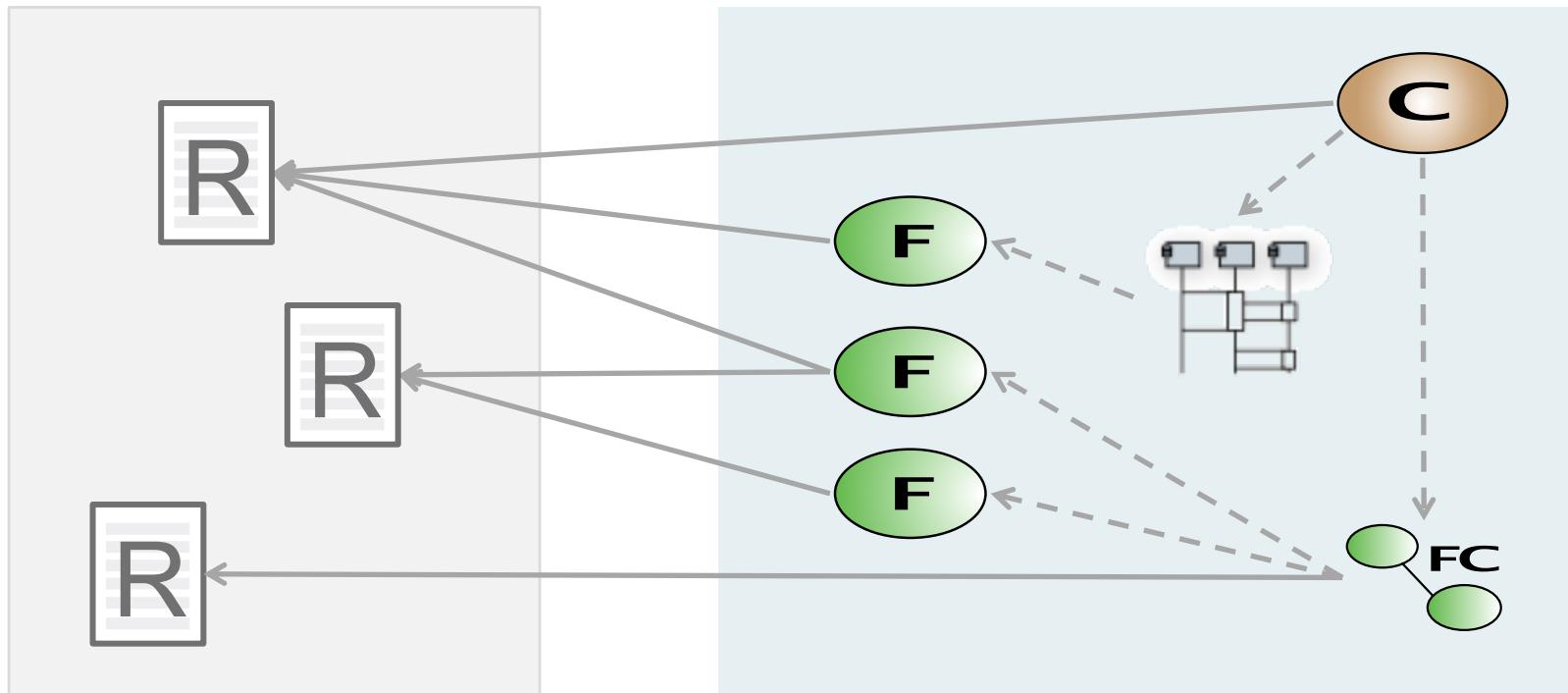


Models add rigor to need expression / solution description

Models enable automated processing

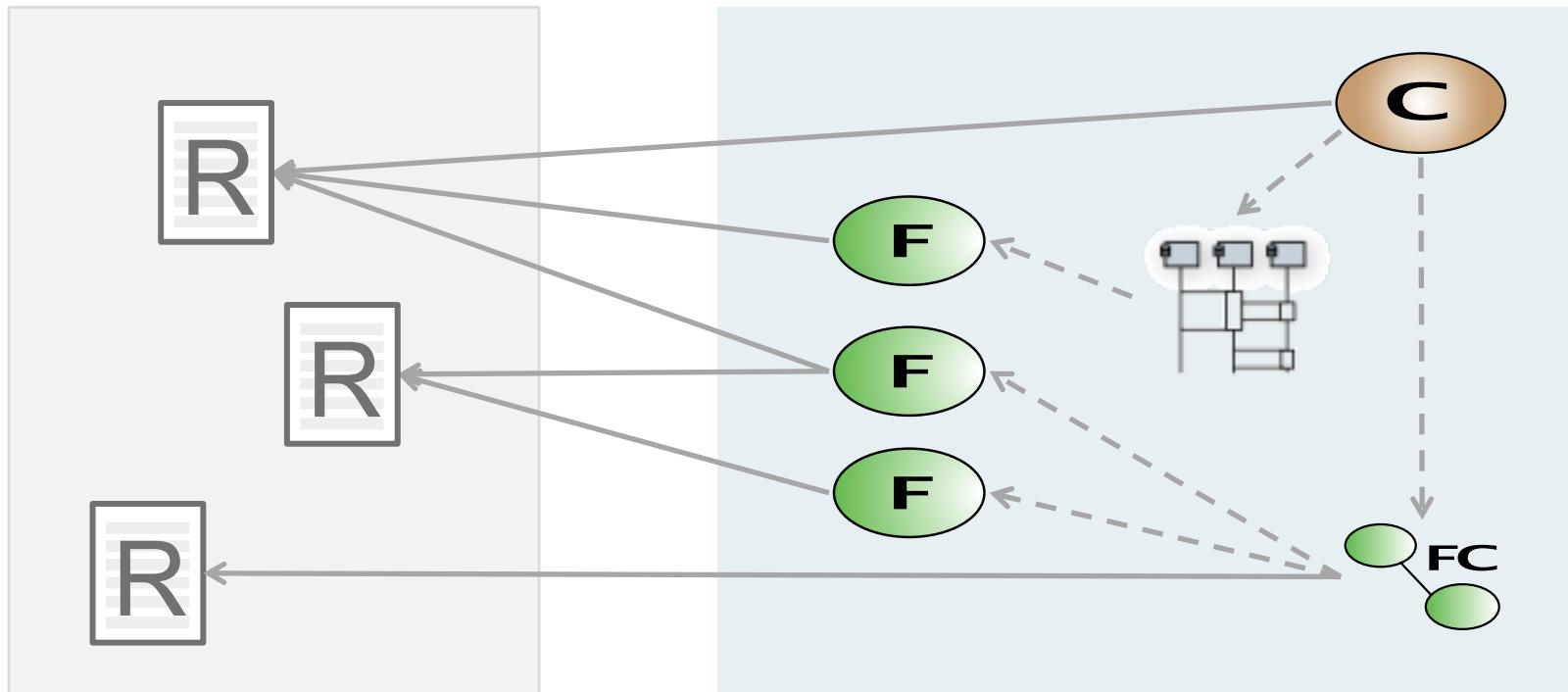
Requirements

Model elements

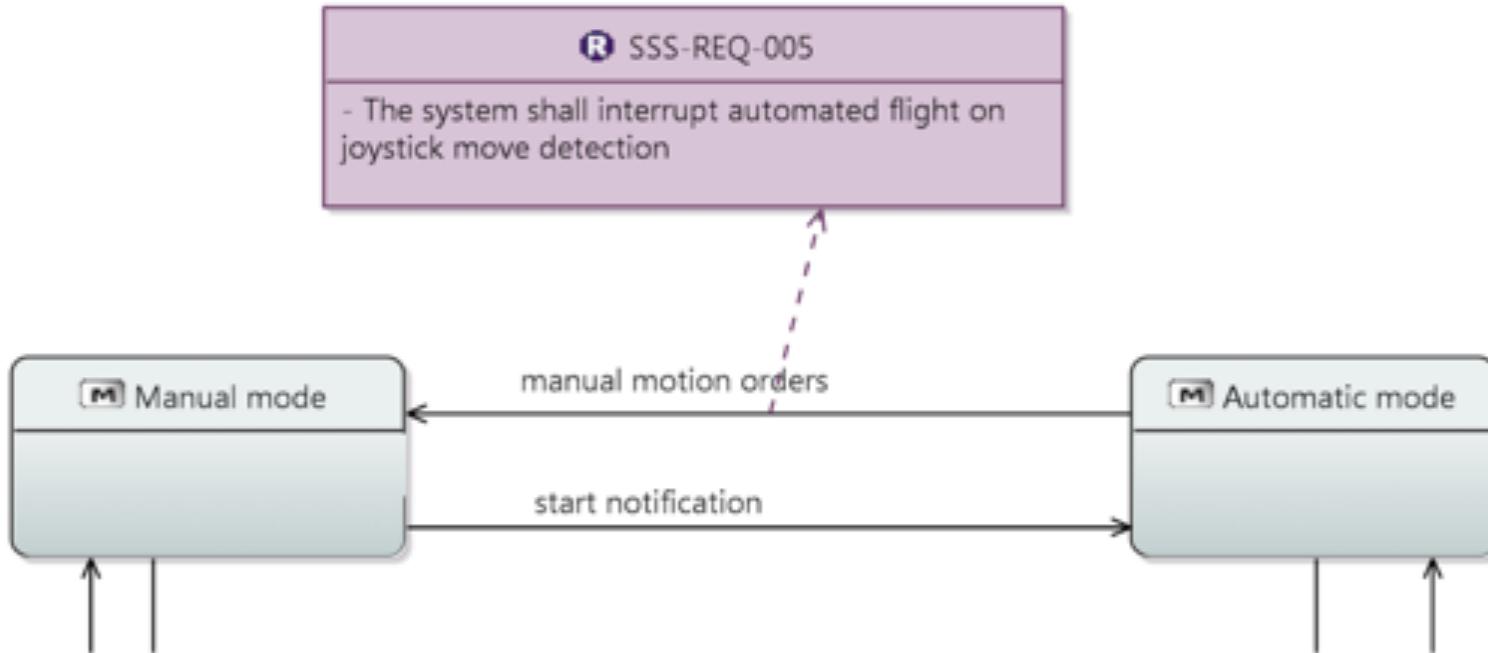


Textual Requirements

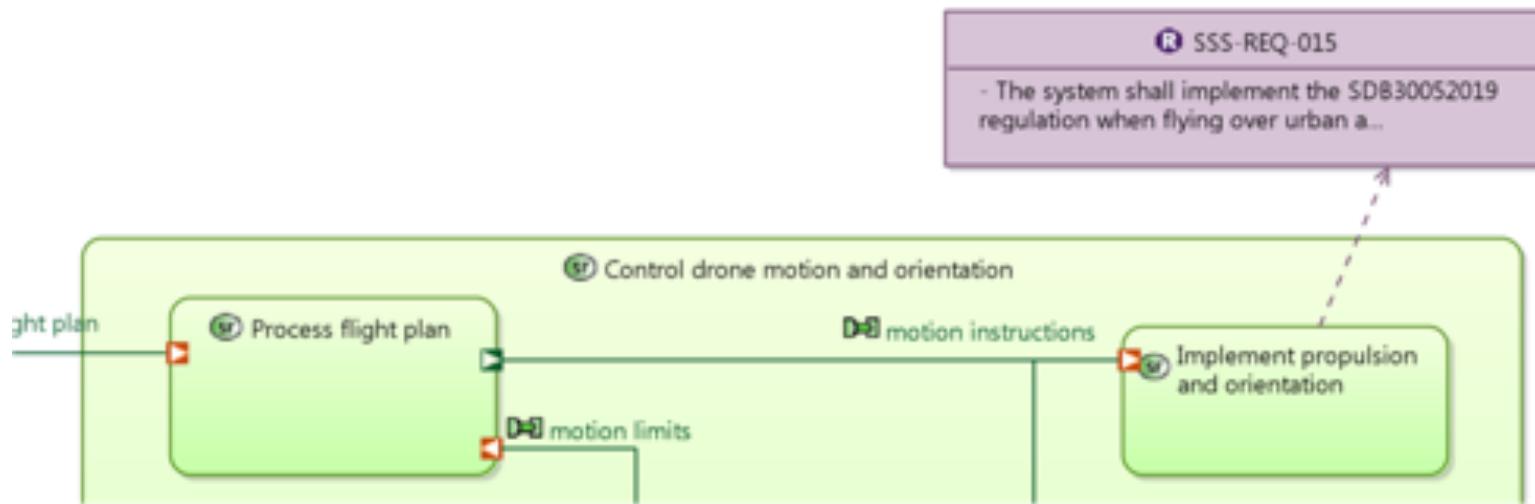
Model elements Requirements



A model requirement can formalize a textual requirement
and explicit its effects and ramifications



Some expectations (environmental, regulations, etc.)
are easier to express with textual descriptions.



Some expectations on a model element at a given engineering level do not require a formal modeling (which is left to subsystem design)





Tooling support

Coupling models and textual requirements

RAT (The Reuse Company) authoring in Capella

The screenshot shows the Capella tool interface with the following details:

- Title Bar:** RAT (The Reuse Company) authoring in Capella
- Menu Bar:** File, Edit, Design, Navigate, Dash, Project, Run, Help, Window, Help
- Project Explorer:** Shows the "Capella Project Explorer" with nodes like "In-Flight Entertainment System", "In-Flight Entertainment System (derived)", "In-Flight Entertainment System (refined)", "System Requirements", "System Functions", "Requirements", "Capabilities", "Interfaces", "Data", "System Context", "System", "Actors", and "Messages".
- Central View:** Displays a UML-like diagram titled "TMC8 Capabilities Center". It includes an actor "User" connected to a boundary object "Provide Video Streaming Services" (represented by a purple rounded rectangle), which in turn connects to a boundary object "Provide Audio and Video Intercommunication Services" (represented by an orange circle).
- Toolbars:** Standard Capella toolbars for selection, creation, and modification.
- Right Panel:** Shows a tree view of "Elements" under "Tools", including "Actor", "Message", "Capability", "Requirement", "Function", "Interface", "Data", "System Context", "System", "Actor", and "Message".
- Bottom Panels:**
 - New - Requirements Authoring Tool:** A floating window with tabs "New", "Suggestions", "View", and "Log". It displays the message "RAT Plugin for Capella" and "Authoring with pattern: Stakeholder Function requirement".
 - Requirements View:** A table showing requirements:

Requirement ID	Description	Status
RQ001	The Account shall provide video streaming services.	Pending Review
 - Metrics Summary:** A table showing quality metrics:

Metric	Value
RQ001 Completeness - TMC - Test length (seconds)	100
RQ001 Uniqueness - OT Language - Test grade (Percent)	100

Capella in Siemens TeamCenter via Obeo SMW

The screenshot displays two windows from the Siemens TeamCenter environment.

Left Window: A 'TMM' (Traceability Matrix) view titled 'Passenger can watch movies during the flight'. It shows a hierarchical structure of requirements:

- Root requirement: Passenger can watch movies during the flight.
- Sub-requirements:
 - After an interruption, movie resumes.
 - Passenger can watch movies during the flight.
- Details for 'Passenger can watch movies during the flight':
 - Requirement ID: RQ00004
 - Description: After an interruption, movie resumes.
 - Notes: In the case of an interruption (paused movie), after an interruption, the passenger can resume watching with automatically movie where it was interrupted.

Right Window: An 'Active Workpage' titled 'Passenger can watch movies during the flight'. It shows the requirement details in a structured format:

Architecture	Documentation	Overview	Parameters	Diagrams
Requirement Overview				
Requirement Name: RQ00004				
Revision: 1				
Review Name: After an interruption, movie resumes				
Description: After an interruption, movie resumes.				
Notes: In the case of an interruption (paused movie), after an interruption, the passenger can resume watching with automatically movie where it was interrupted.				
Requirement Details				
Requirement ID: RQ00004	Requirement Name: After an interruption, movie resumes	Requirement Description: After an interruption, movie resumes.	Requirement Notes: In the case of an interruption (paused movie), after an interruption, the passenger can resume watching with automatically movie where it was interrupted.	Requirement Status: Active



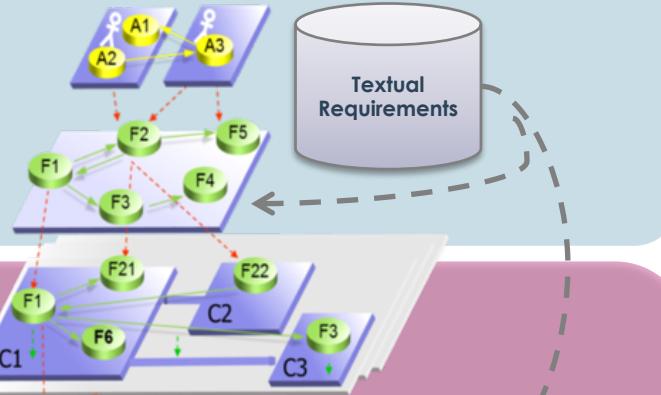
Certain model elements can
be considered as requirements



3. Contracts between engineering levels: workflow

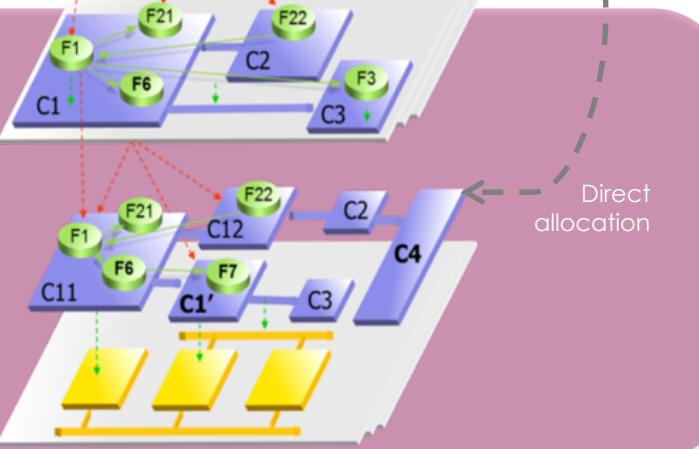
Level N

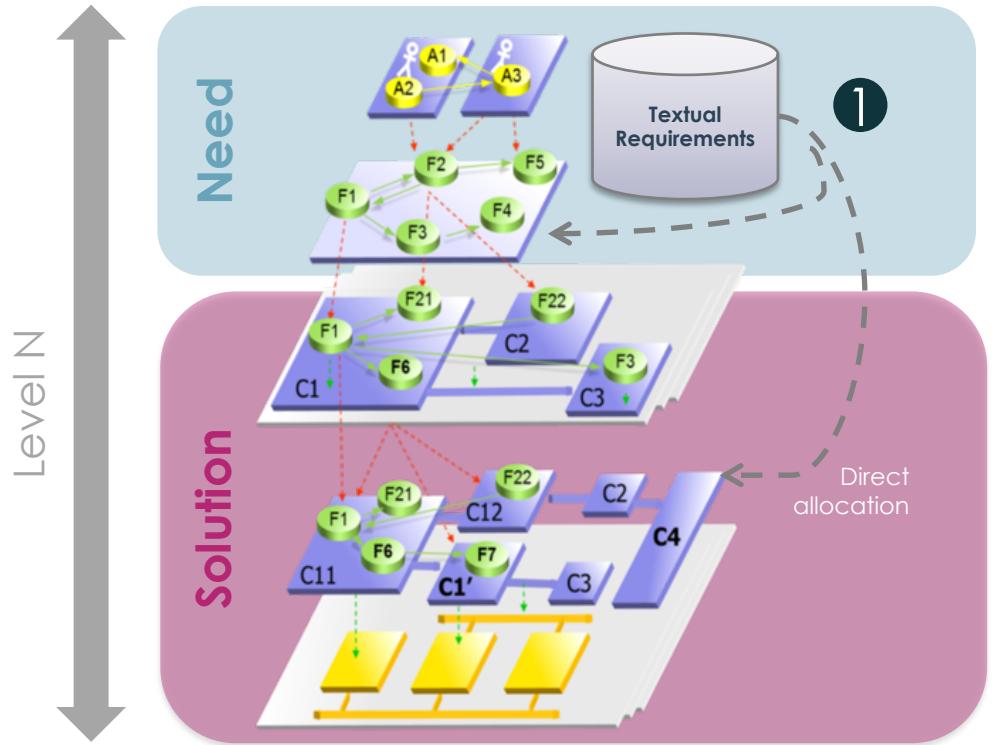
Need



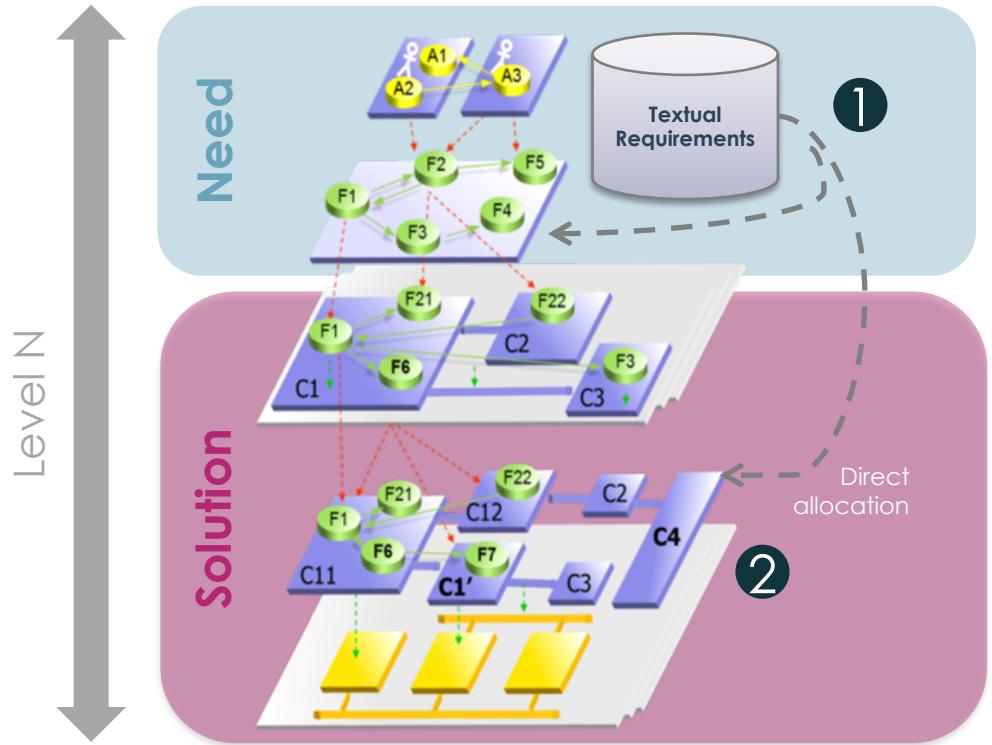
Direct allocation

Solution





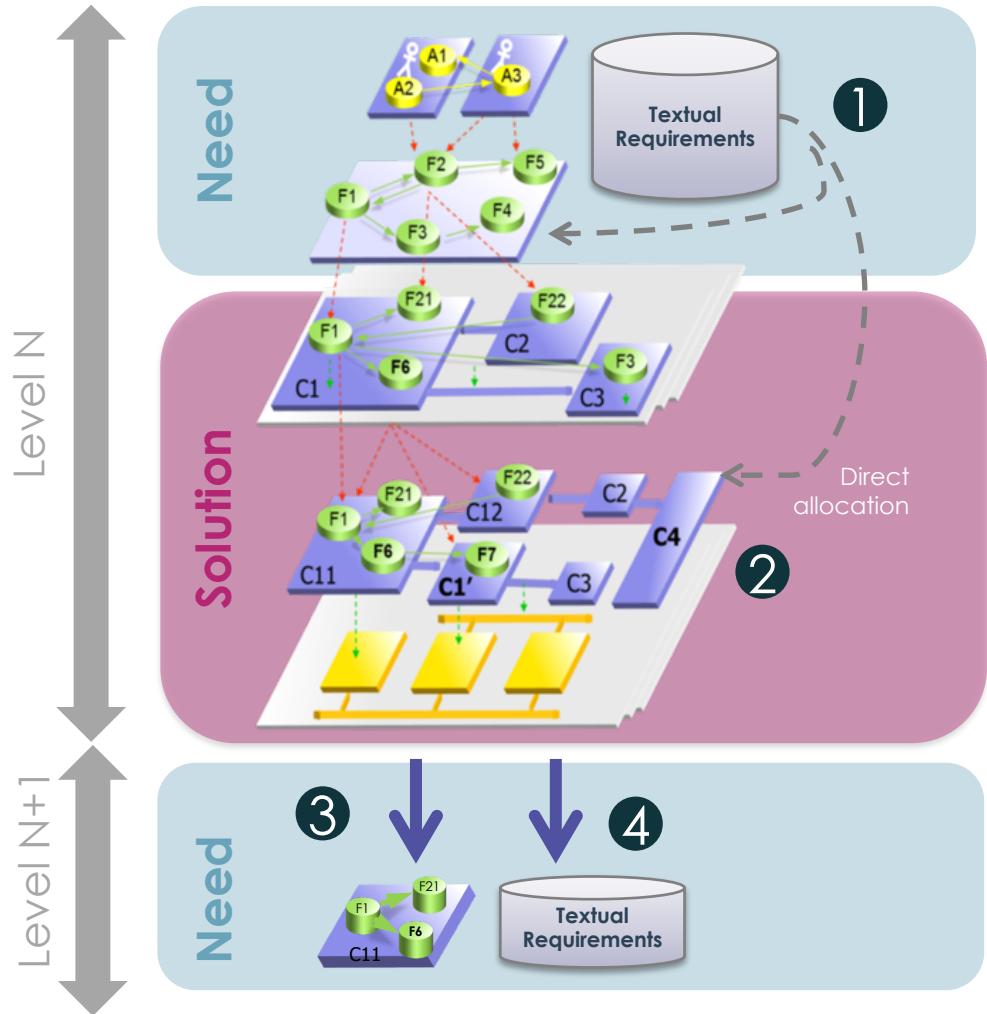
1. Elicitation of model and textual requirements on the system



2.

Architecture description specifies with the adequate level of detail how the system works and what is expected from each constituent

The goal here is to prepare the contracts for all subsystems and guarantee their proper integration.



3.

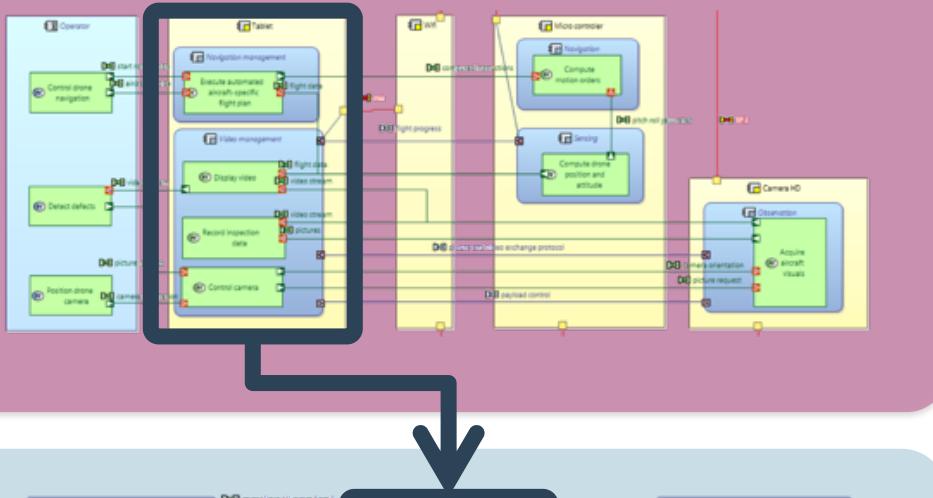
The context of a given system constituent is entirely computed (anything contributing to the definition of this constituent including allocated Functions, interfacing Components, etc.)

4.

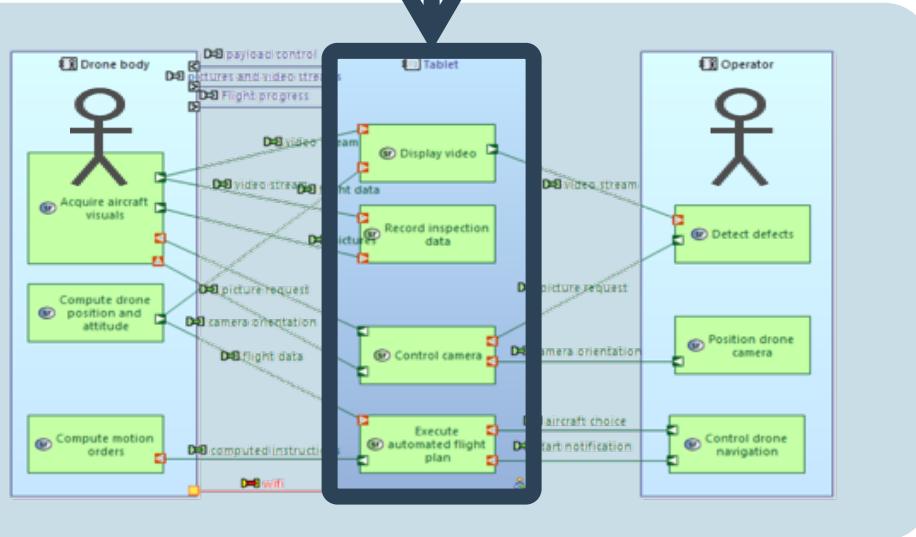
Textual requirements are created when needed, in addition to the model requirements: legal, non-functional, additional specification of internal expected behaviour

3

Tablet is a constituent of a drone-based system



Tablet is the (sub)system of interest





Model-based workflow favors co-engineering over the traditional differentiation between “customer” requirements and “system” requirements

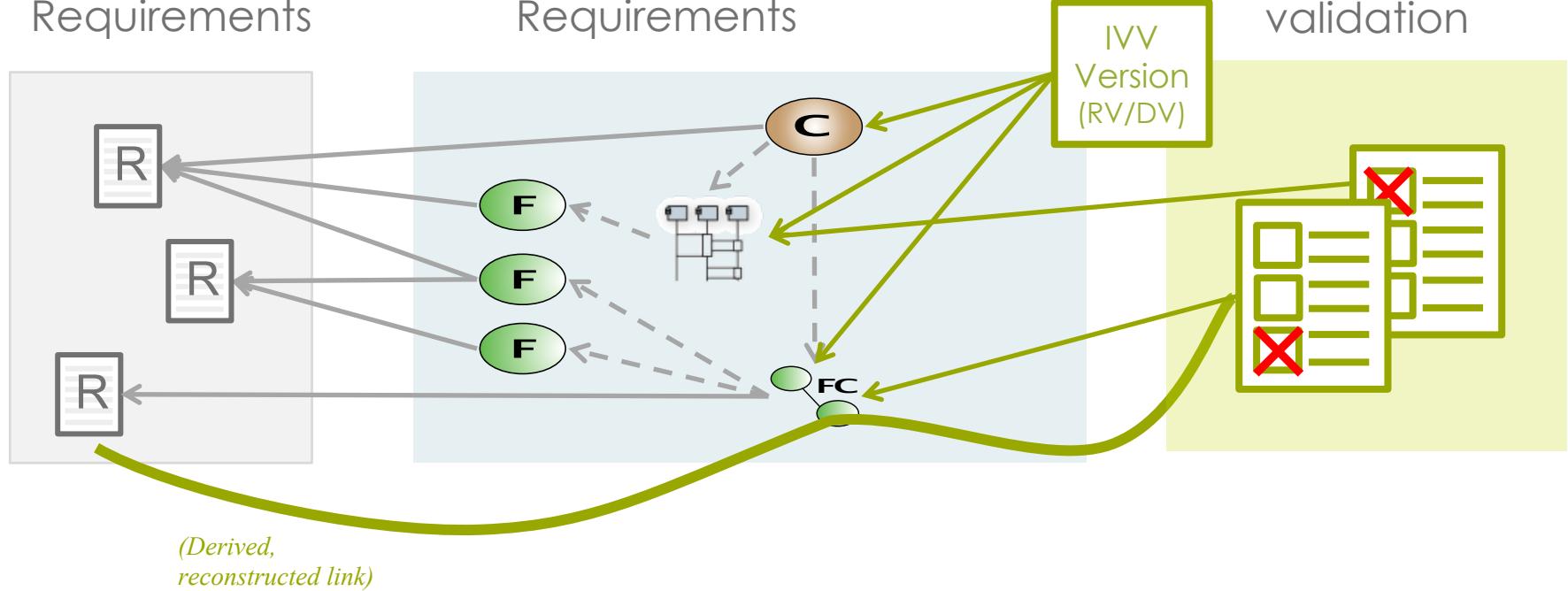
4. (Happy) consequences on V&V and incremental development strategy



Textual Requirements

Model Requirements

Verification and validation



C

Visualize data in live during flight



Display acquired HD video in live



Display multi-spectral image in live



Display thermal image in live



Visualize all collected mission data

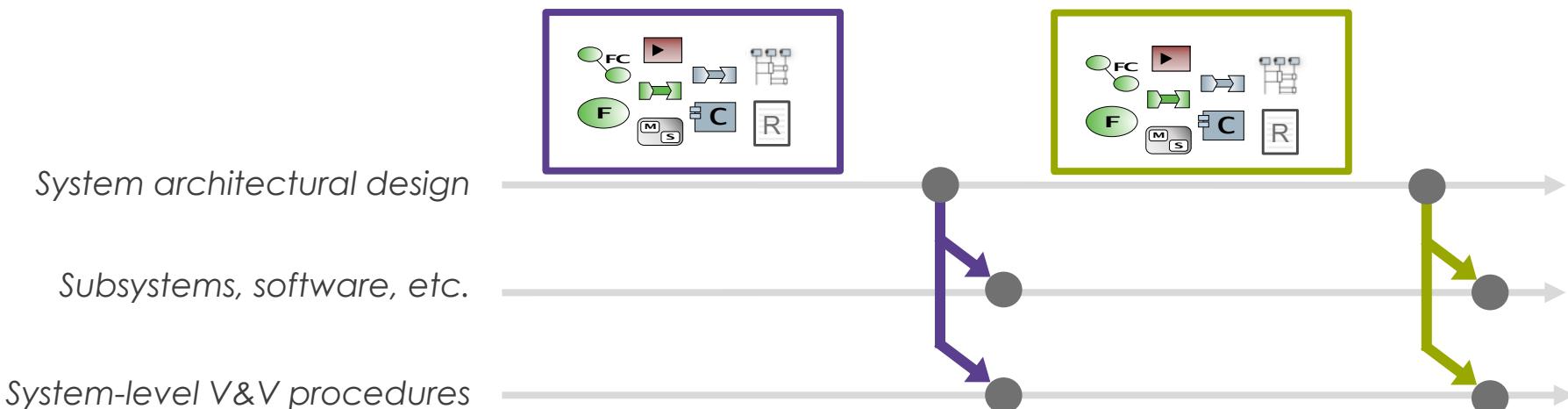


Visualize substance level in live



Definition of increments with expected functional chains (user stories)

Vertical slices of architectural design across need and solution models

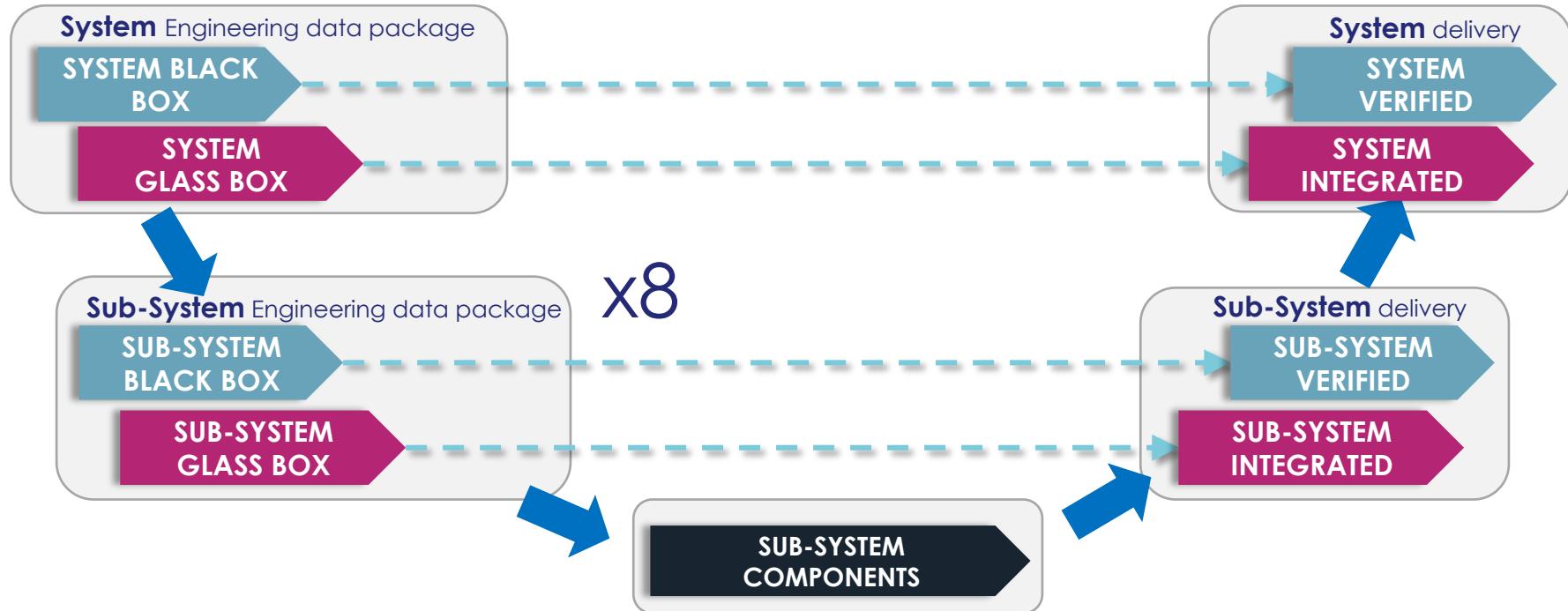




5. Instanciated workflow

Two years, 30 persons

Classic scheme, rolled out by increments



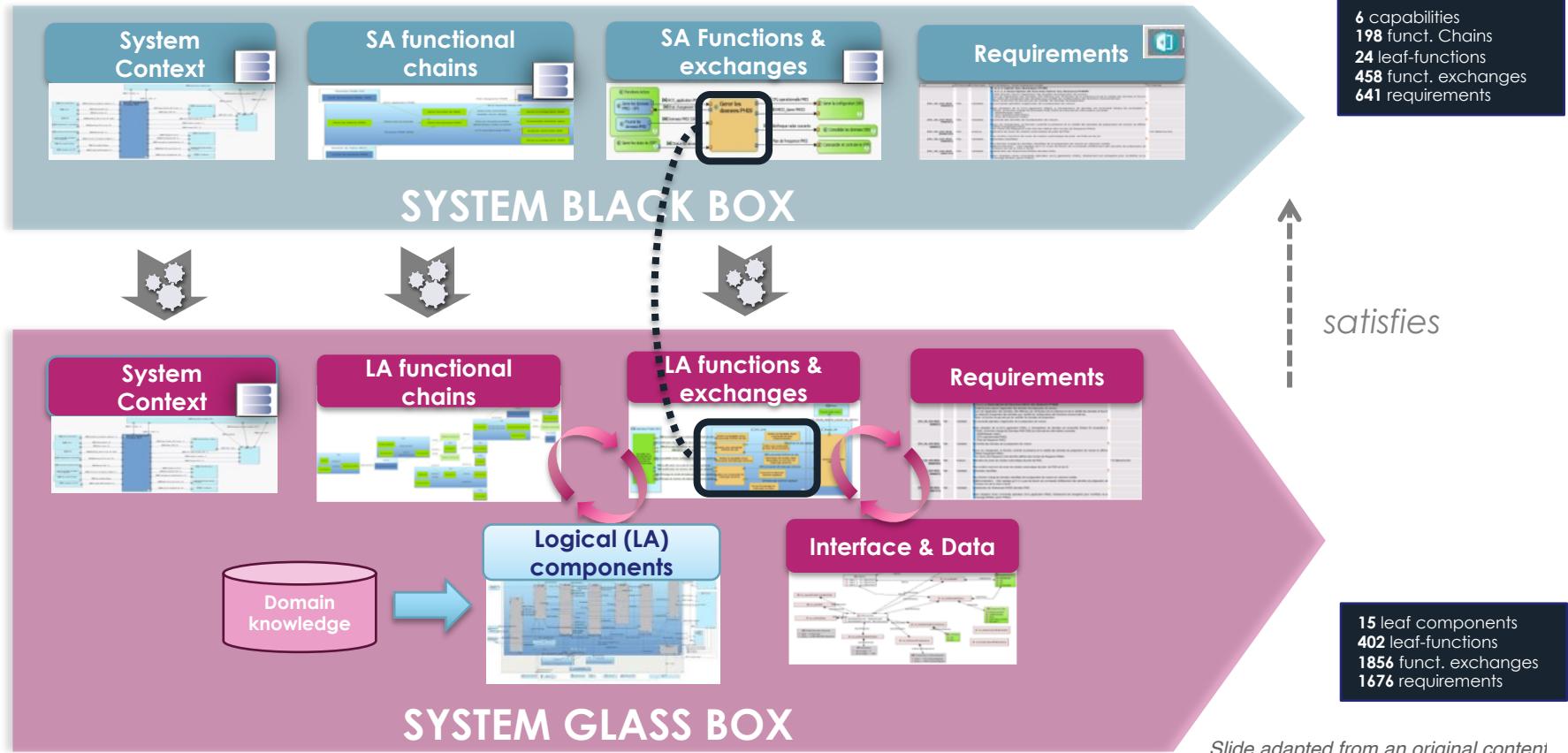
Slide adapted from an original content
created by the project team

Identical pattern at all levels

The diagram illustrates a top-down flow of requirements management:

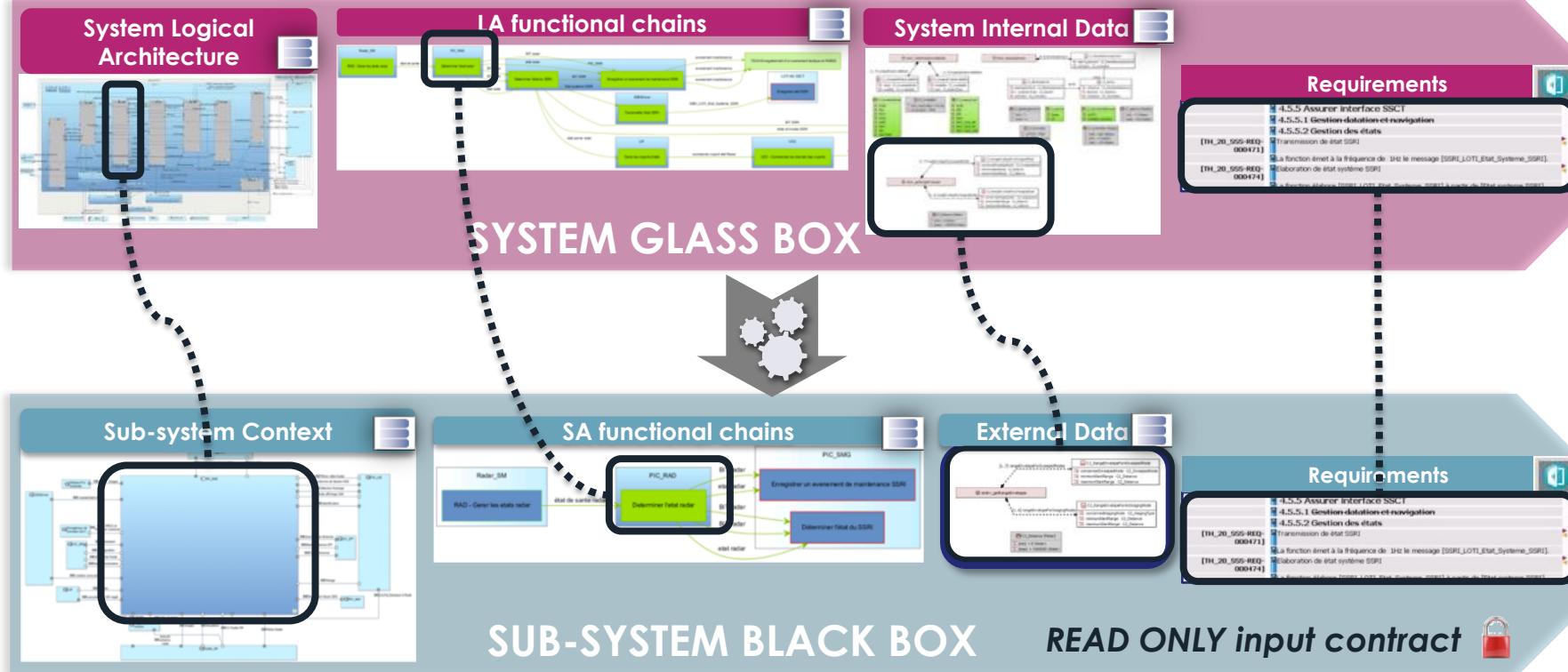
- Functional Chains list**: Shows a list of functional chains, such as "CF-SMG-020 - Déterminer et lancer l'état du système" and "CF-SMG-0201 - Alerter l'opérateur d'une panne de liaison SSR1-SSCT".
- Functional Chains release definition**: A table for "ATL2 - SSR1" defining components like SSR1_V0, SSR1_V1, SSR1_V2, SSR1_V3, SSR1_V4, SSR1_V5, and SSR1_V6.
- Functional chain**: A diagram showing a sequence of events: Radar (SSR1) → SSR1_V0 → SSR1_V1 → SSR1_V2 → SSR1_V3 → SSR1_V4 → SSR1_V5 → SSR1_V6 → Radar (SSR1). It includes a note: "Sous un certain état, l'émission peut être stoppée par le secteur de blanking"
- Requirements**: A table listing requirements like "4.5.1 Gestion datation et navigation" and "4.5.2 Gestion des états". Specific requirement **[TH_20_555-REQ-000471]** is highlighted.
- IVV procedure**: A table for "JHS_JCC_ARO_200" with steps 3 through 7. Step 7 is highlighted with a box.

Slide adapted from an original content
created by the project team



Slide adapted from an original content
created by the project team

Automated transitions



System verification: enhanced progress monitoring

Content removed

Test results: models facilitate analyses

Content removed



Tooling support

Managing increments with Capella

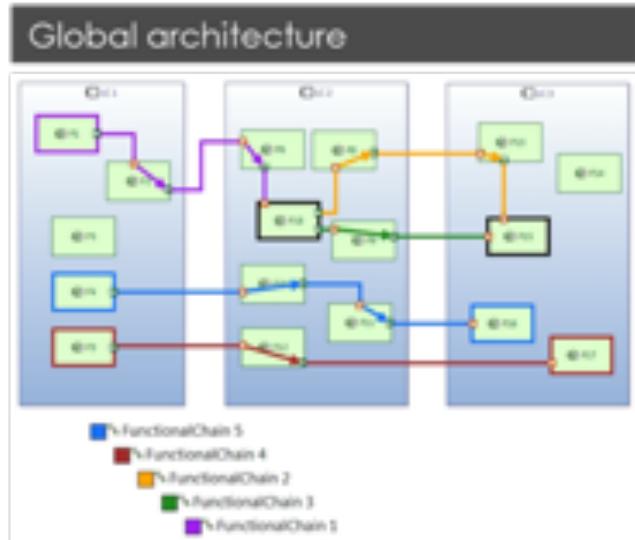
The screenshot shows a software application window with three main panels:

- Functional Chains list**: A tree view of functional chains, including:
 - CF-SMG-0201 - Alerter l'opérateur d'une panne de liaison SSRU-SSCT
 - CF-SMG-0202 - Alerter l'opérateur d'une panne équipement CVI
 - CF-SMG-0203 - Alerter l'opérateur d'une panne équipement IFF
 - CF-SMG-0204 - Alerter l'opérateur d'une panne équipement radar
 - CF-SMG-0205 - Réagir à une panne générale Radar
- Functional Chains release definition**: A table view of release definitions for functional chains, showing columns for Name, Description, Components, Functions, and Status.
- Functional chain**: A diagram illustrating the functional chain structure, showing components like "Alerter l'opérateur" and "IFF" connected by arrows.

Associate functional chains to increments

Compute, visualize and compare the footprints
in terms of expected components and functions

Focus on the dedicated tooling



RV 1 expected content

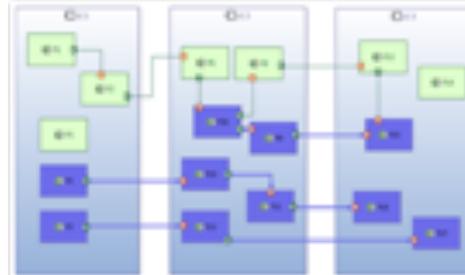
[Requested Version] RV 1

Elements

To add an element to this version, drag it from the palette.

Element	Correction
FunctionalChain 3	
FunctionalChain 4	
FunctionalChain 5	

REV 1 footprint



RV 2 expected content

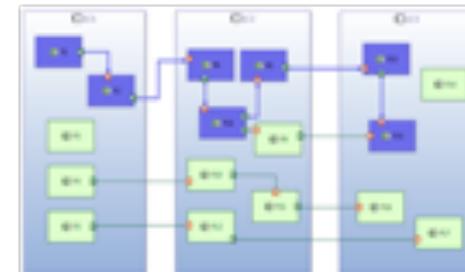
[Requested Version] RV 2

Elements

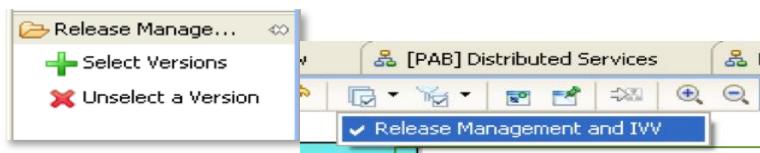
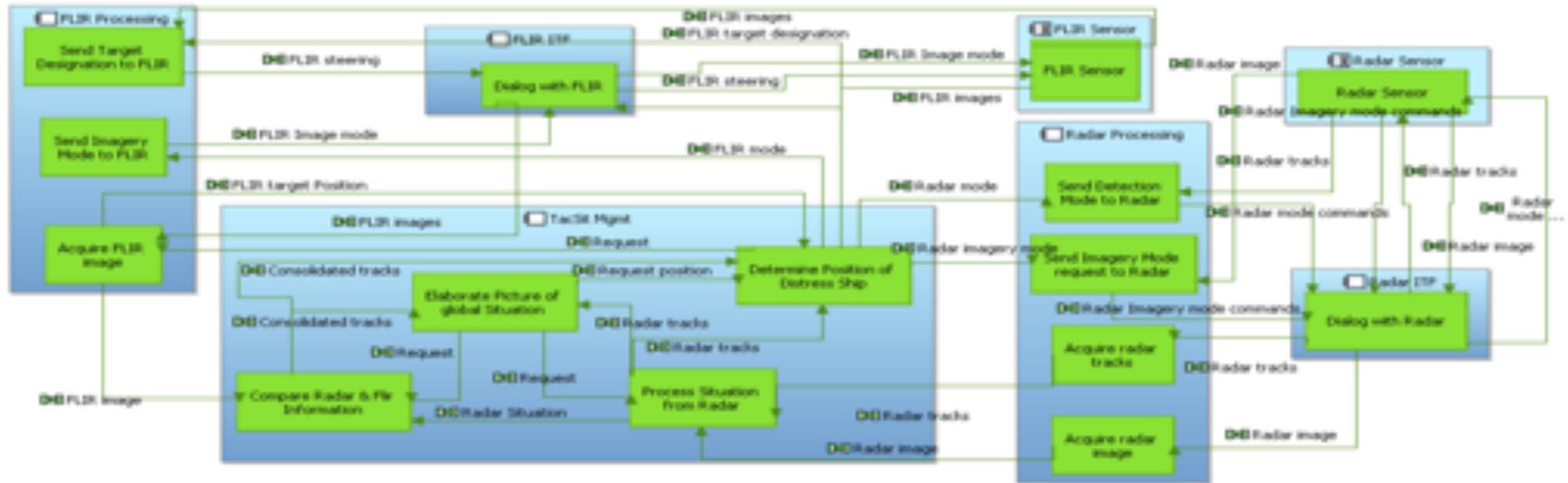
To add an element to this version, drag it from the palette.

Element	Correction
FunctionalChain 1	
FunctionalChain 2	

REV 2 footprint



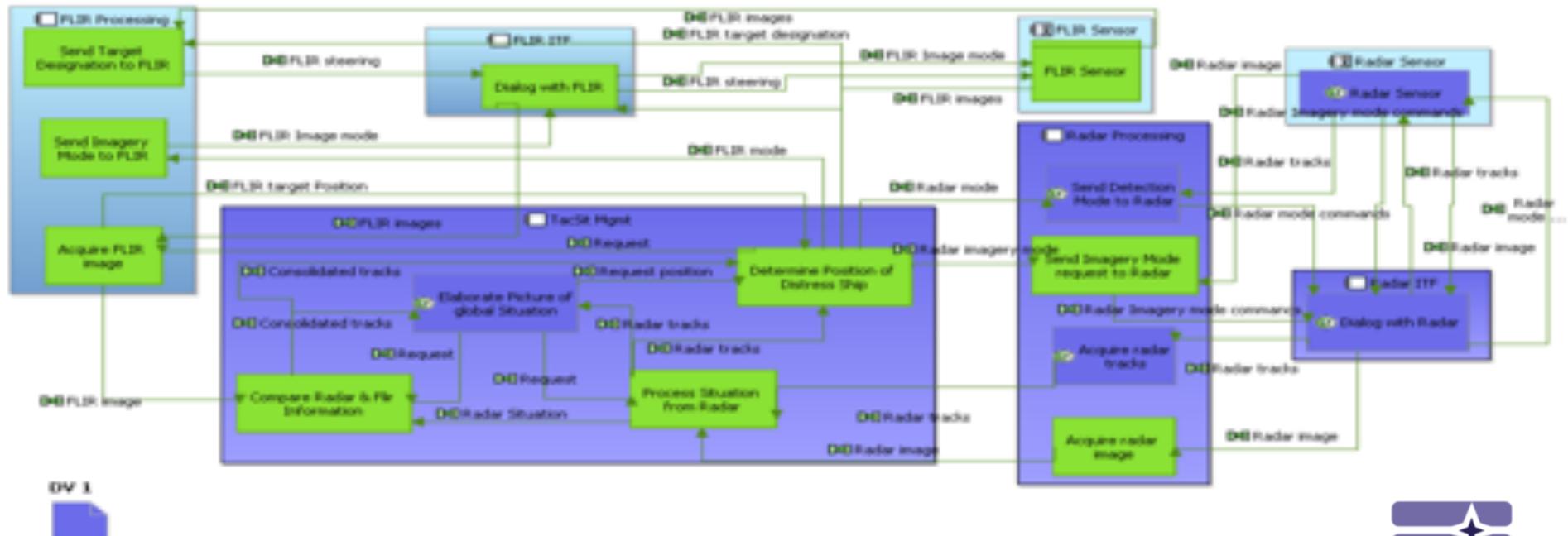
Preview of functional increments footprints



Release management viewpoint:
Automated visualization of versions



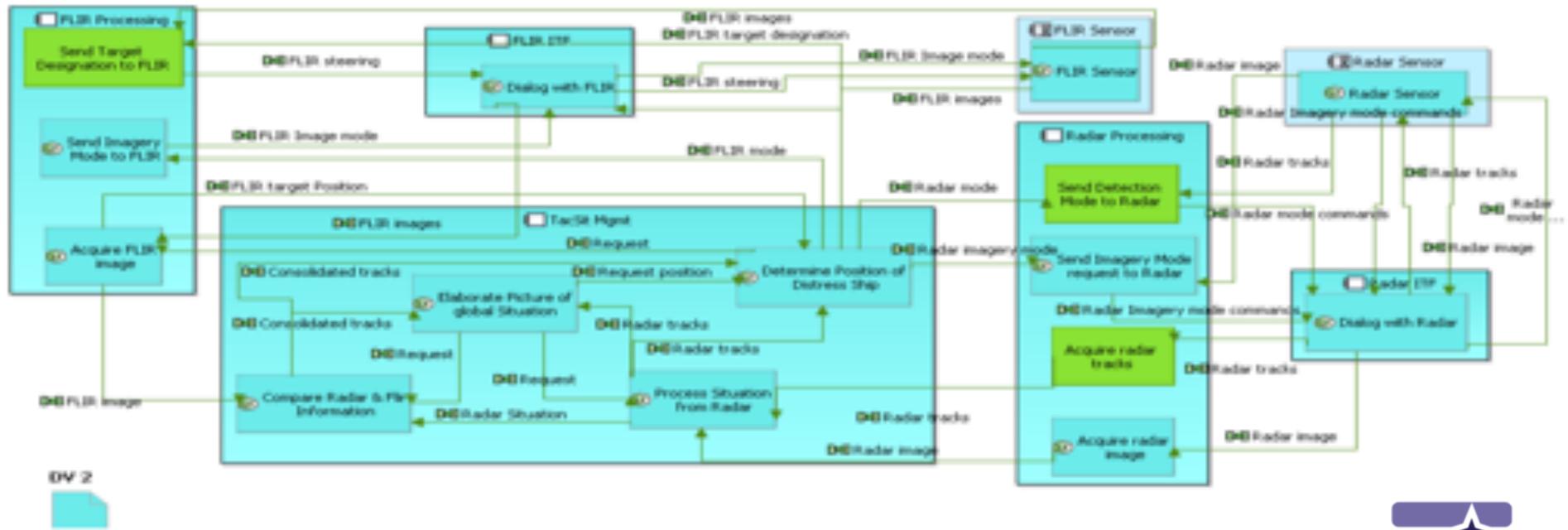
Preview of functional increments footprints



Developed Version 1
Available elements in BLUE

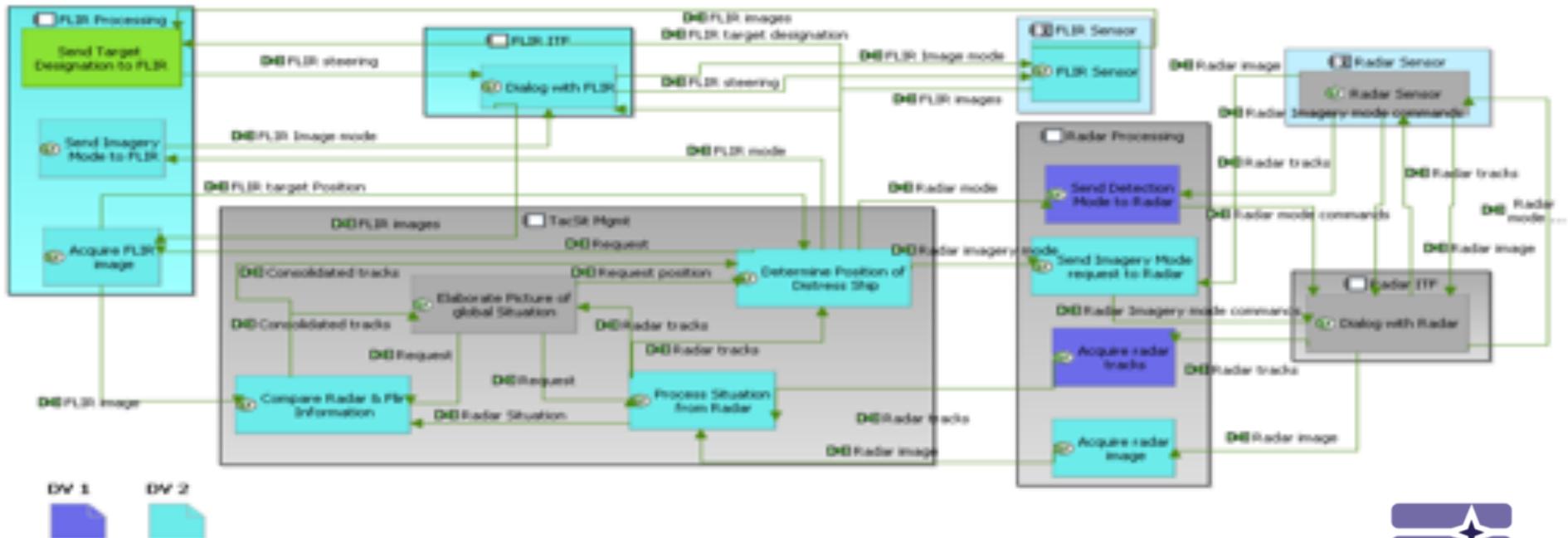


Preview of functional increments footprints



Developed Version 2
Available elements in CYAN

Preview of functional increments footprints



Developed Versions 1 & 2
Common available elements in GREY





Resources

Public forums, webinars and Youtube channel

Show: [Today's Messages](#) :: [Unread Messages](#) :: [Unanswered Messages](#) :: [Show Polls](#) :: [Message Navigator](#)
Admin: | [Reported Messages \(5\)](#) | [Custom Avatar Queue \(2\)](#) | [Group Manager](#)

Home > Capella

Forum	Messages	Topics	Last message
Capella - Capella Modeling Workbench and Arcadia MBSE Method - Click +/- to expand/collapse			
General Information Latest news about Capella	171	65	Mon, 29 October 2018 By: stephanie.lacrampe.obeo.fr
Arcadia The Arcadia model-based engineering method	214	50	Wed, 21 November 2018 By: xurenfei.glaway.com
Capella workbench Any tool-related question, suggestions, problem, etc	1625	363	Thu, 06 December 2018 By: zhangtonghui.glaway.com
Capella Studio Capella Studio related questions.	296	85	Mon, 03 December 2018 By: yvan.lissaad.obeo.fr

[[Mark all messages read](#)]

<https://polarsys.org/forums/index.php/f/10/>

THALES

How is Capella different?

WEBINAR, SEPTEMBER 12th 2017

Stéphane Bonnef
In charge of Thales Corporate MBSE Coaching & Community
Capella Design Authority

stéphane.bonnef@thalesgroup.com

www.thalesgroup.com

Webinar - How is Capella different?

Capella Webinars

Polaris Capella - 1/8

- 1 Capella [Webinar] - 51:52 PolarSys Capella
- 2 Capella [Webinar] - 49:47 [Webinar] Thales return on experience: usage of Capella in PolarSys Capella
- 3 3D Capella [Webinar] - 44:46 Equivalences and differences between Arcadia/Capella and PolarSys Capella
- 4 Capella [Webinar] - 47:24 What's new in Capella 1.2? PolarSys Capella

Aerospace Engineering Transformation Systems Engineering Transformation Naval Air Warfare Center Aircraft Division 4.9k views

<https://www.youtube.com/playlist?list=PLfrEYVpSGVLxEFR0DSWUTP8N5i3NTG4o->

Capella website



ARCADIA METHOD+

WORKBENCH+

SERVICES

COMMUNITY

CONTACT

DOWNLOAD

OPEN SOURCE SOLUTION FOR MODEL-BASED SYSTEMS ENGINEERING

Comprehensive, extensible and field-proven MBSE tool and method
to successfully design systems architecture

<https://polarsys.org/capella/index.html>



CapellaDay

MUNICH 2019

bit.ly/CapellaDay2019
Sept. 16th 2019



Thank You! Questions?

Capella website:

<http://www.polarsys.org/capella/>

LinkedIn 

<https://www.linkedin.com/groups/8605600>

Twitter 

https://twitter.com/capella_arcadia

Arcadia forum:

<https://polarsys.org/forums/index.php/f/12/>

Capella forum:

<https://polarsys.org/forums/index.php/f/13/>

IFE model & doc.:

<http://www.polarsys.org/capella/start.html>

www.thalesgroup.com

