Systemic Intevenction

[2024][TE-265]

Prof. Dr. Christopher S. Cerqueira

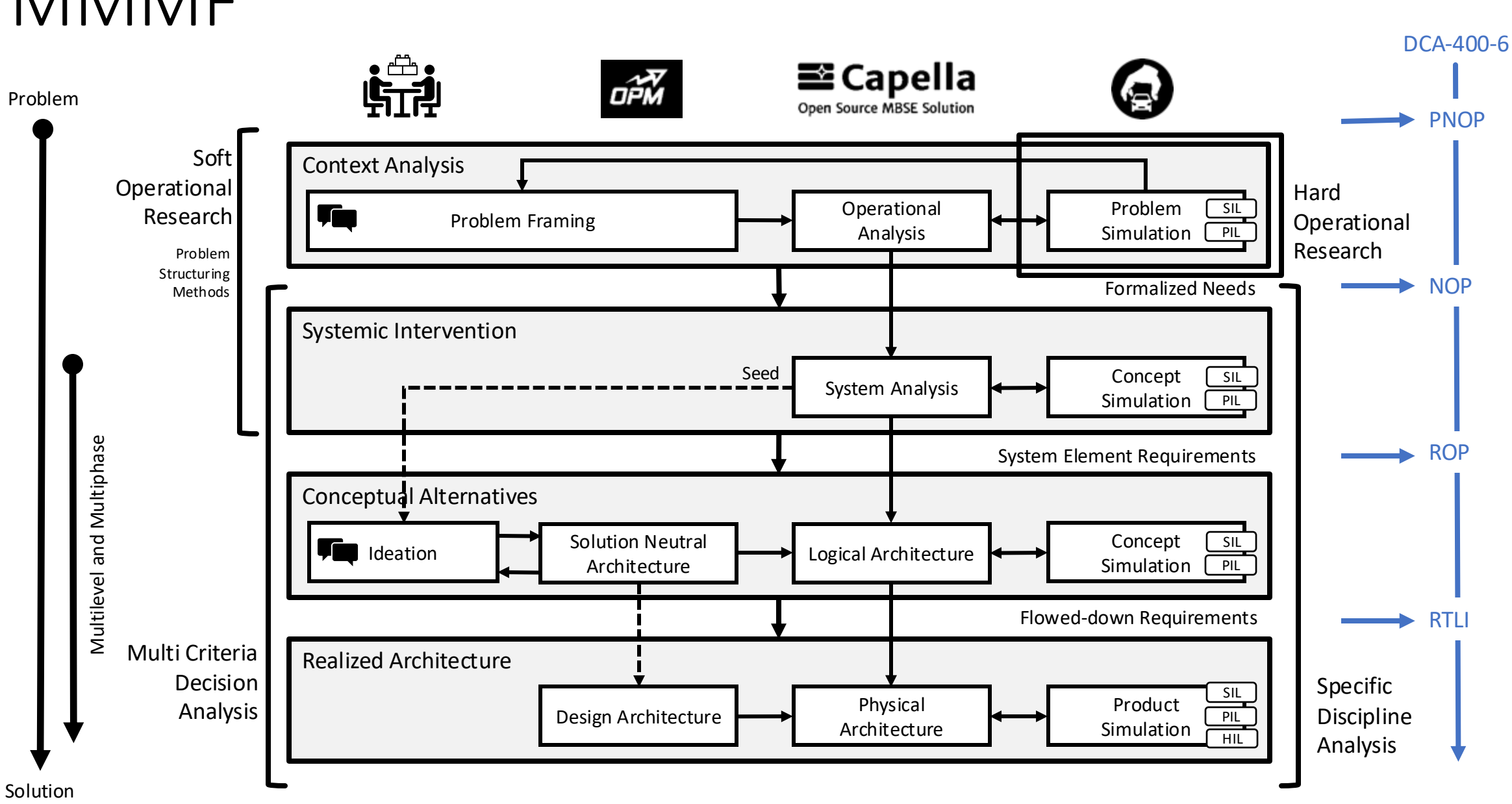
SEMANA		TEORIA	INDIVIDUAL	PESO	GRUPO	PESO
1	1	Estrutura e Filosofia do Curso	AI-01 - Resumo Cap 1 - HB INCOSE	10%		
05-Aug	1	O que é Engenharia de Sistemas? INCOSE				
	1	Elementos da Eng Sis.				
	1	Introdução aos diagrams clássicos.				
2		* (Viagem ao EUA)	AI-02 - Leitura/Resumo paper sobre representações clássicas.	10%		
12-Aug						
3		* (Viagem ao EUA)	AI-03 - Exercício sobre arquitetura e escrita de requisitos.	10%		
19-Aug						
4	1	Metodologias de MBSE e uso de modelos.	AI-04 - Resumo Artigo de Metodologias	10%		
26-Aug	1	Revisão de UML-SysML.				
	1	OPM				
	1	Arcadia				
5	1	OPM	AI-05 - Lista de exercícios	10%		
02-Sep	1					
	1					
	1					
6	1	Blocos e Classes	AI-06 - Lista de Exercícios	20%		
09-Sep	1					
	1	Máquina de Estados				
	1					
7	1	Casos de Uso	AI-07 - Lista de Exercícios	20%		
16-Sep	1					
	1	Sequência				
	1					
8	1	Integração dos pontos de vistas em um	AI-08 - Resumo sobre Ciclo de Vida de Modelos	10%	AI-08 - Descrição e Contorno do Problema.	100%
23-Sep	1	Associação dos artefatos de SE com modelos				
	1	Análise Operacional				
	1					
				100%		**100%**
SEM						
30-Sep						

SEMANA		TEORIA	INDIVIDUAL	PESO	GRUPO	PESO
9	1	Apresentação das necessidades			AG-09 - Apresentação Necessidades	20%
07-Oct	1	Intervenção Sistêmica				
	1	Associação com Requisitos				
	1					
10	1	Apresentação da Arq e Req de sistema	AI-10 - Exercícios de Arquitetura Funcional	20%	AG-10- Apresentação Arq / Caixa Preta	20%
14-Oct	1	Conceitos de Arquitetura Funcional				
	1	Arquitetura Conceitual				
	1					
11	1	Utilização de modelos para outros processos			AG-11 - Geração de documentos	10%
21-Oct	1					
	1	Exportação automática de documentos				
	1					
12	1	Apresentação da arquitetura Conceitual	AI-12 - Explorar RCE lendo arquivo do Capella	20%	AG-12 - Apresentação Arq. Conceitual e Proposta de VV	20%
28-Oct	1	Co-Engineering / CDF / RCE				
	1	Arquitetura Concreta				
	1					
13		* (ADS-HLG)	AG.13 - Explorar Plugin M2DOC (extra)	20%		
04-Nov						
14		* (ADS-HLG)	AG-14 - Explorar Plug in P4C (extra)	20%		
11-Nov						
15	1	Metamodelo	AG=5 - Figura do Metamodelo	20%	AG-15 = Relatório de Proposta de plugin	20%
18-Nov	1	Capella Studio - Criação de plugins				
	1					
	1					
16	1	Apresentação final			AG-16 - Apresentação do Projeto Completo	20%
25-Nov	1					
	1					
	1	Encerramento do Curso				
				100%		110%

EXAME
02-Dec
13-Dec

Grupo: Apresentação / Relatório / Gravação / Código de um: plugin ou doc **100%**

MMMF


Problem
Solution
Multilevel and Multiphase
Soft Operational Research
Problem Structuring Methods
Multi Criteria Decision Analysis
Capella
Open Source MBSE Solution
Context Analysis
Problem Framing
Operational Analysis
Problem Simulation
SIL
PIL
Hard Operational Research
Formalized Needs
Systemic Intervention
Seed
System Analysis
Concept Simulation
System Element Requirements
Conceptual Alternatives
Ideation
Solution Neutral Architecture
Logical Architecture
Flowed-down Requirements
Realized Architecture
Design Architecture
Physical Architecture
Product Simulation
HIL
Specific Discipline Analysis
DCA-400-6
PNOP
NOP
ROP
RTLI

Emergence

https://www.sebokwiki.org/wiki/Emergence

- Principle of Emergence:

As the entities of a system are brought together, their interaction will cause function, behavior, performance, and other intrinsic properties to emerge.

- Emergence is the **power and the magic of systems**. Emergence refers to **what appears, materializes, or surfaces when a system operates**. Obtaining the desired emergence is why we build systems. **Understanding emergence is the goal**—and the art—of system thinking.
- What emerges when a system comes together? **Most obviously and crucially, function emerges.** Function is what a system does: its actions, outcomes, or outputs. In a designed system, we design so that the anticipated desirable primary function emerges (cars transport people).

TABLE 2.1 | Types of emergent functions

	Anticipated Emergence	Unanticipated Emergence
Desirable	Cars transport people Cars keep people warm/cool Cars entertain people	Cars create a sense of personal freedom in people
Undesirable	Cars burn hydrocarbons	Cars can kill people

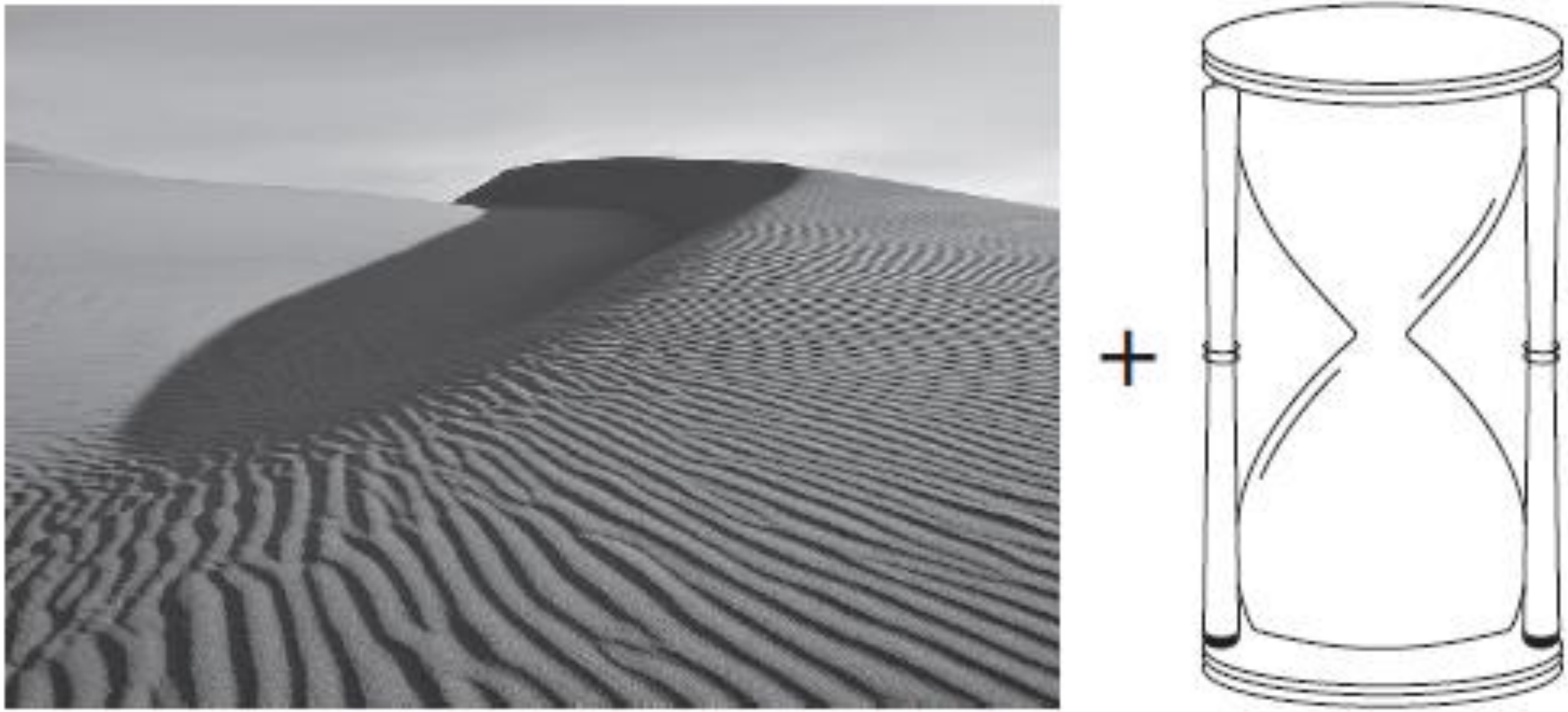


FIGURE 2.1 Emergent function from sand and a funnel: Time keeping. (Source: LOOK Die Bildagentur der Fotografen GmbH/Alamy)

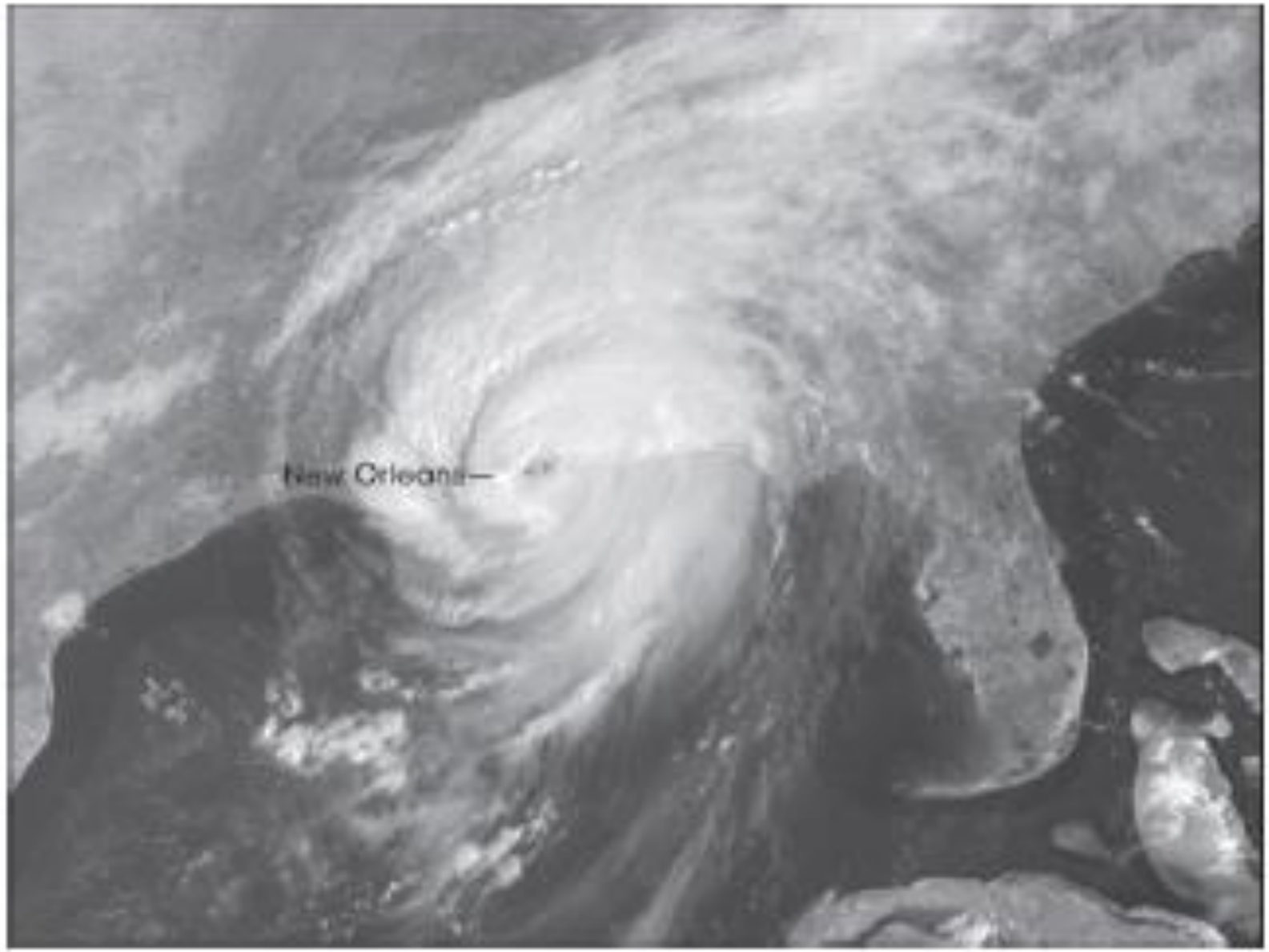


FIGURE 2.3 Emergency as emergence: Hurricane Katrina. (Source: Image courtesy GOES Project Science Office/NASA)

- These emergent properties associated with function, performance, the “ilities,” and the absence of emergencies are closely related to the value that is created by a system. Value is benefit at cost. We build systems to deliver the benefit (the worth, importance, or utility as judged by a subjective observer).

Table 1-1. Partial list of “-ilities”

accessibility	accountability	accuracy	adaptability	administrability
affordability	agility	auditability	autonomy	availability
compatibility	composability	configurability	correctness	credibility
customizability	debugability	degradability	determinability	demonstrability
dependability	deployability	discoverability	distributability	durability
effectiveness	efficiency	usability	extensibility	failure transparency
fault tolerance	fidelity	flexibility	inspectability	installability
integrity	interoperability	learnability	maintainability	manageability
mobility	modifiability	modularity	operability	orthogonality
portability	precision	predictability	process capabilities	producibility
provability	recoverability	relevance	reliability	repeatability
reproducibility	resilience	responsiveness	reusability	robustness
safety	scalability	seamlessness	self-sustainability	serviceability
securability	simplicity	stability	standards compliance	survivability
sustainability	tailorability	testability	timeliness	traceability

https://medium.com/continuousdelivery/evolvability-124ee5a8dd07

Requirements in Capella

https://www.slideshare.net/Obeo_corp/capella-webinar-writing-perfect-textual-requirements

DEFINITIONS

requirement [ri-**kwahy*uh*** **r**-m*uh*** nt] SHOW IPA

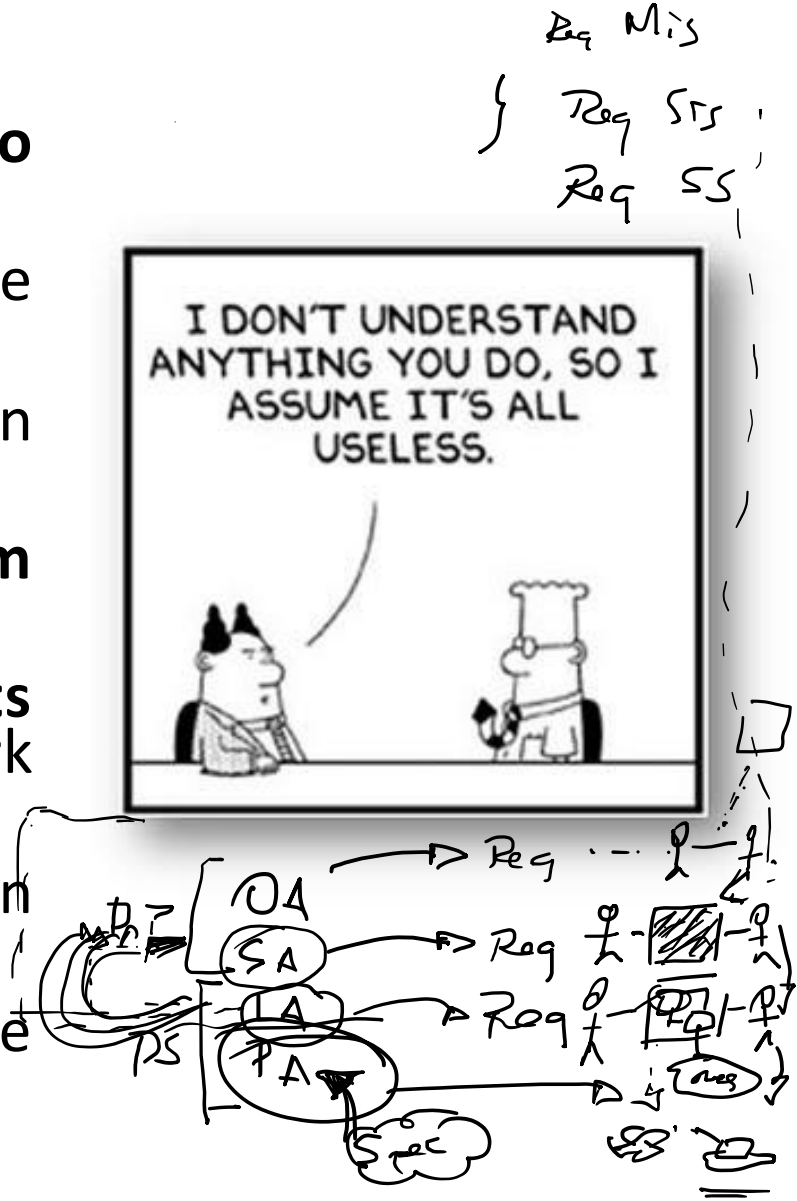
SEE SYNONYMS FOR *requirement* ON THESAURUS.COM

noun

1. that which is required; a thing demanded or obligatory:
   One of the requirements of the job is accuracy.
2. an act or instance of requiring.
3. a need or necessity:
   to meet the requirements of daily life.

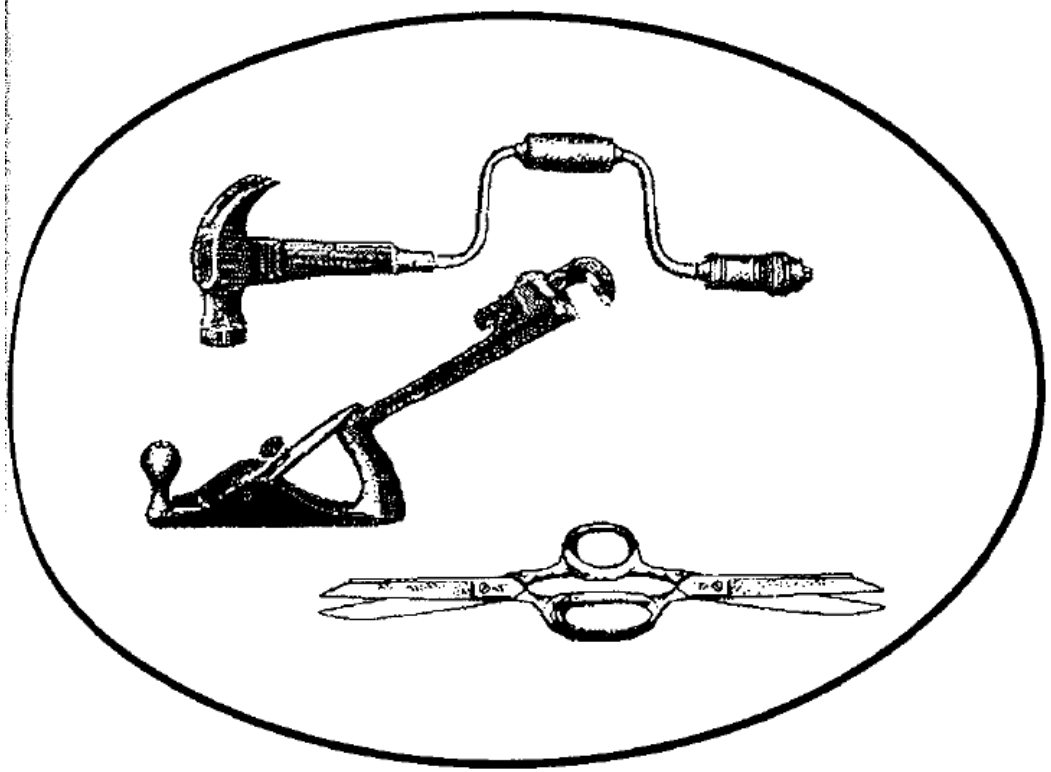
IMPORTANCE OF HAVING GOOD REQUIREMENTS

- Requirements tell you **what the system needs to do** (functional requirements).
- **How well** the system needs to do it (performance requirements)
- **What environment** the system has to work in (environmental requirements).
- What the system **must do to fit into the bigger system** (interface requirements).
- What **lower level subsystems/assemblies/components must do to fit** into the system and make it all work (allocation of requirements/resources).
- What you need to **do before you fly** (verification activities).
- And basically, **when you are done** (requirements are met).



TO DO OR NOT TO DO

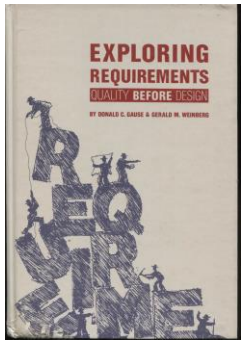
- **Functional Requirements** describe what the system should do and **Non-functional Requirements** place constraints on how these functional requirements are implemented.



How bad, poor and conflicting requirements create hassles:

DEFINING THE CORRECT HEURISTICS TO UNDERSTAND A REQUIREMENT

Mary had a little lamb.
Its fleece was white as snow.
And everywhere that Mary went,
The lamb was sure to go.



Mary *had* a little lamb.
(She no longer has the lamb.)

Mary had *a* little lamb.
(She had only one lamb, not several.)

Mary had a *little* lamb.
(It really was surprisingly small.)

Mary had a little *lamb.*
(She didn't have a dog, cat, cow, goat, or parakeet.)

Mary had a little lamb.
(John still has his little lamb.)

s a tour de force, we offer all five words emphasized:

Mary had a little lamb.
(As contrasted with Pallas, who still has four large turtles.)

Textual requirements and model requirements

Models add rigor to need expression / solution description

Models enable automated processing

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

Textual requirements and model requirements

Text is normally better for the first interactions with customers and suppliers

Legally binding documents are normally written in text

High-level needs and other expectations (environmental, regulations, etc) are easier to express with textual descriptions

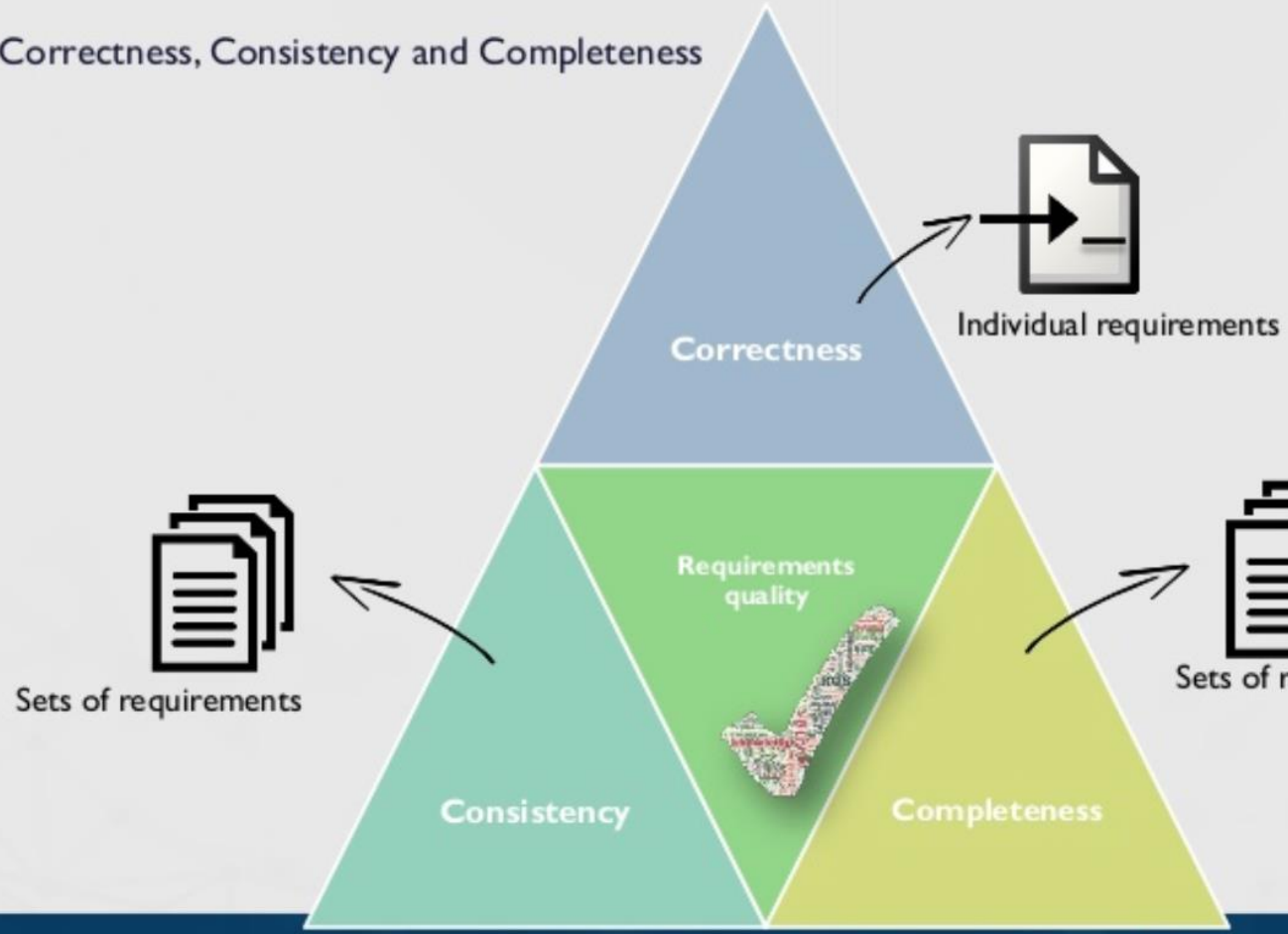
Some expectations on a given element at a given engineering level do not require any formal modeling (which is left to subsystem design)

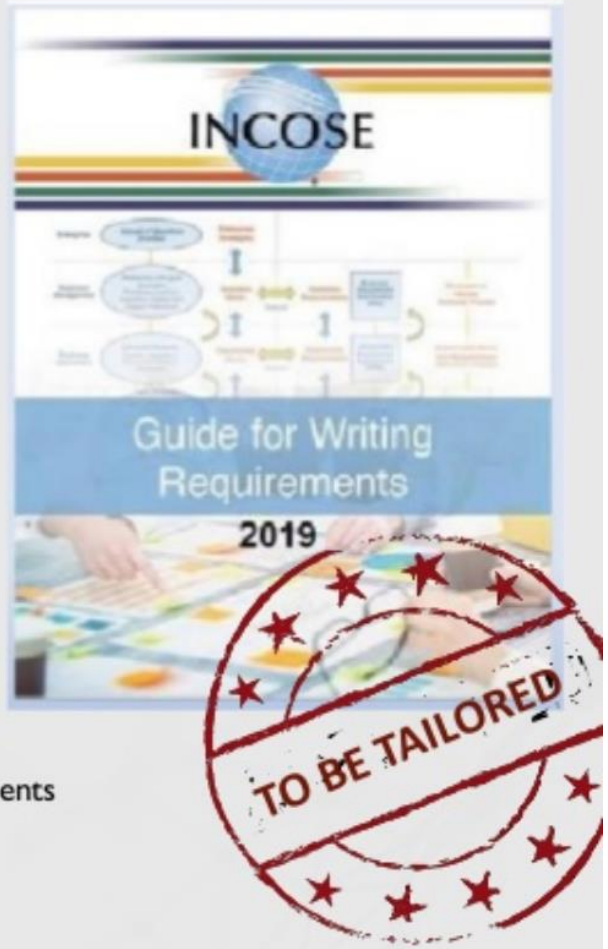
Text allows for a much earlier focus on quality (verification of textual requirements). Remember: "Quality is everyone's responsibility" by E. Deming

So, **textual form** of needs and requirements are not only useful, they are **fully necessary**

Real-time quality analysis: CCC Approach

- CCC – Correctness, Consistency and Completeness

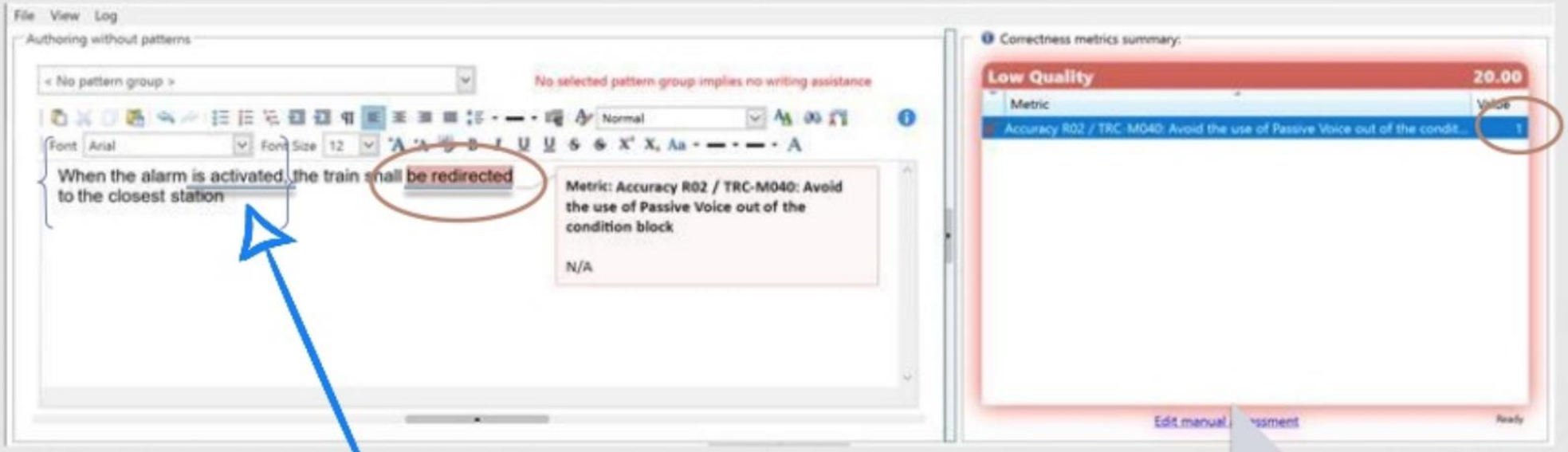




Real-time quality analysis: Patterns

Patterns to contextualize how correctness metrics are executed:

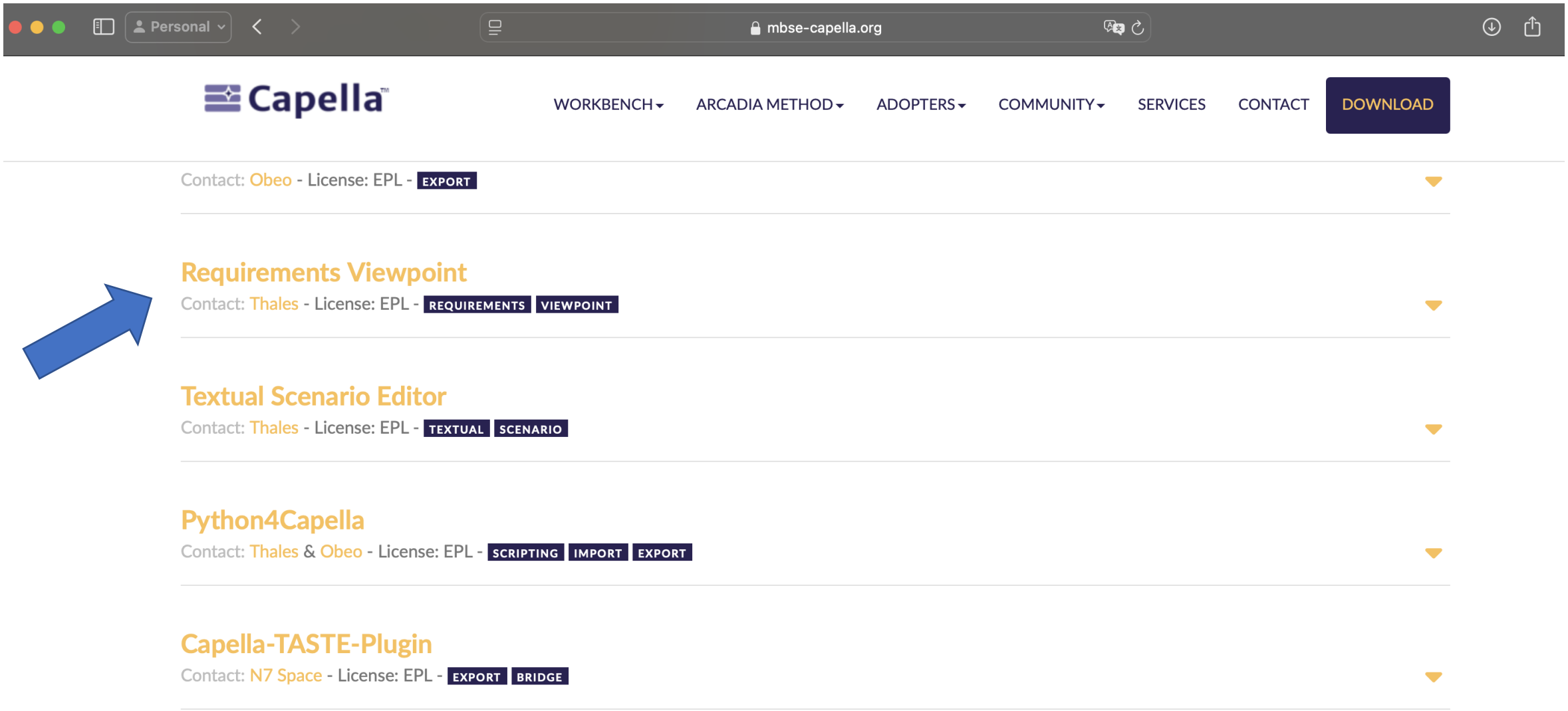
Example: Application of INCOSE R02 (Use Active Voice) to detect passive voice **only outside conditions:**



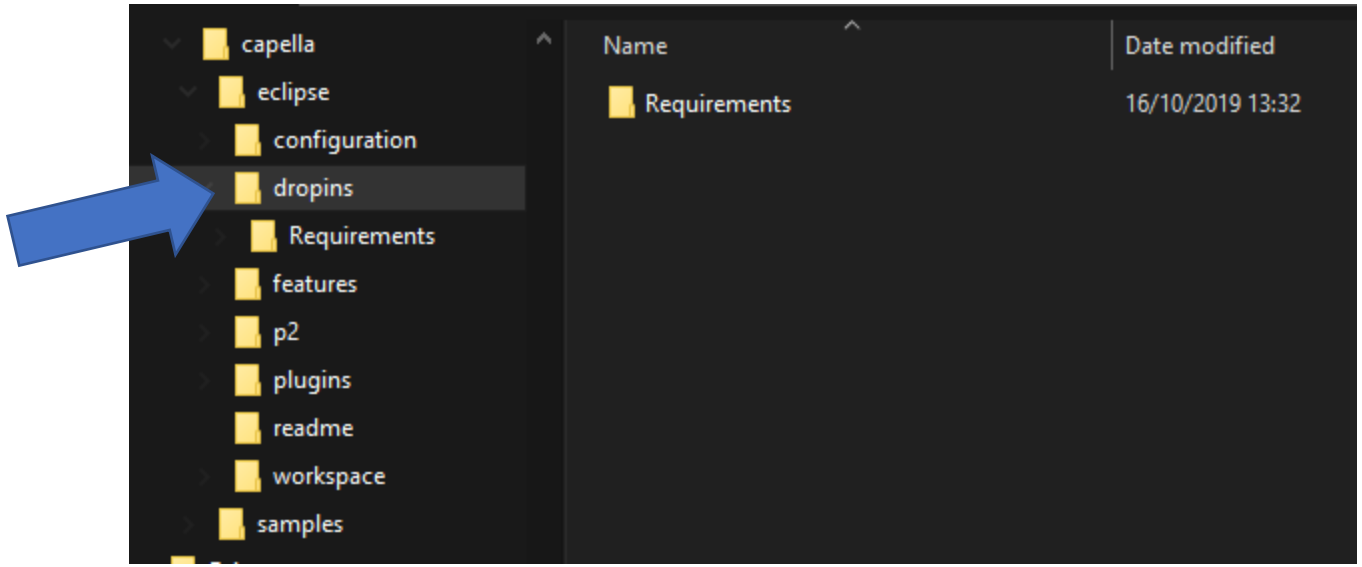
When / After / If ... | [Condition] | <Subject> | Shall | <Action> | <Object> | [Constraint]

REQUIREMENTS ADDON

[IF NOT INSTALLED] Add the req addon

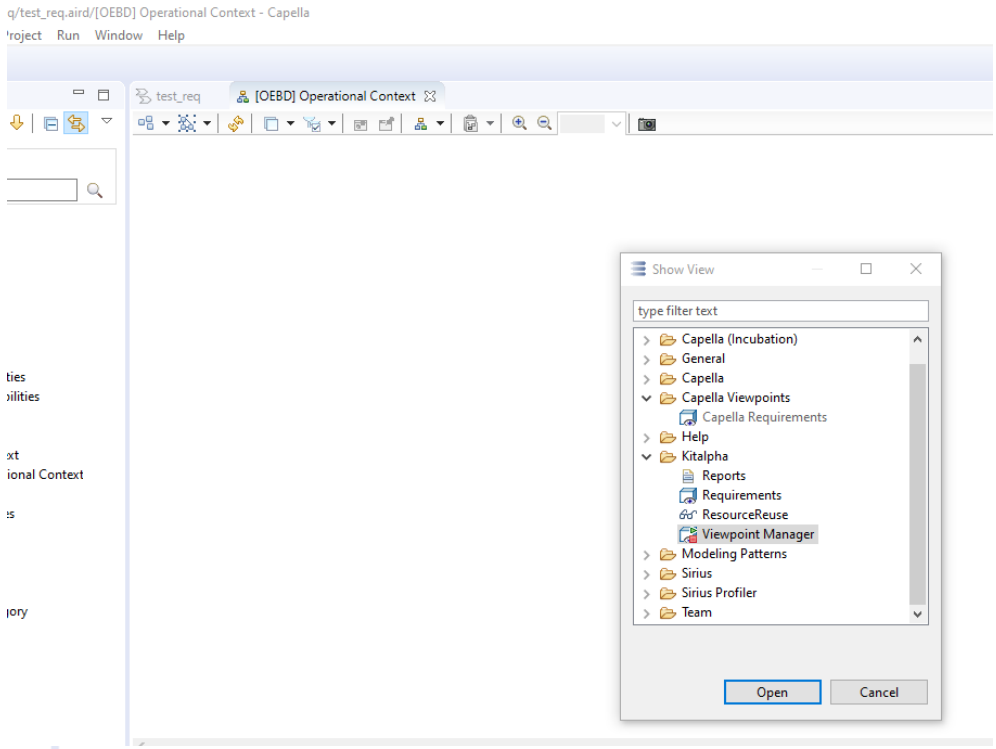


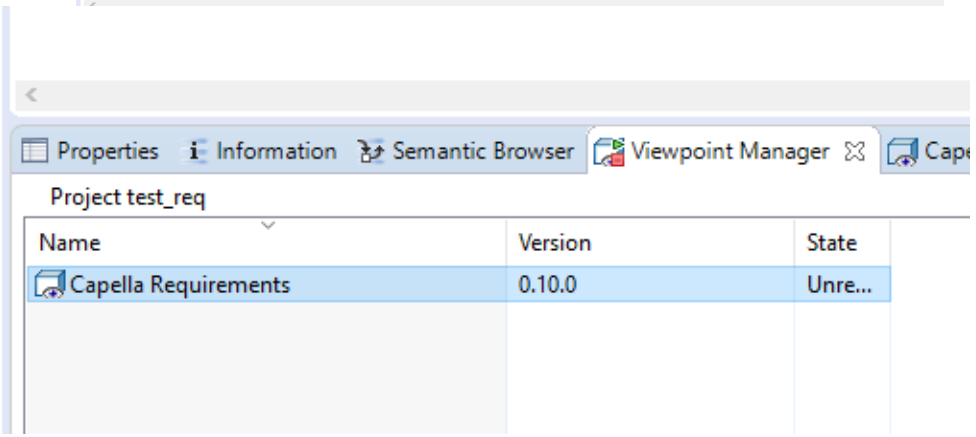
[IF NOT INSTALLED] Unzip in dropin folder



[IF NOT INSTALLED] Last Steps

- Start Capella
- Open the **Viewpoint Manager** view using **Window** menu then **Show View** and **Other…**
- Select **Viewpoint Manager** in **Kitalpha** directory and press **OK**
- The **Viewpoint Manager** view is displayed
- The viewpoints available in the platform are listed in this view.
- Select any model element (diagram element, element in the project explorer) related to your project
- Right-click on the name of a viewpoint and select **Reference** in order to start the viewpoint

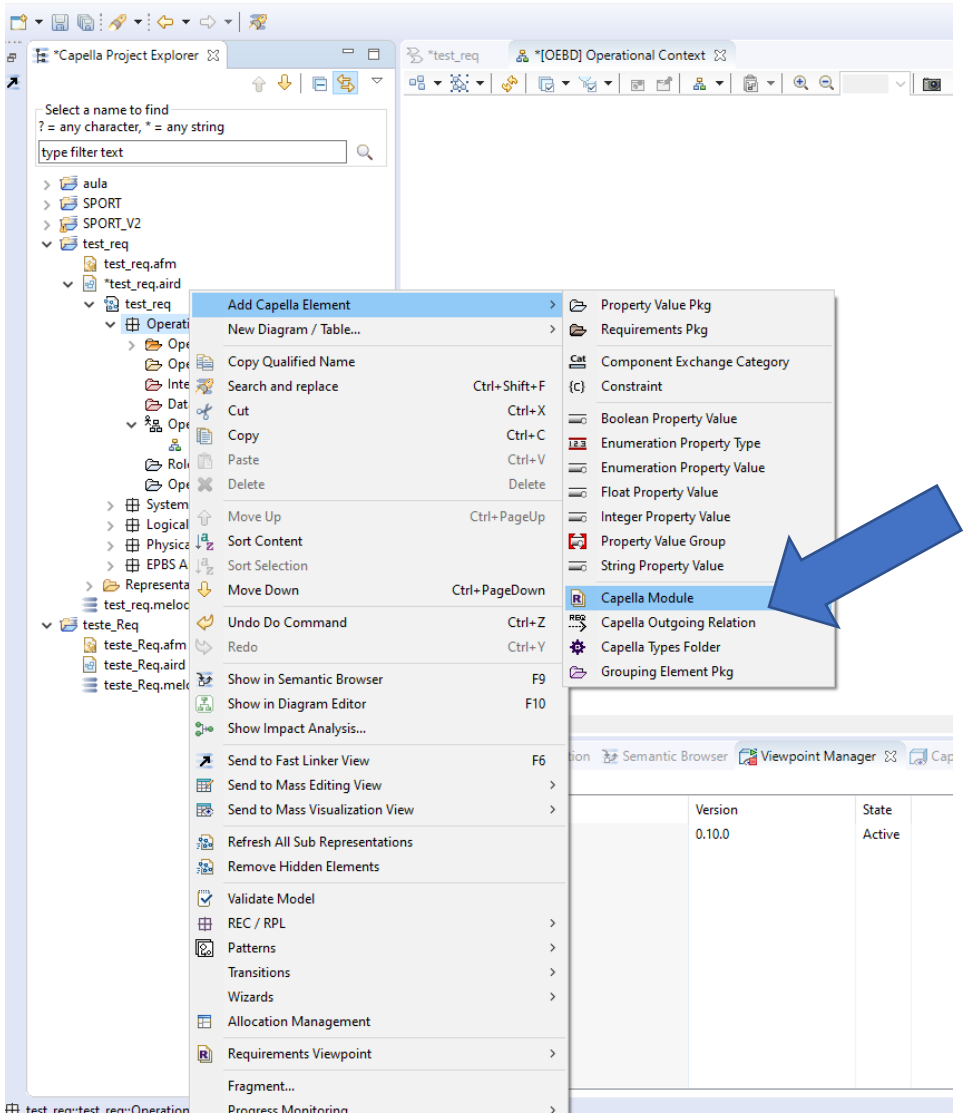


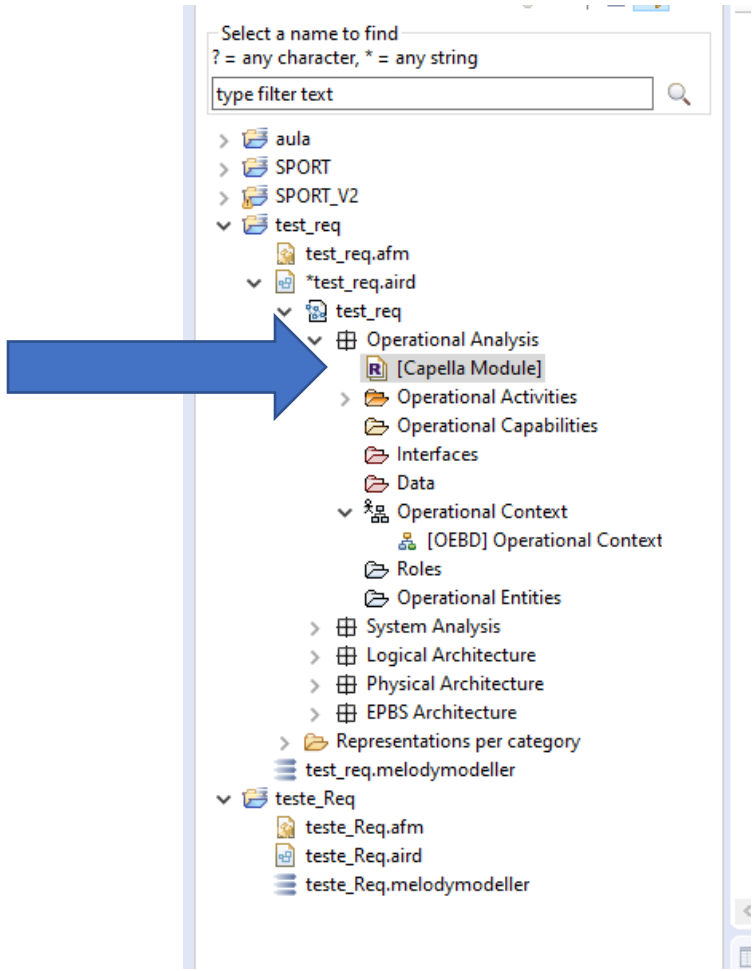


Can be used in all layers

- Operational Analysis Requirements
- System Analysis Requirements
- Logical Architecture Requirements
- Physical Architecture Requirements
- EPBS Architecture Requirements

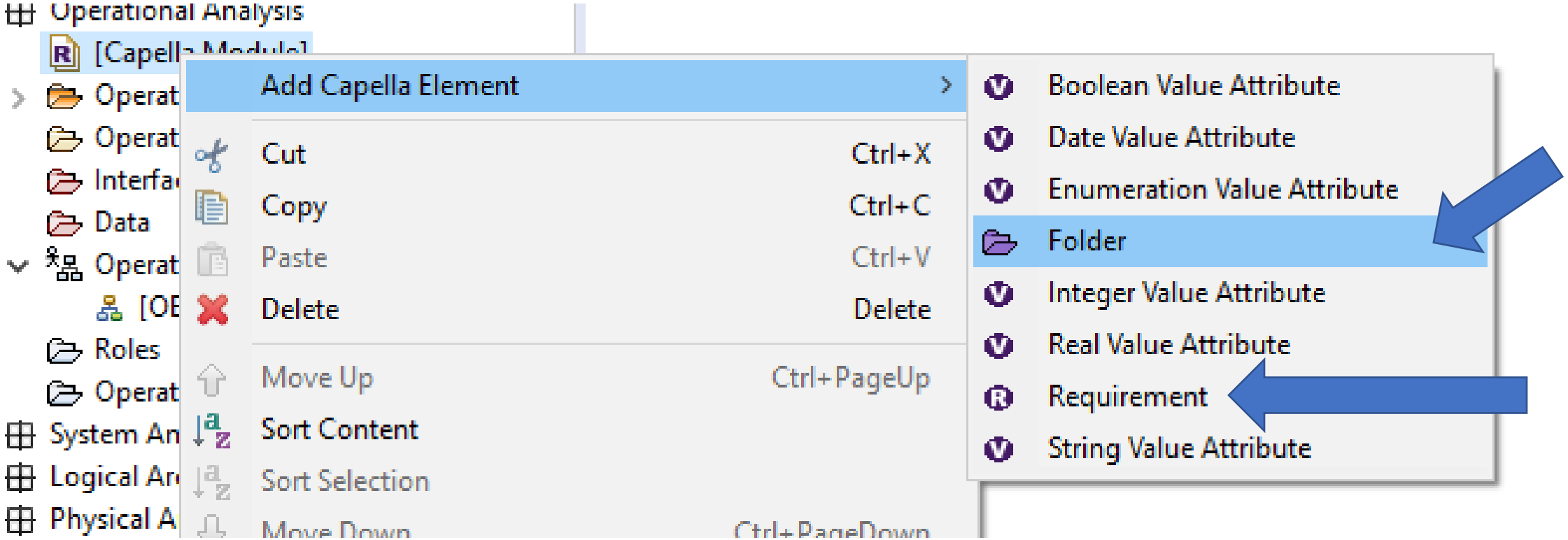
Add a capella module in the layer





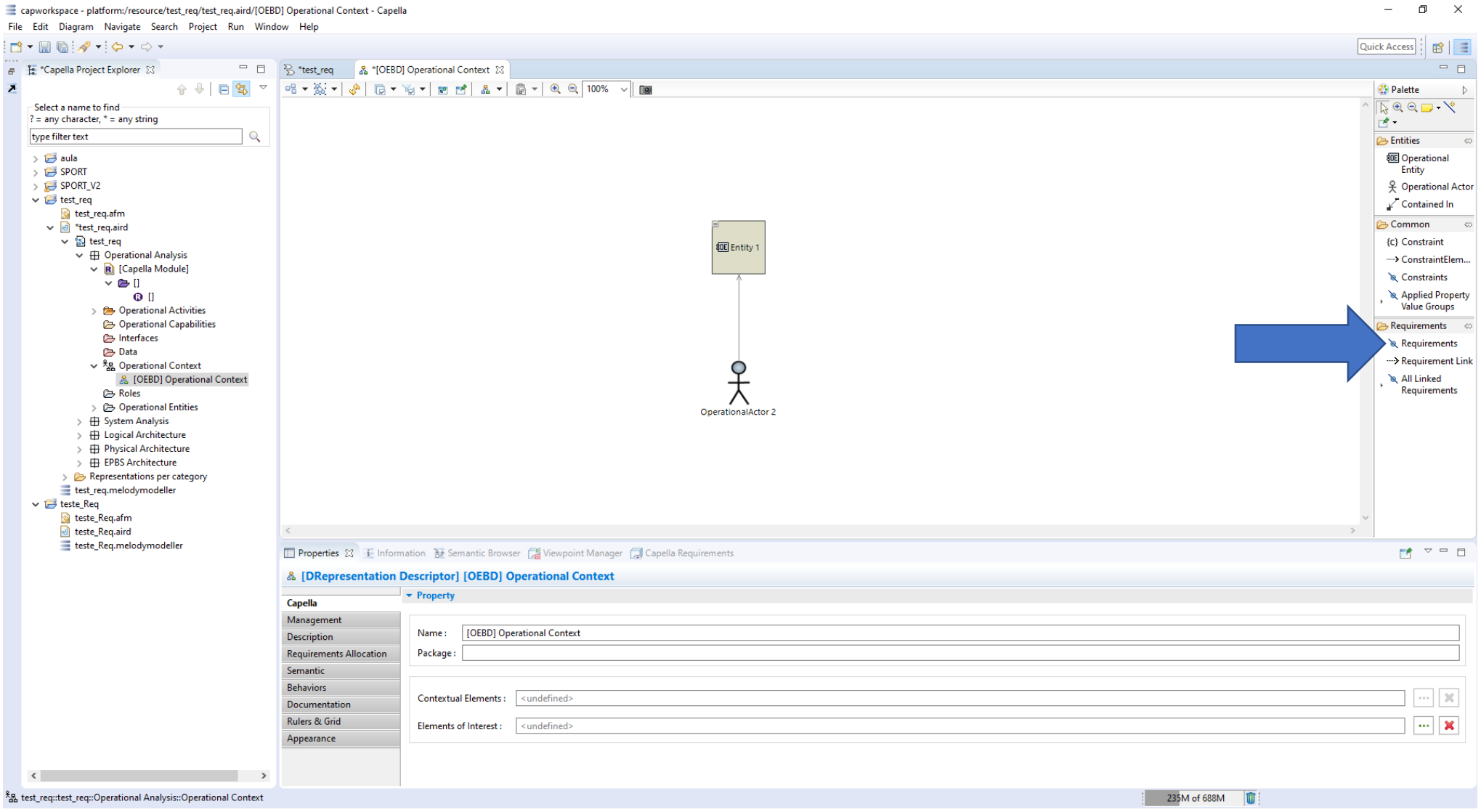
** if not seen check the filters

Create a requirement folder & requirement

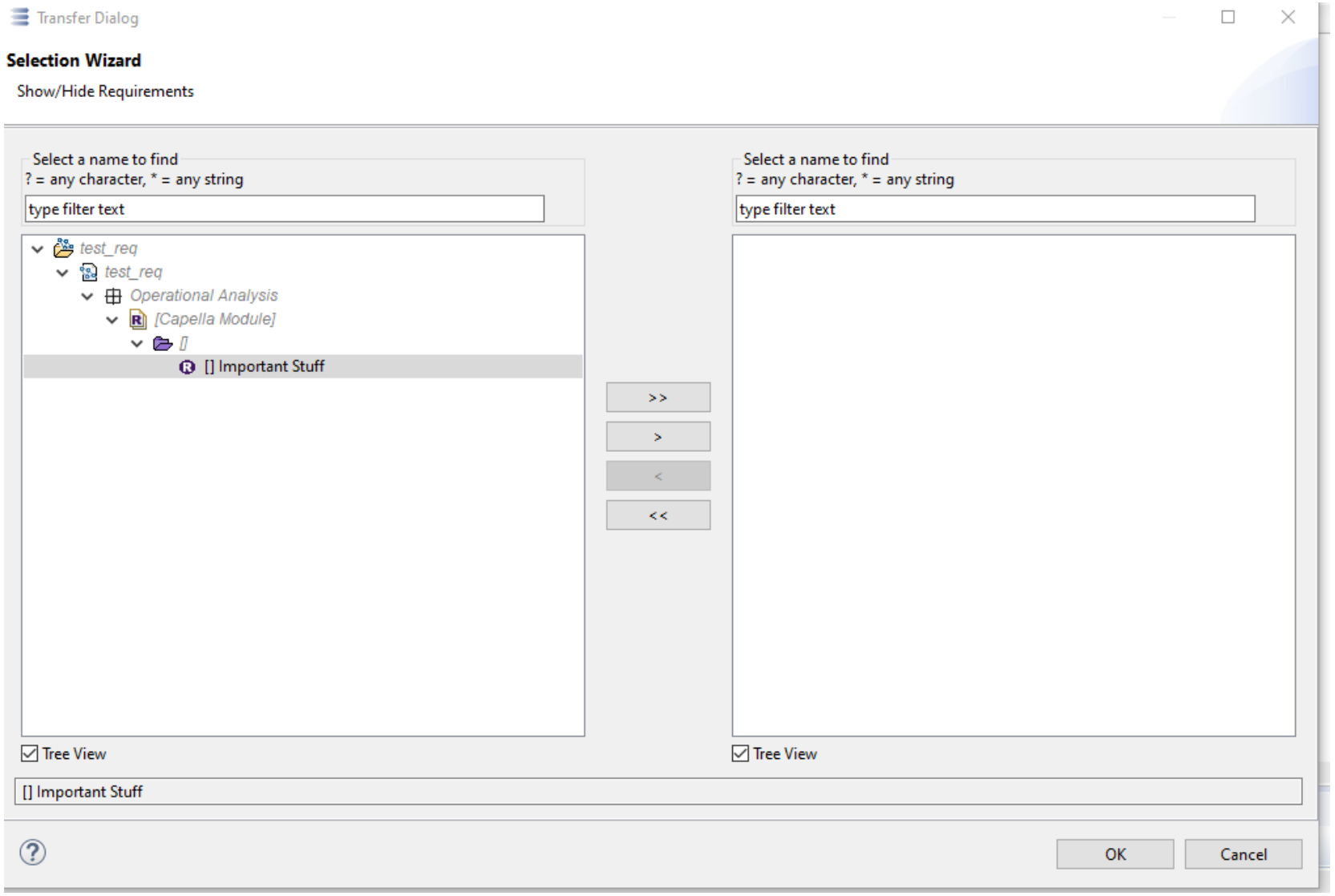


The only way to create requirements is through the Project Explorer.
[good side] Capella <could> connects to Doors ($$$) to import requirements.

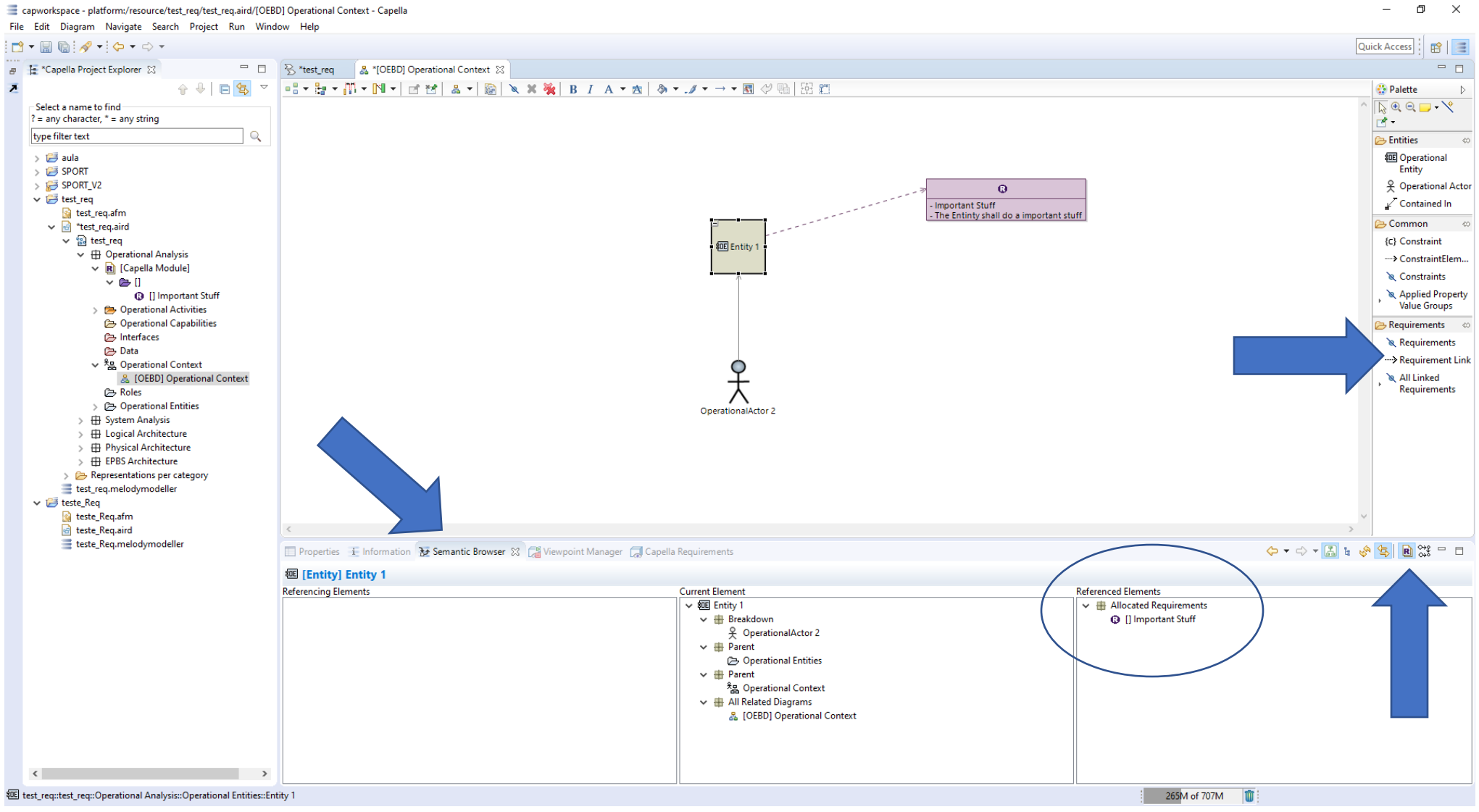
Requirements can be used in any view



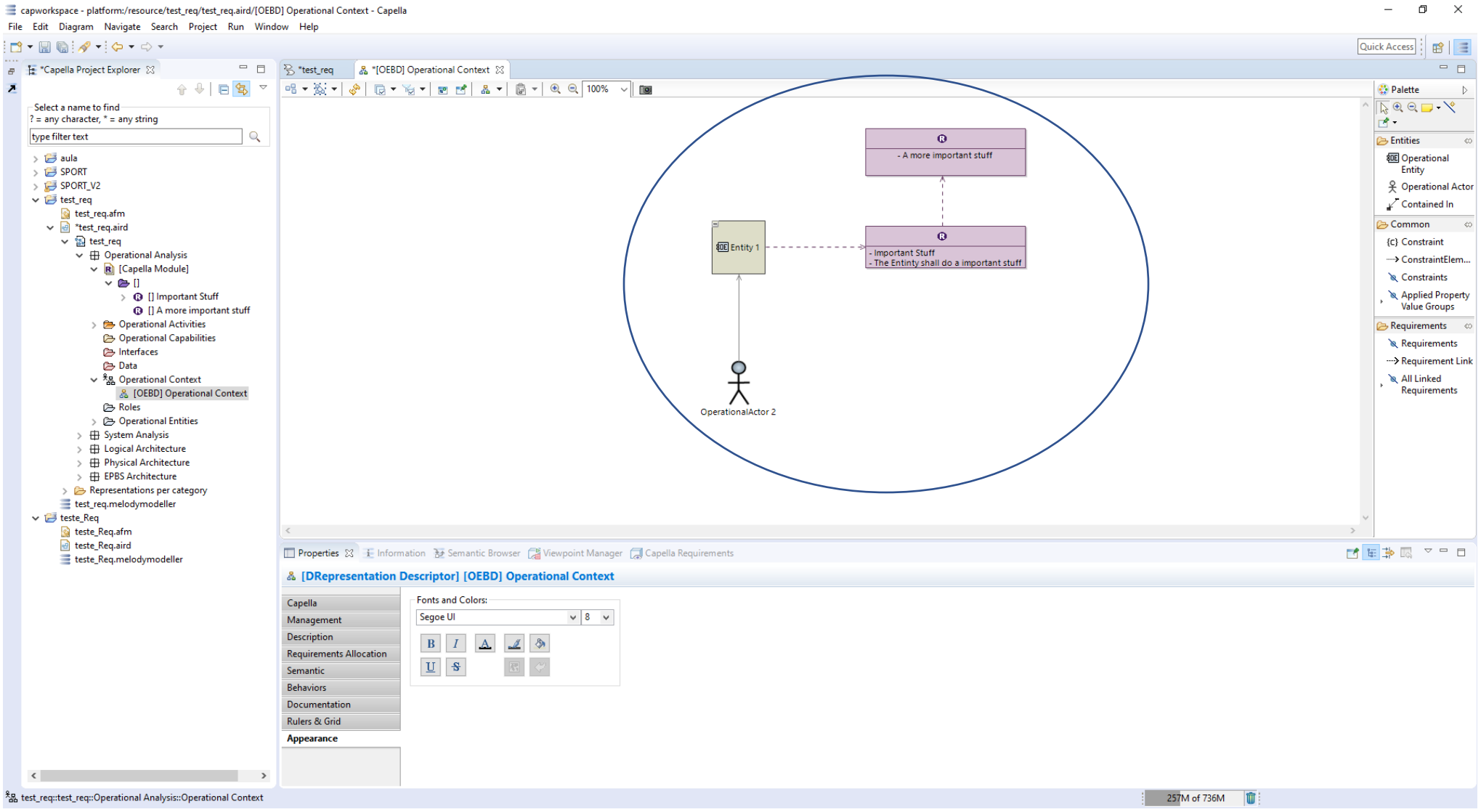
Select the requirements that want to use in the view.



Add a link / check relations

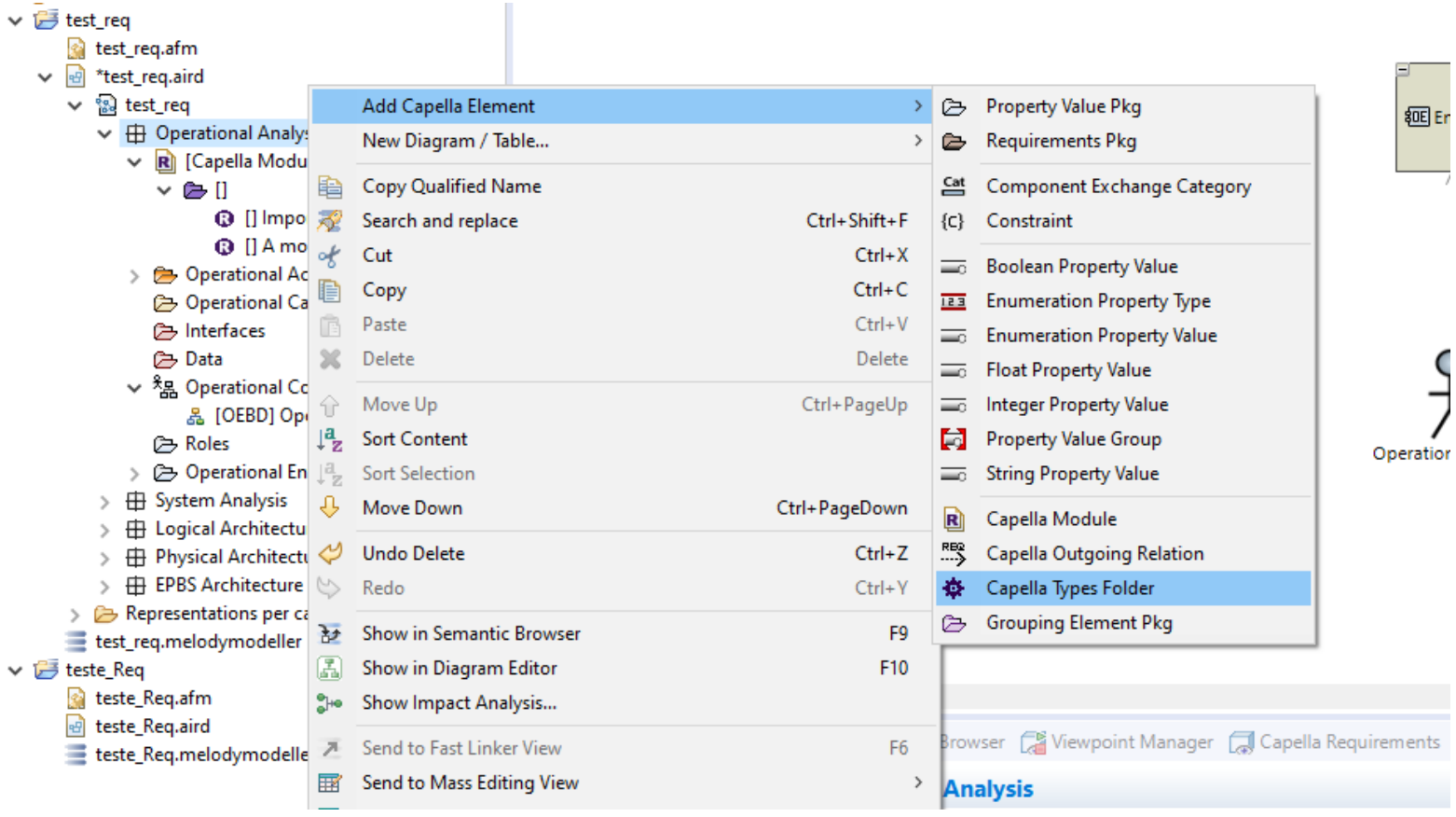


Requirement trees



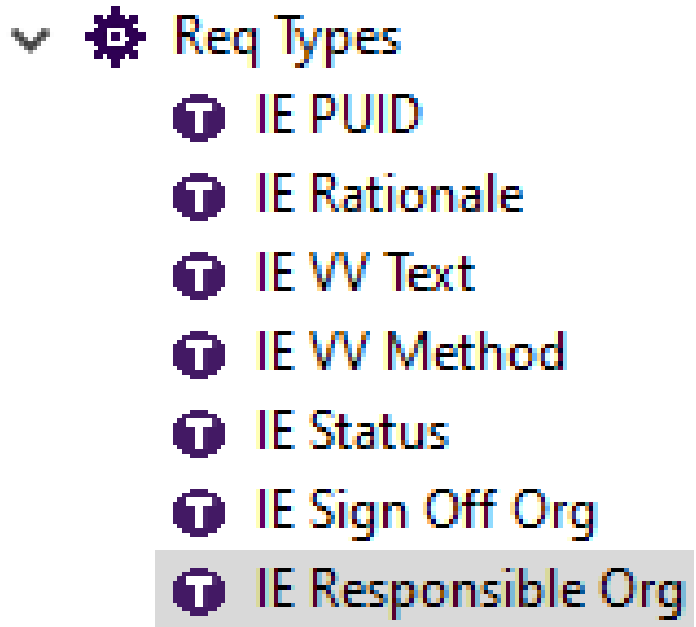
Add requirement metadata

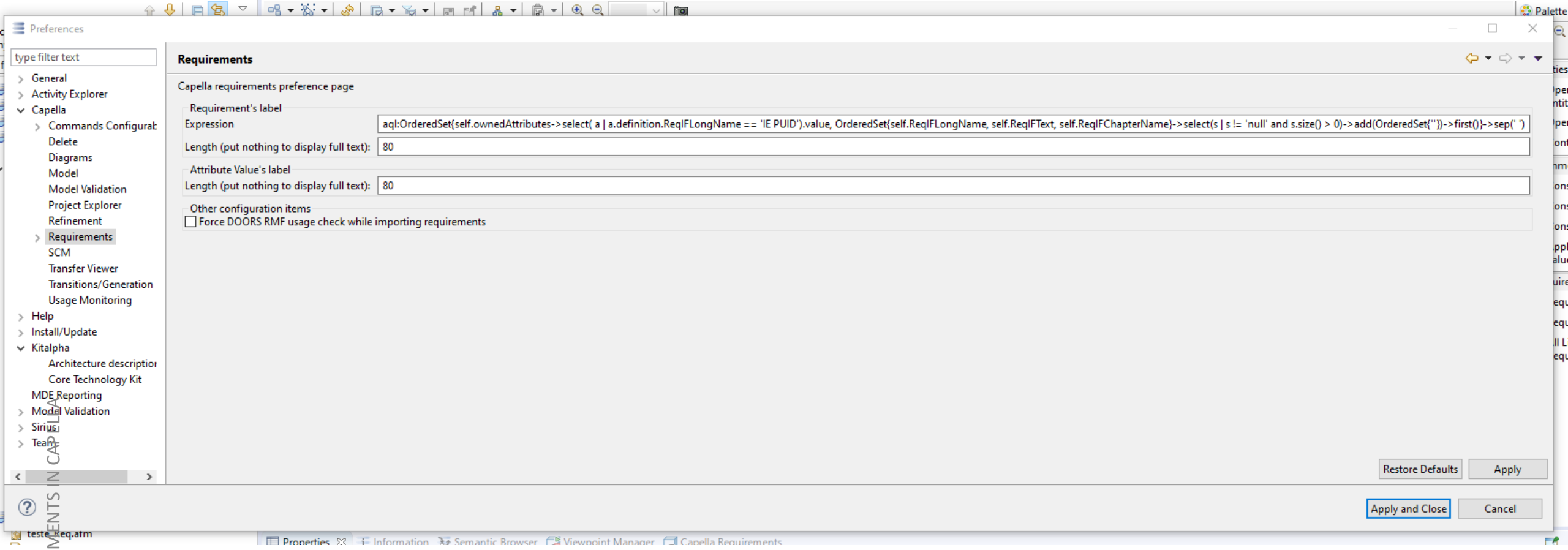
- It is required to create a new Type
- Create a Capella Types Folder → Rename Req Types



Requirement Data Type Definitions

- IE PUID (Requirement ID – name comes from DOORS)
- IE Rationale
- IE Verification Text
- IE Verification Method Expected
- IE Requirement Status
- IE Sign off Org
- IE Responsible Org

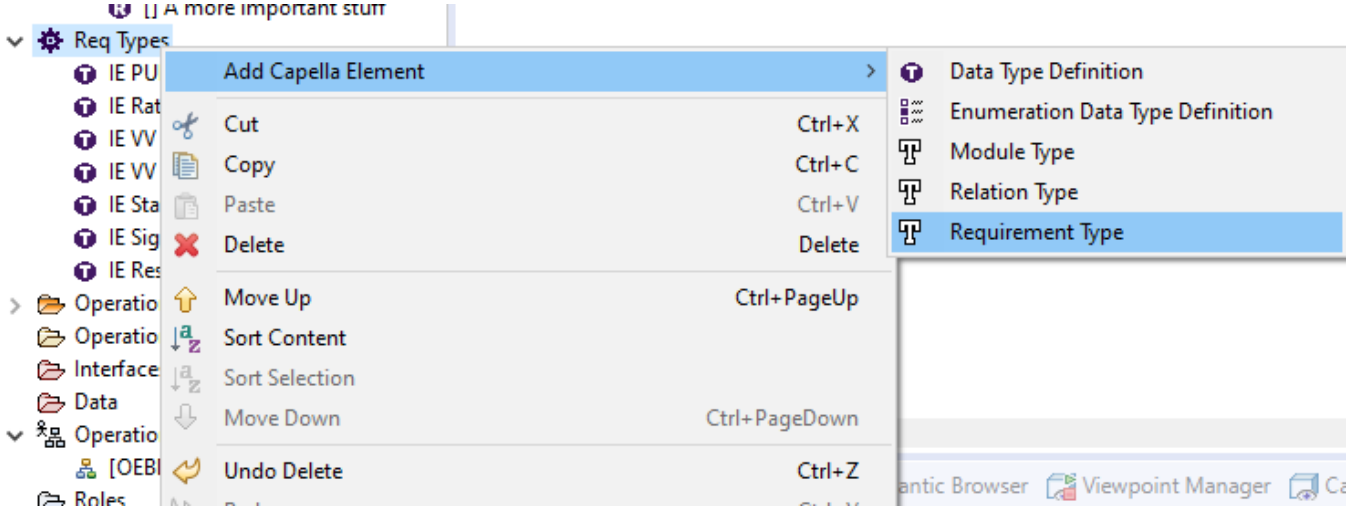


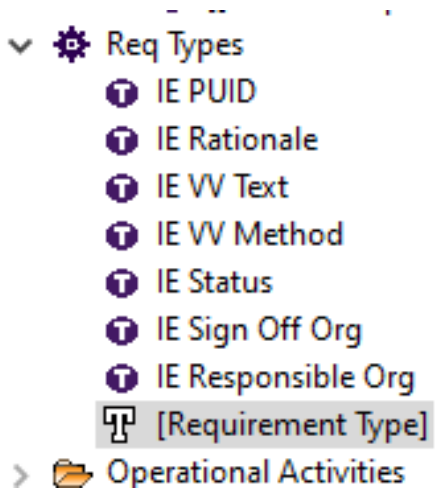


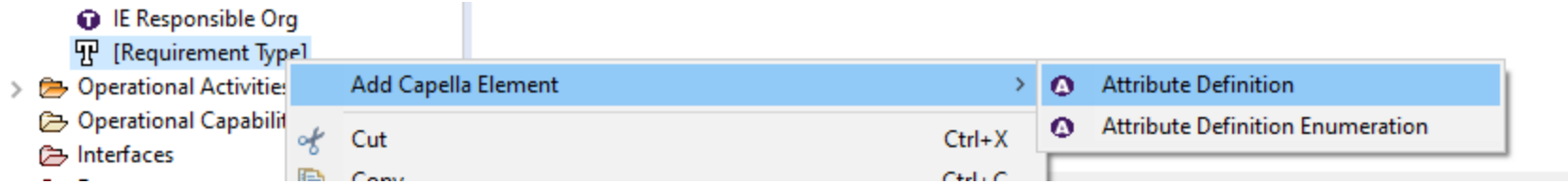
Annotation Query Language (AQL)

- aql:OrderedSet{self.ownedAttributes->select(a | a.definition.ReqIFLongName == '**IE PUID**').value, OrderedSet{self.ReqIFLongName, self.ReqIFText, self.ReqIFChapterName}->select(s | s != 'null' and s.size() > 0)->add(OrderedSet{''})->first()}->sep(' ')

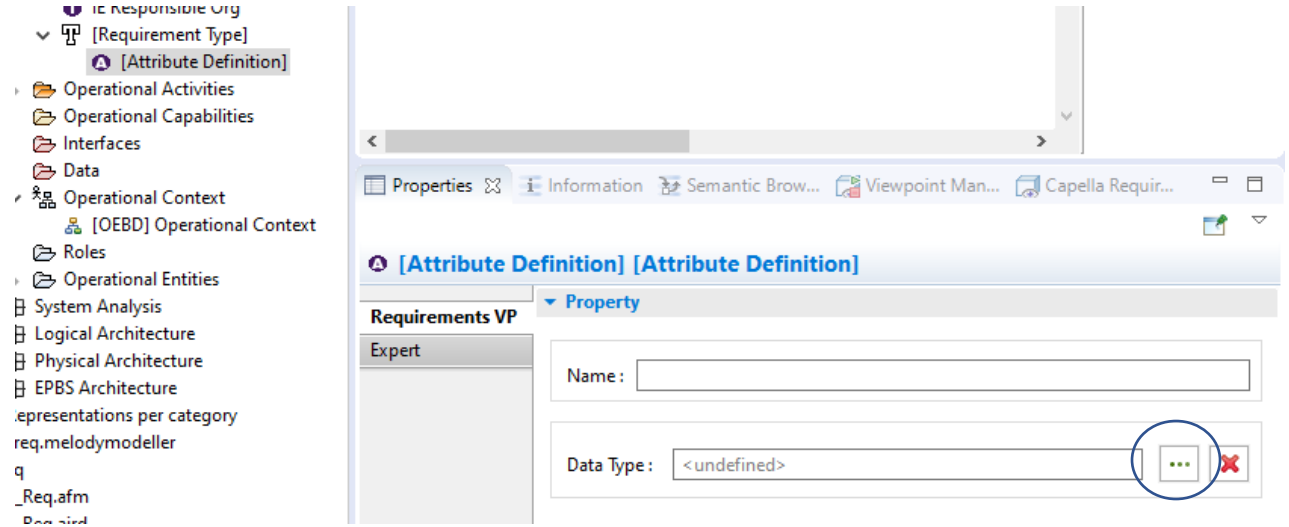
Create the Requirement Type that include the Data types as Attributes

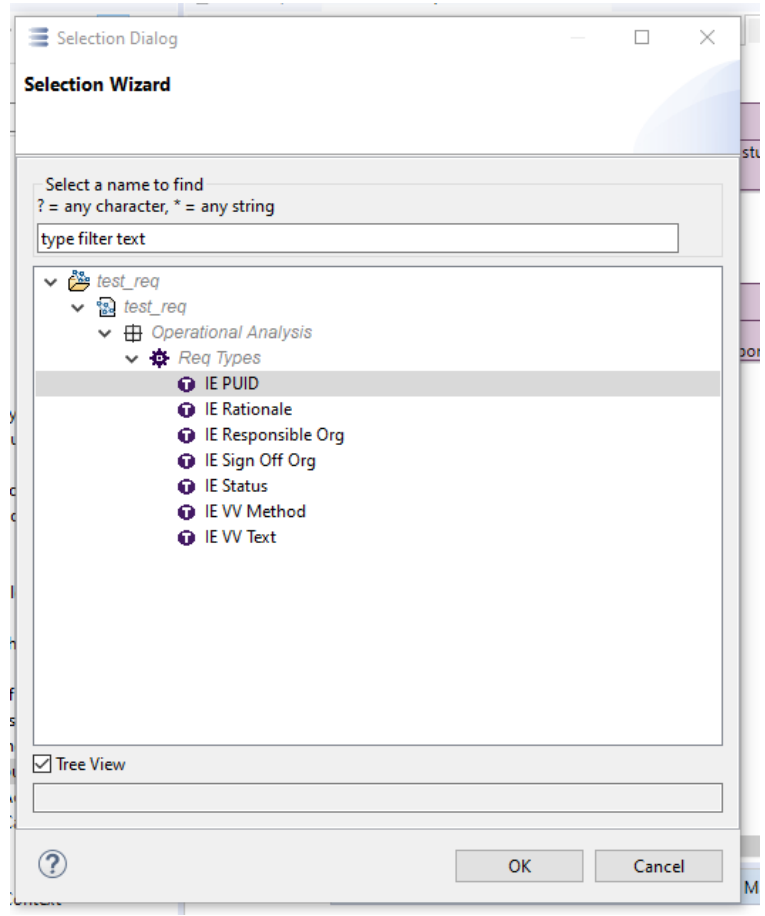


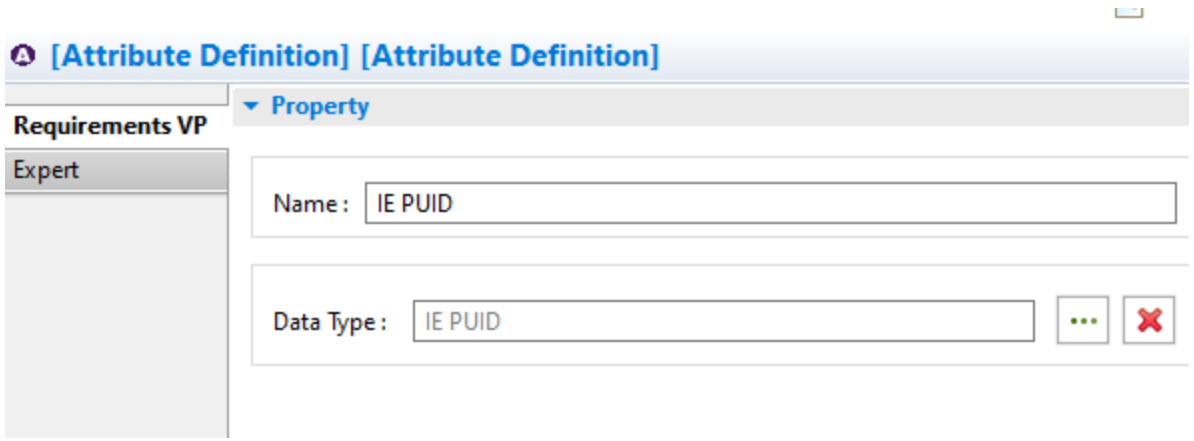




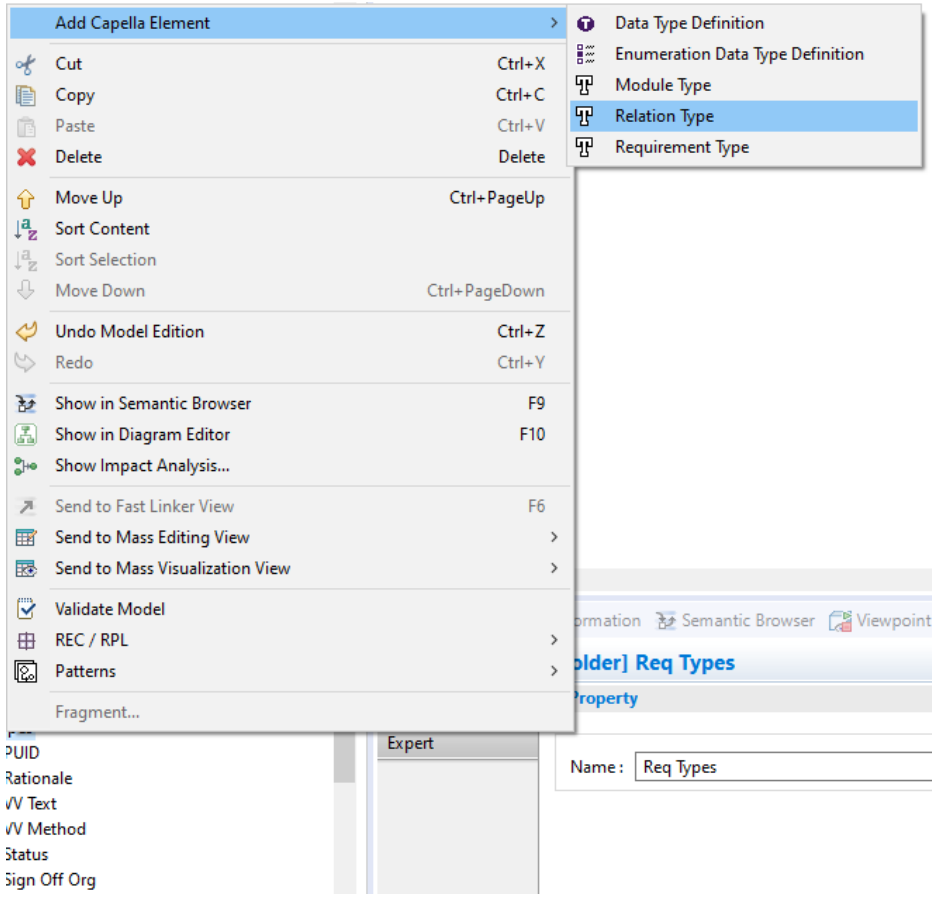
Configure the attribute

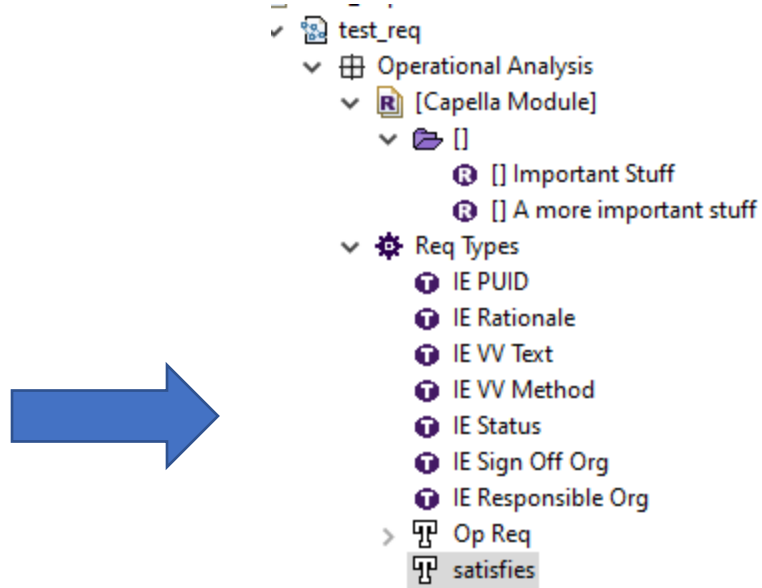


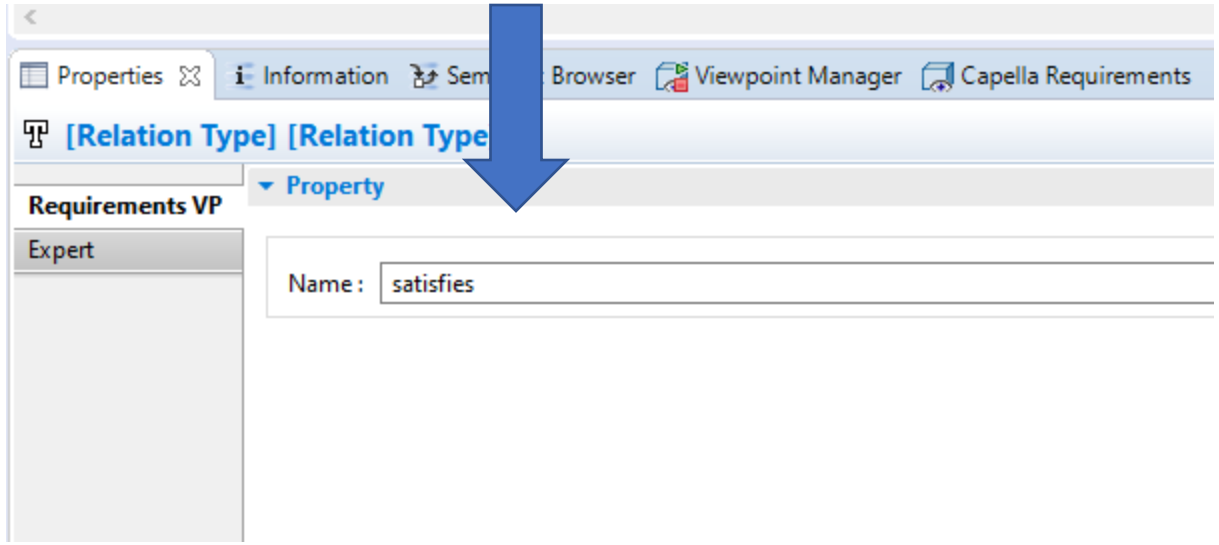




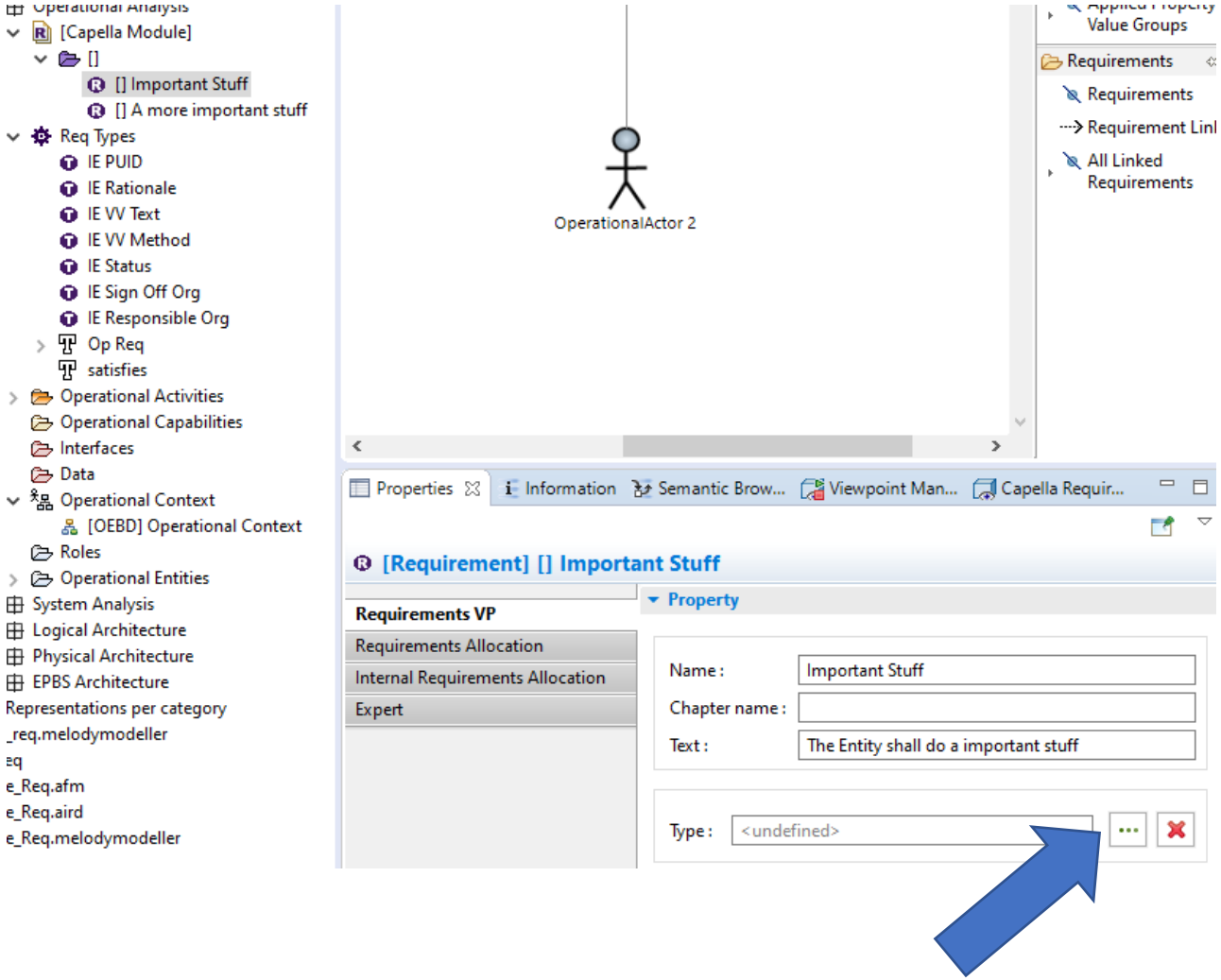
Add relation metadata

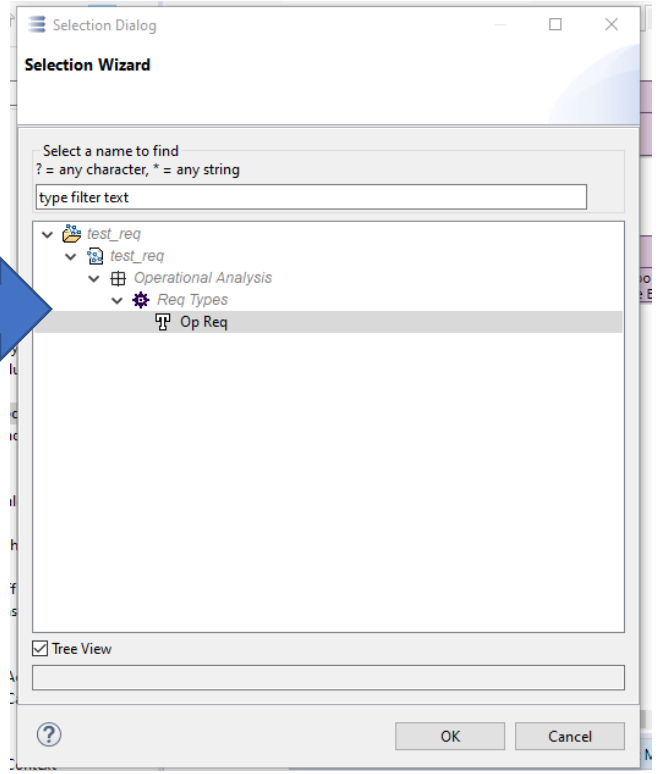




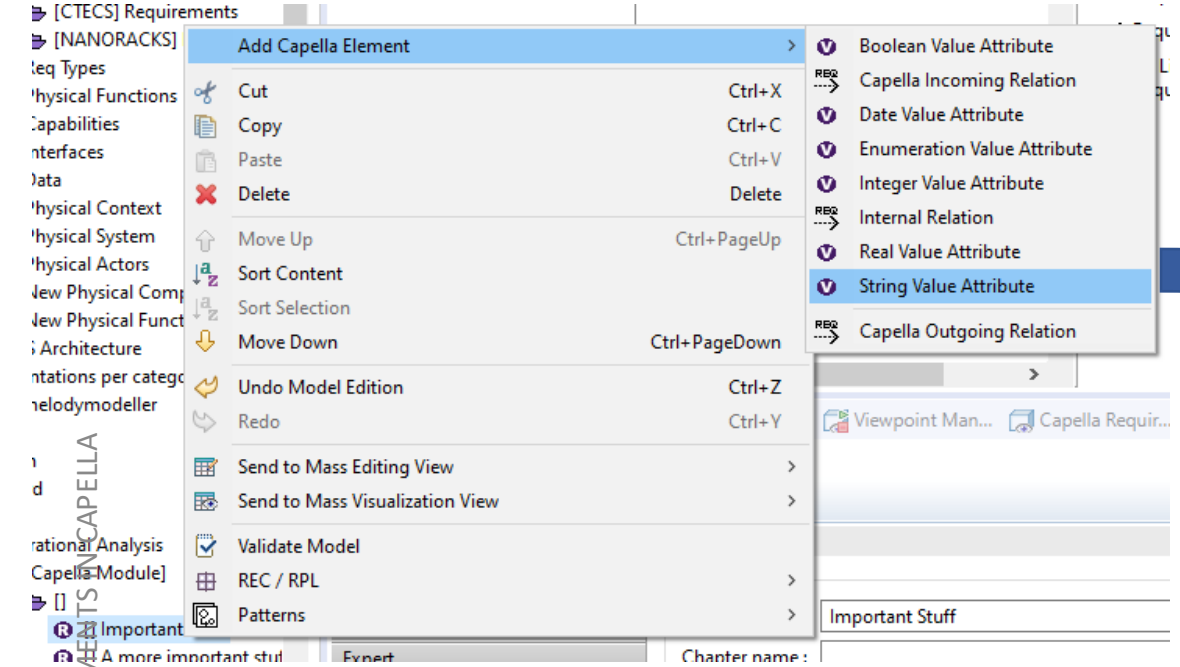


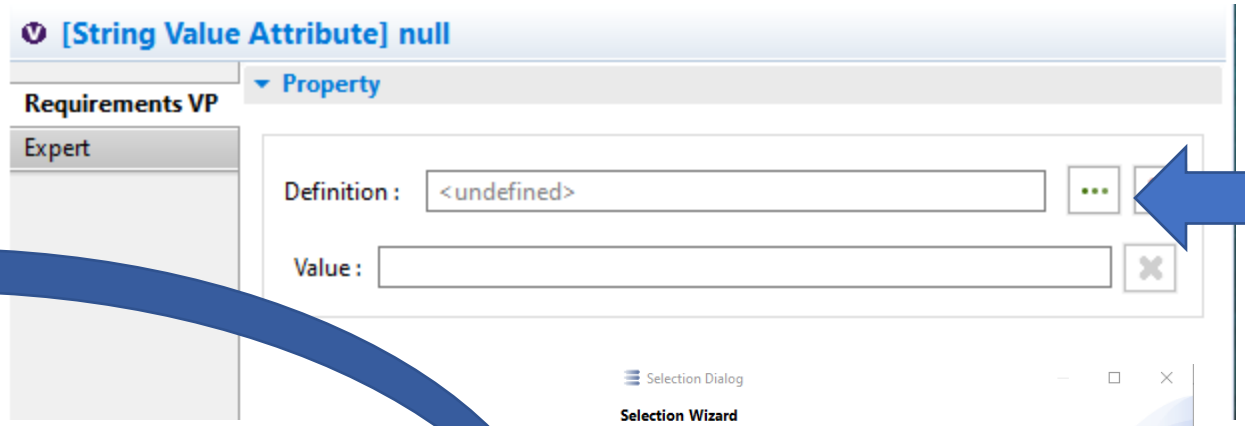
Apply to the Requirement Set

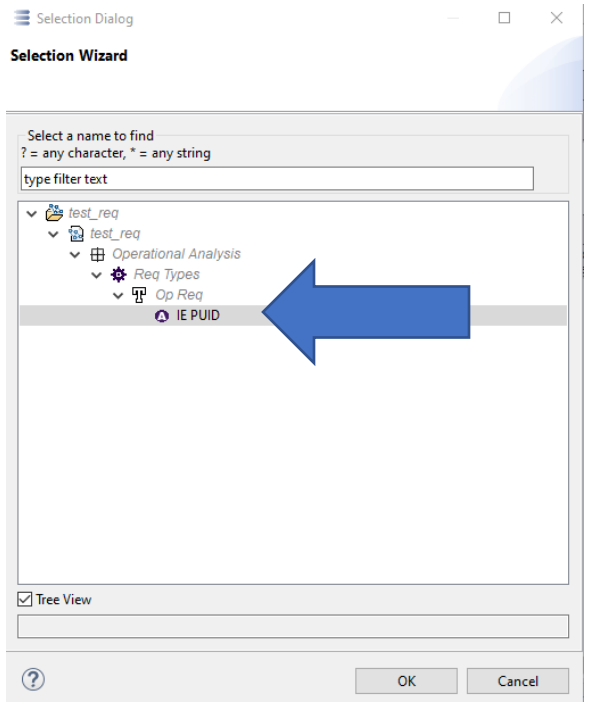


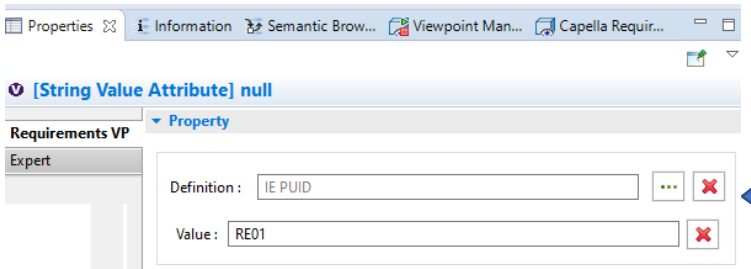


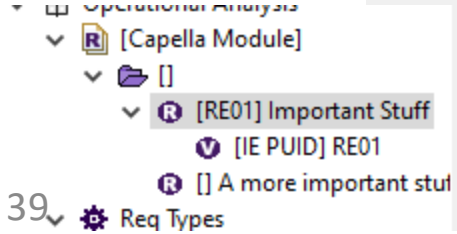
Creating the attributes









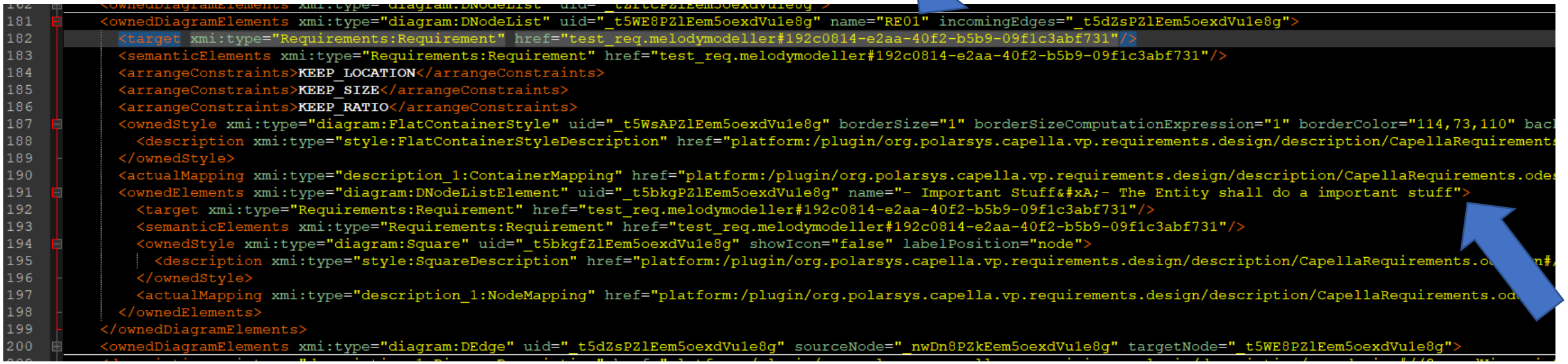


REQUIREMENTS IN CAPELLA

Each layer has a “default” req relation table

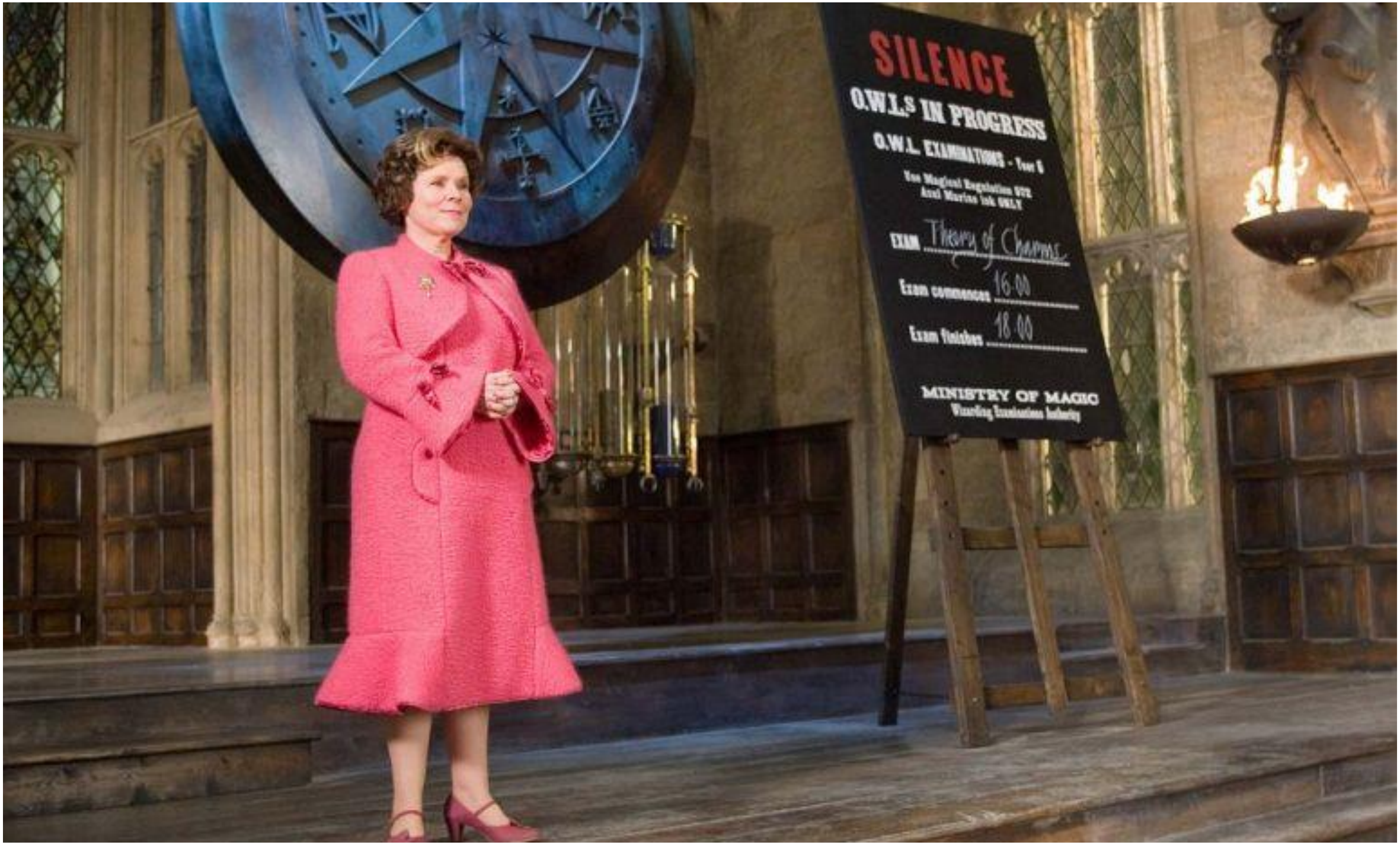
- Operational:
  - Activities X Requirements
- System:
  - System Function X Requirements
- Logical
  - Logical Functions x Requirements
  - Logical Component x Requirements
  - Logical Architecture Requirement Refinements
- Physical
  - Physical Functions x Requirements
  - Physical Component x Requirements
- EPBS
  - Configuration Itens x Requirements
  - EPBS Requirement Refinements

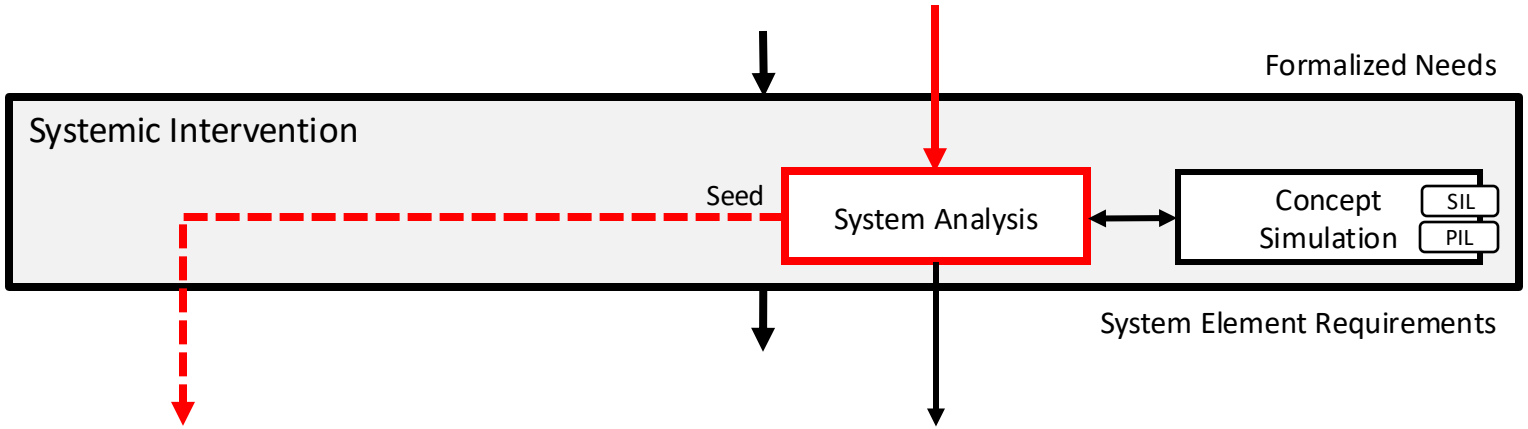
Everything is written in xmi



```
181  <ownedDiagramElements xmi:type="diagram:DNodeList" uid="_t5WE8PZlEem5oexdVu1e8g" name="RE01" incomingEdges="_t5dZsPZlEem5oexdVu1e8g">
182    <target xmi:type="Requirements:Requirement" href="test_req.melodymodeller#192c0814-e2aa-40f2-b5b9-09f1c3abf731"/>
183    <semanticElements xmi:type="Requirements:Requirement" href="test_req.melodymodeller#192c0814-e2aa-40f2-b5b9-09f1c3abf731"/>
184    <arrangeConstraints>KEEP_LOCATION</arrangeConstraints>
185    <arrangeConstraints>KEEP_SIZE</arrangeConstraints>
186    <arrangeConstraints>KEEP_RATIO</arrangeConstraints>
187    <ownedStyle xmi:type="diagram:FlatContainerStyle" uid="_t5WsAPZlEem5oexdVu1e8g" borderSize="1" borderSizeComputationExpression="1" borderColor="114,73,110" bac
188      <description xmi:type="style:FlatContainerStyleDescription" href="platform:/plugin/org.polarsys.capella.vp.requirements.design/description/CapellaRequirements
189    </ownedStyle>
190    <actualMapping xmi:type="description_1:ContainerMapping" href="platform:/plugin/org.polarsys.capella.vp.requirements.design/description/CapellaRequirements.odes
191    <ownedElements xmi:type="diagram:DNodeListElement" uid="_t5bkgPZlEem5oexdVu1e8g" name="- Important Stuff&#xA;- The Entity shall do a important stuff">
192      <target xmi:type="Requirements:Requirement" href="test_req.melodymodeller#192c0814-e2aa-40f2-b5b9-09f1c3abf731"/>
193      <semanticElements xmi:type="Requirements:Requirement" href="test_req.melodymodeller#192c0814-e2aa-40f2-b5b9-09f1c3abf731"/>
194      <ownedStyle xmi:type="diagram:Square" uid="_t5bkgfZlEem5oexdVu1e8g" showIcon="false" labelPosition="node">
195        <description xmi:type="style:SquareDescription" href="platform:/plugin/org.polarsys.capella.vp.requirements.design/description/CapellaRequirements.od
196      </ownedStyle>
197      <actualMapping xmi:type="description_1:NodeMapping" href="platform:/plugin/org.polarsys.capella.vp.requirements.design/description/CapellaRequirements.od
198    </ownedElements>
199  </ownedDiagramElements>
200  <ownedDiagramElements xmi:type="diagram:DEdge" uid="_t5dZsPZlEem5oexdVu1e8g" sourceNode="_nwDn8PZkEem5oexdVu1e8g" targetNode="_t5WE8PZlEem5oexdVu1e8g">
```

ARCADIA SYSTEM ANALYSIS


SILENCE
O.W.L.s IN PROGRESS
O.W.L. EXAMINATIONS - Year 5
EXAM Theory of Charms
Exam commences 16·00
Exam finishes 18·00
MINISTRY OF MAGIC

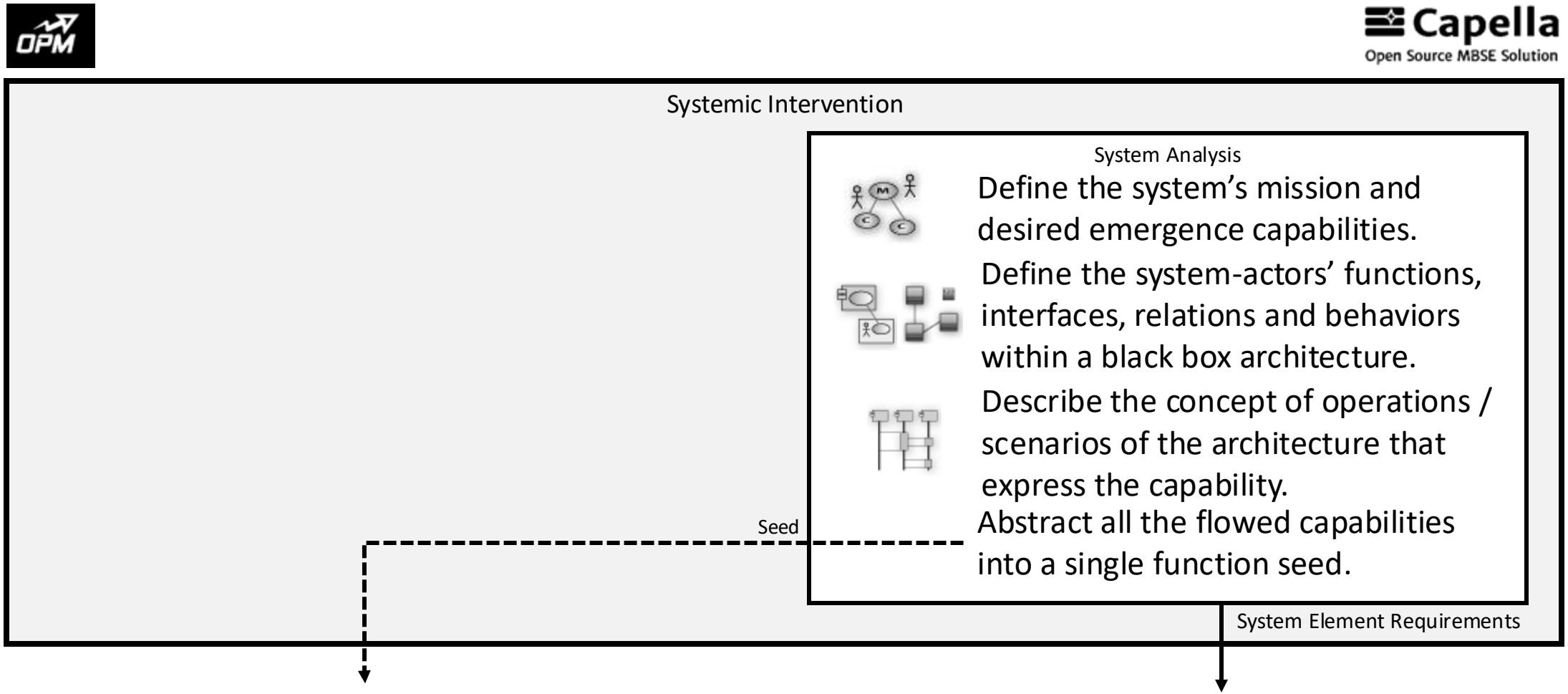


DCA-400-6

NOP

ROP

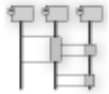
Systemic Intevenction



Analysis of the new system

Define the system's mission and desired emergence capabilities.

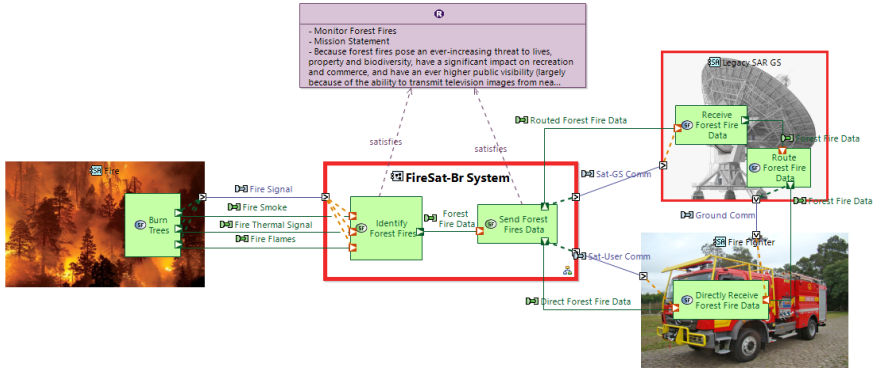
Define the system-actors' functions, interfaces, relations and behaviors within a black box architecture.



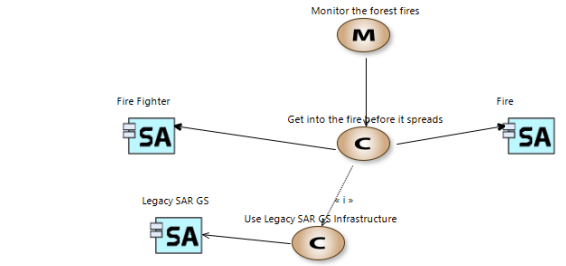
Describe the concept of operations / scenarios of the architecture that express the capability.

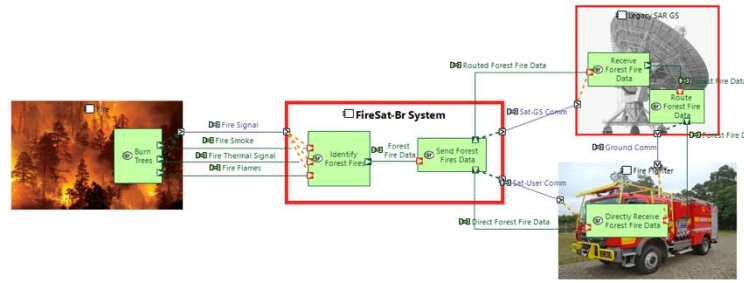
Seed

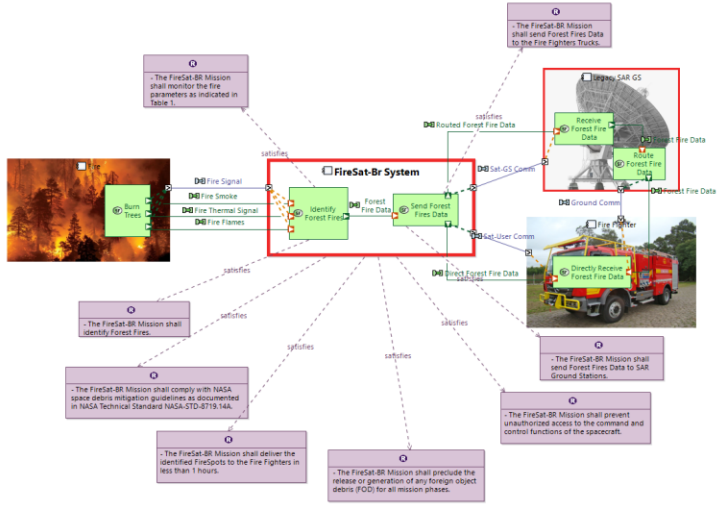
Abstract all the flowed capabilities into a single function seed.



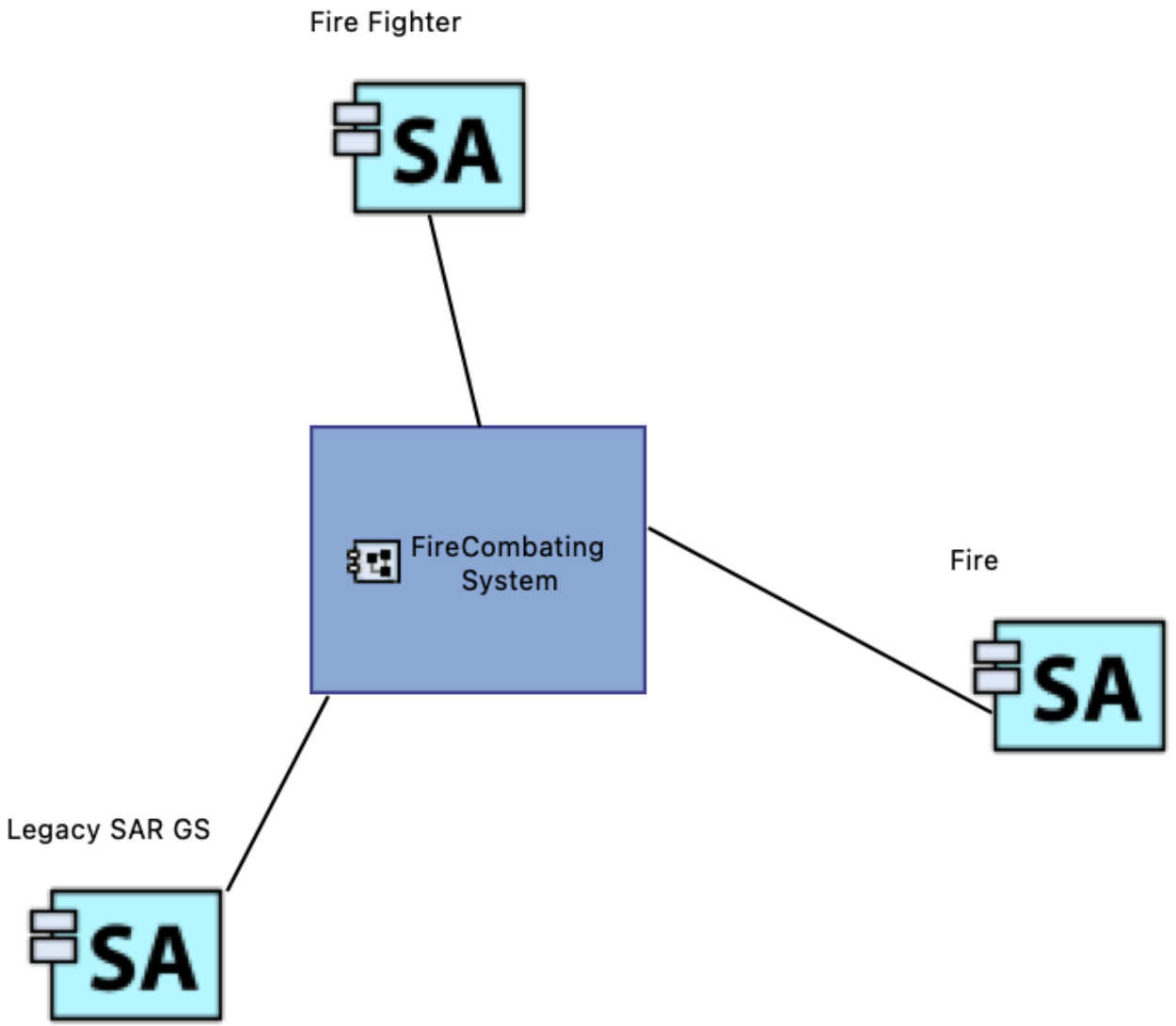
System Element Requirements







Context

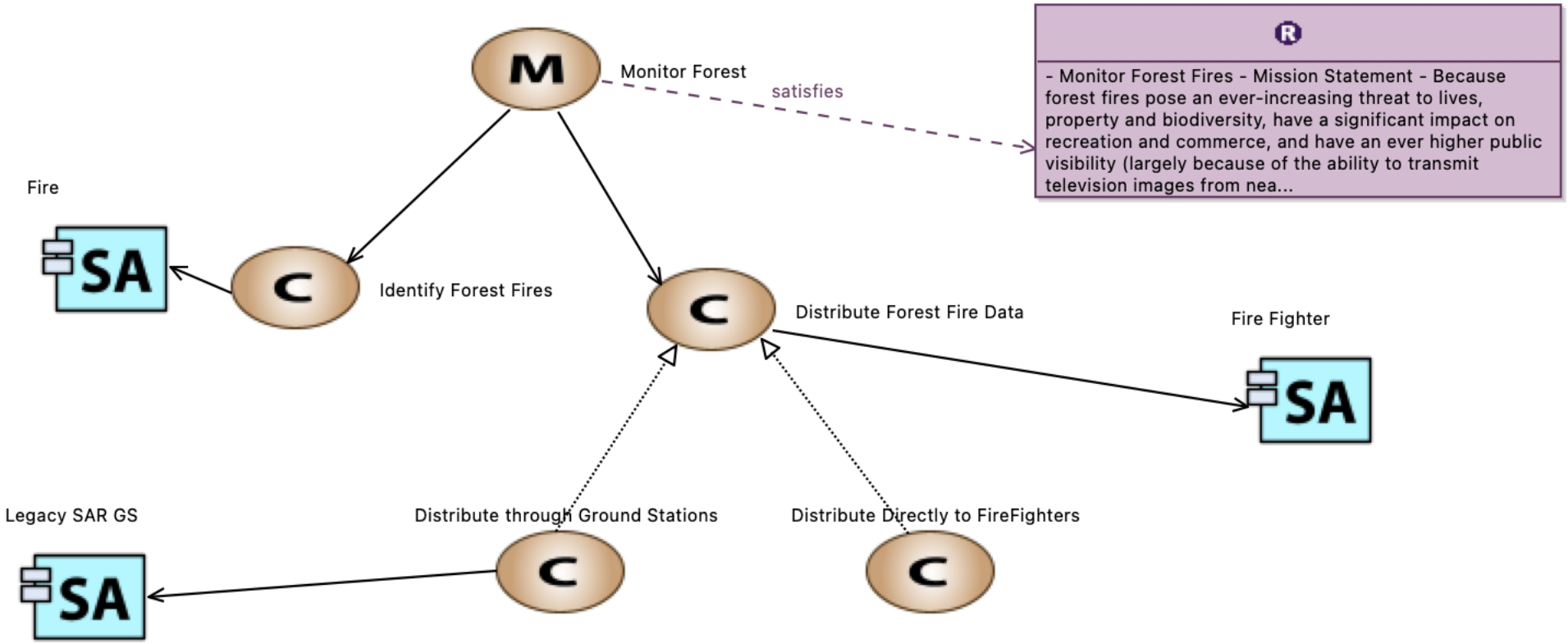

Fire Fighter
SA
FireCombating System
Fire
SA
Legacy SAR GS
SA

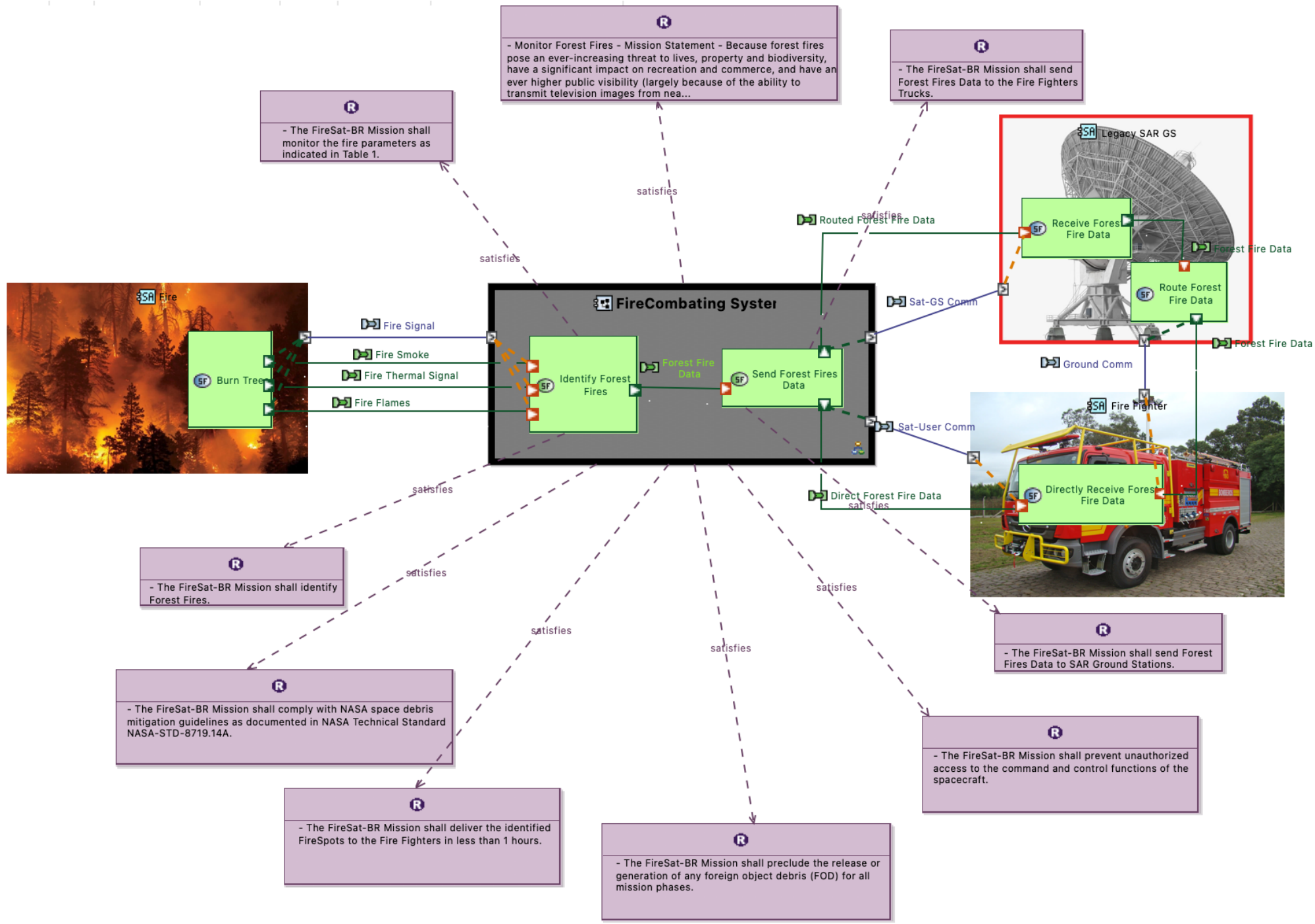
MISSION STATEMENT

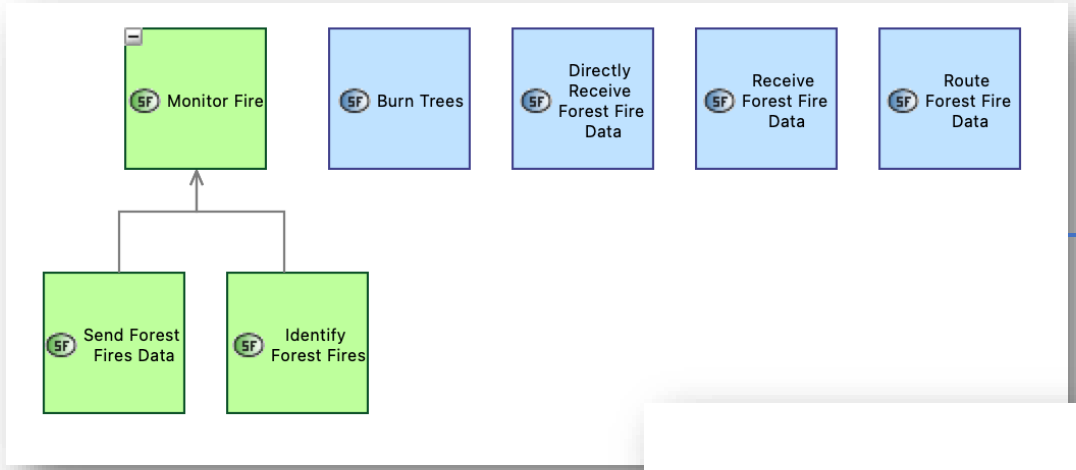
Because forest fires pose an ever-increasing threat to lives, property and biodiversity, have a significant impact on recreation and commerce, and have an ever higher public visibility (largely because of the ability to transmit television images from nearly anywhere in real time), the USFS needs a more effective system to identify and monitor them. In addition, it would be desired (but not required) to monitor forest fires for other nations; collect statistical data on fire outbreaks, spread, speed and duration, and provide other forest management data. This must be done at low cost to make the system affordable to the Forest Service and not give the perception of wasting money that could be better spent on fire-fighting equipment or personnel.

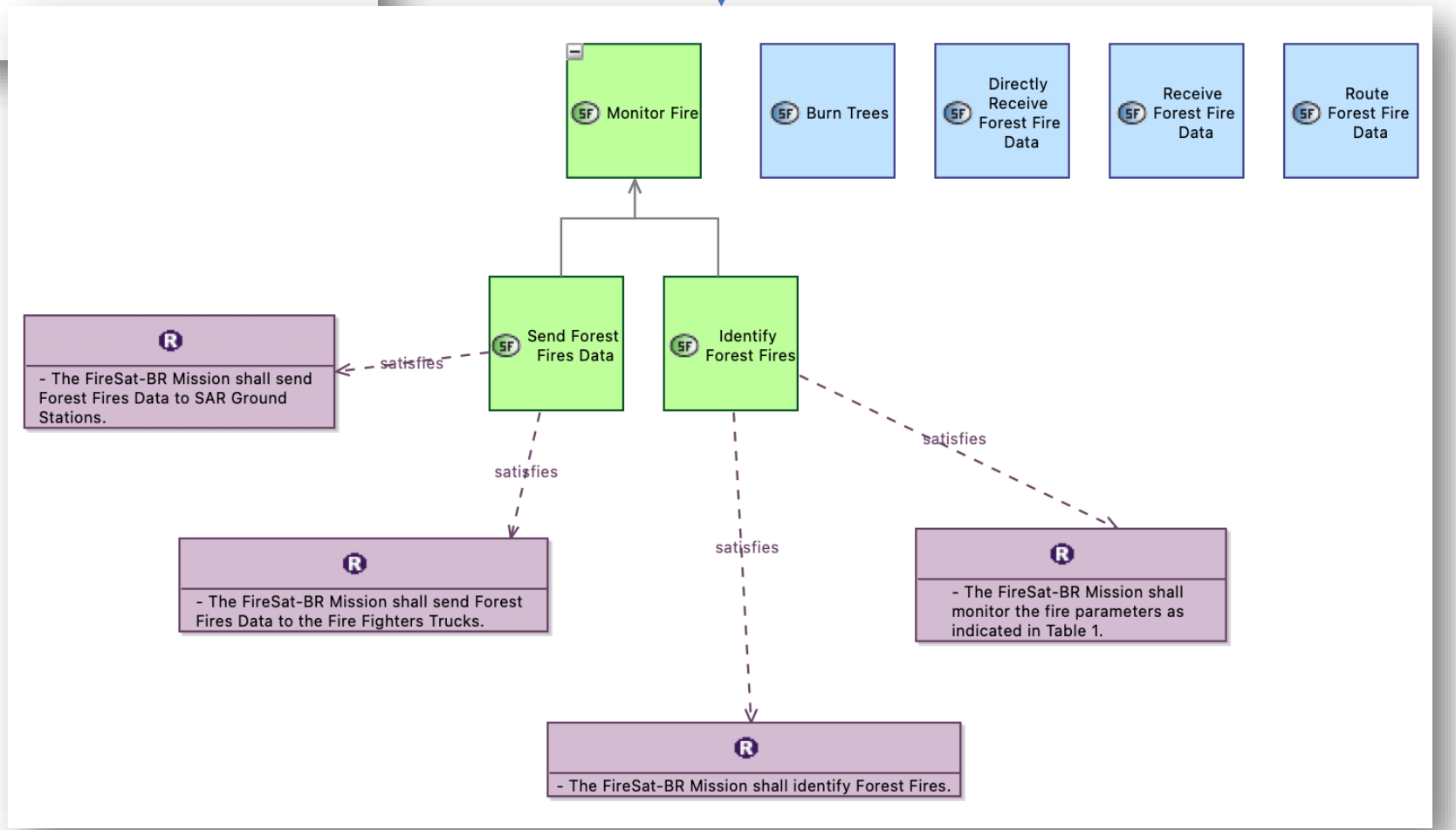
Ultimately, the Forest Service's fire monitoring office, fire management officers in the field, and individual firefighters and rangers fighting the fire will use the data. Data flow and formats must meet the needs of all the groups without specialized training and must allow them to respond promptly and efficiently to changing conditions.

(adapted from "Space Mission Engineering: the new SMAD, 2011")


M
Monitor Forest
satisfies
R
- Monitor Forest Fires - Mission Statement - Because forest fires pose an ever-increasing threat to lives, property and biodiversity, have a significant impact on recreation and commerce, and have an ever higher public visibility (largely because of the ability to transmit television images from nea...
Fire
SA
C
Identify Forest Fires
C
Distribute Forest Fire Data
Fire Fighter
SA
Legacy SAR GS
SA
Distribute through Ground Stations
C
Distribute Directly to FireFighters
C

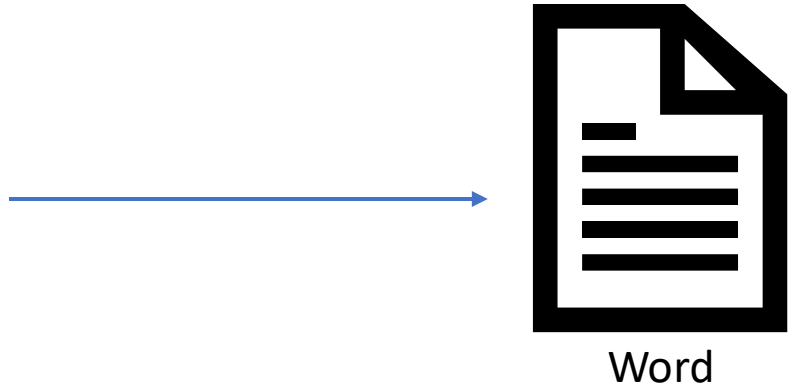

- Monitor Forest Fires - Mission Statement - Because forest fires pose an ever-increasing threat to lives, property and biodiversity, have a significant impact on recreation and commerce, and have an ever higher public visibility (largely because of the ability to transmit television images from nea...
- The FireSat-BR Mission shall send Forest Fires Data to the Fire Fighters Trucks.
- The FireSat-BR Mission shall monitor the fire parameters as indicated in Table 1.
satisfies
Legacy SAR GS
Receive Fores Fire Data
Route Forest Fire Data
Forest Fire Data
Routed Forest Fire Data
Sat-GS Comm
Ground Comm
Fire
Burn Tree
Fire Signal
Fire Smoke
Fire Thermal Signal
Fire Flames
FireCombating Syster
Identify Forest Fires
Forest Fire Data
Send Forest Fires Data
Sat-User Comm
Fire Fighter
Directly Receive Fores Fire Data
Direct Forest Fire Data
- The FireSat-BR Mission shall identify Forest Fires.
- The FireSat-BR Mission shall send Forest Fires Data to SAR Ground Stations.
- The FireSat-BR Mission shall comply with NASA space debris mitigation guidelines as documented in NASA Technical Standard NASA-STD-8719.14A.
- The FireSat-BR Mission shall prevent unauthorized access to the command and control functions of the spacecraft.
- The FireSat-BR Mission shall deliver the identified FireSpots to the Fire Fighters in less than 1 hours.
- The FireSat-BR Mission shall preclude the release or generation of any foreign object debris (FOD) for all mission phases.


Monitor Fire
Burn Trees
Directly Receive Forest Fire Data
Receive Forest Fire Data
Route Forest Fire Data
Send Forest Fires Data
Identify Forest Fires

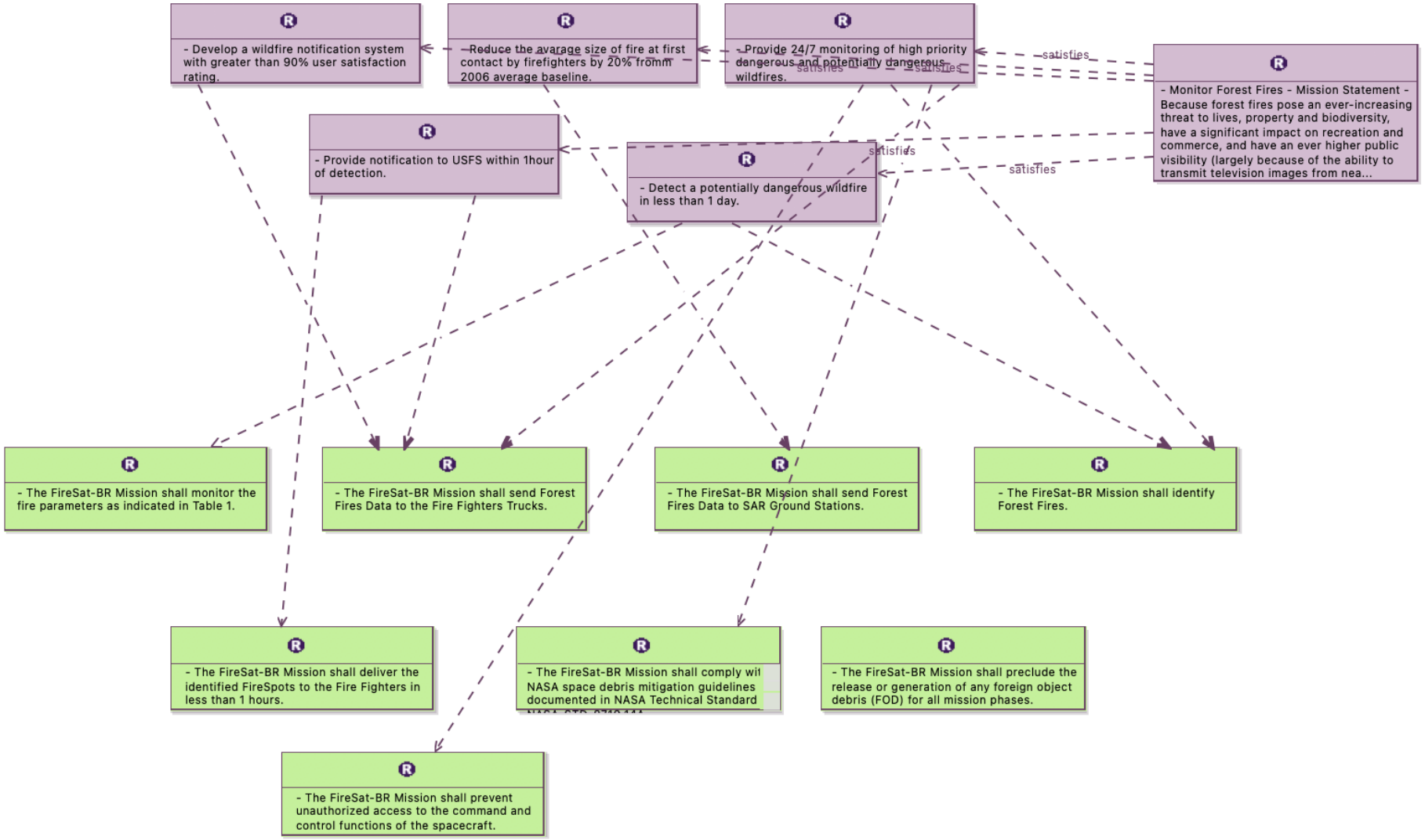

Monitor Fire
Burn Trees
Directly Receive Forest Fire Data
Receive Forest Fire Data
Route Forest Fire Data
Send Forest Fires Data
Identify Forest Fires
- The FireSat-BR Mission shall send Forest Fires Data to SAR Ground Stations.
satisfies
satisfies
satisfies
satisfies
- The FireSat-BR Mission shall send Forest Fires Data to the Fire Fighters Trucks.
- The FireSat-BR Mission shall monitor the fire parameters as indicated in Table 1.
- The FireSat-BR Mission shall identify Forest Fires.

Requirements

- System Analysis
  - [Capella Module]
    - Mission Statement
      - [Mission Statement] Because forest fires pose an ever-increasing threat to lives...
    - Mission Requirements
      - Functional Requirements
        - [MIS-XXX] The FireSat-BR Mission shall identify Forest Fires.
          - [IE PUID] MIS-XXX
          - [Rationale] null
          - [VV Method] null
          - [VV Success Criteria] null
          - [VV Phase] null
          - [VV Procedure] null
          - [VV Report] null
        - [MIS-XXX] The FireSat-BR Mission shall send Forest Fires Data to SAR Ground Stat...
        - [MIS-XXX] The FireSat-BR Mission shall send Forest Fires Data to the Fire Fighte...
        - [MIS-XXX] The FireSat-BR Mission shall monitor the fire parameters as indicated ...
      - Non-Functional Requirements

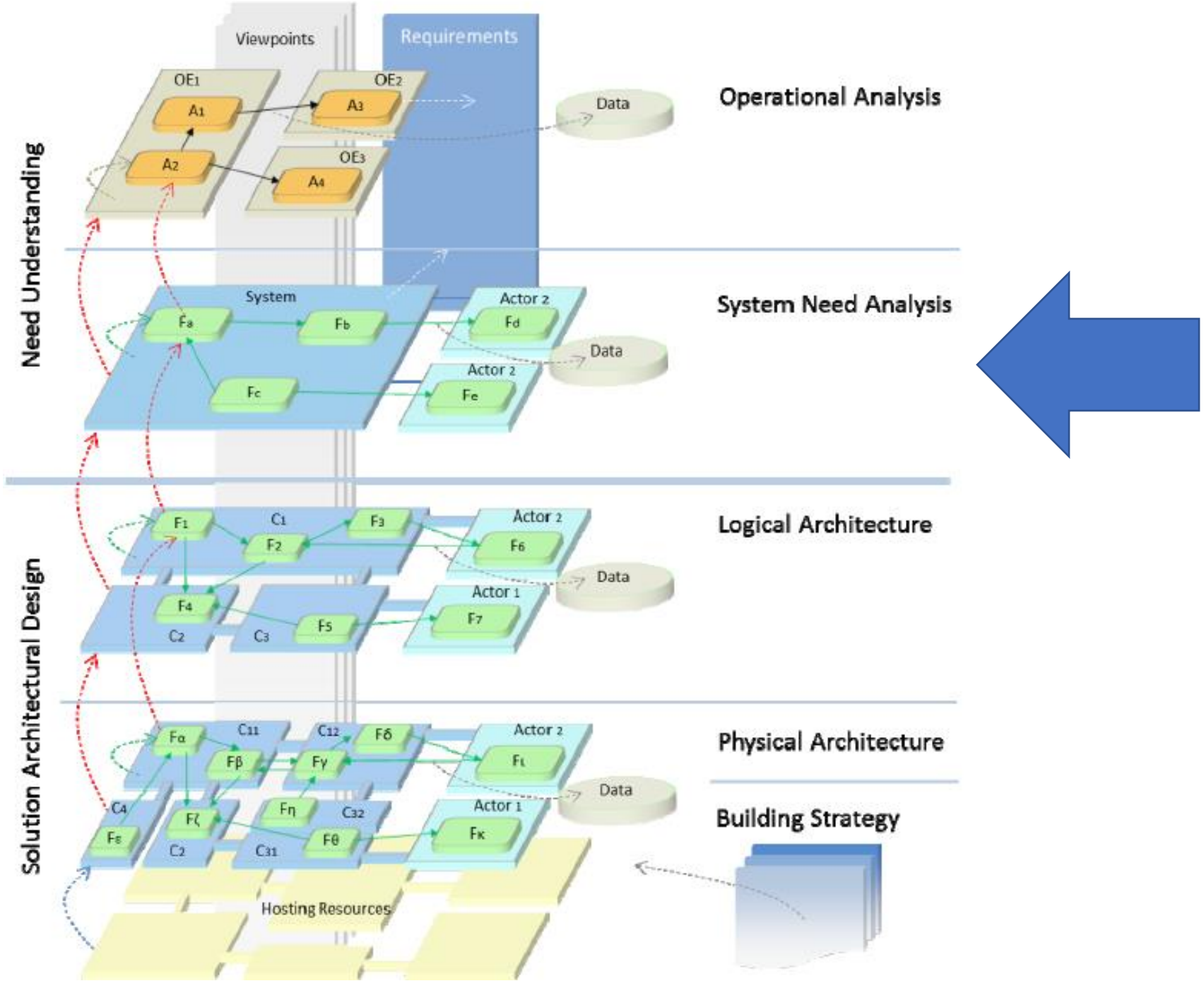


Traceability req_user – req_sys (nop-rop)



Fab: publicação da rop

- Descrever o que o sistema tem que fazer para os stakeholders (OMs)
- Descrever o conceito de operação geral desse sistema com os stakeholders.
- Rastrear as necessidades aos requisitos.
- Justificar as interfaces e funções.
- Formaliza o que o sistema tem que prover sem explicar como e dar margem para os fornecedores.


Viewpoints
Requirements
Need Understanding
Solution Architectural Design
OE1
OE2
OE3
A1
A2
A3
A4
Data
Operational Analysis
System
Actor 2
Actor 2
Actor 1
Fa
Fb
Fc
Fd
Fe
Data
System Need Analysis
F1
C1
F2
F3
F4
C2
C3
F5
F6
F7
Data
Logical Architecture
Fα
C11
Fβ
C12
Fγ
Fδ
Fε
Fζ
Fη
Fθ
C4
C2
C31
C32
Fι
Fκ
Data
Physical Architecture
Building Strategy
Hosting Resources

09:21

WHAT IS IN THE SYSTEM ANALYSIS (SA)

09:21

System Analysis

"What the system must achieve for users"

"What the system has to accomplish for the users"

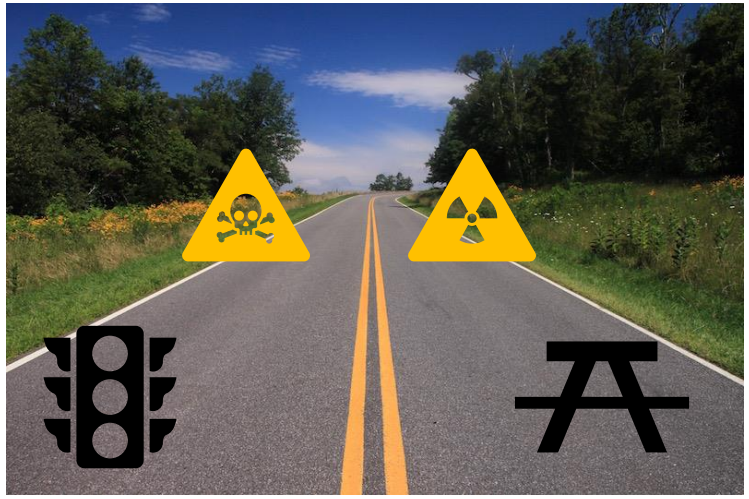
- The SA perspective defines the **expectations of the system**, that is to say **what the system has to perform for users**: it builds an external functional analysis, based on the OA and input textual requirements, to identify in response functions, services and expected system behaviors, necessary to its users.
  - external functional analysis as a response to identify the system functions needed by its users (e.g. "calculate the optimal path" and "detect a threat"), limited by the non-functional properties asked for.
- The System is identified as a modeling element at this level. It is a "black box" containing no other structural elements, only allocated Functions.

- The purpose of system needs analysis (referred to as SA further in the text) is to define the contribution expected of the system to users' needs, as they are described in the previous operational analysis (OA) and/or in the form of requirements expressed by the client.
- SA delimits the functions required of the system, distinguishing them from those assumed by the users or external systems.
- *It is essential to limit the functional analysis conducted in SA to the sole capture of the need, and* ***only of need, excluding any implementation choice or details***. This allows freedom of choice to be maintained during the subsequent development of the solution,

09:21

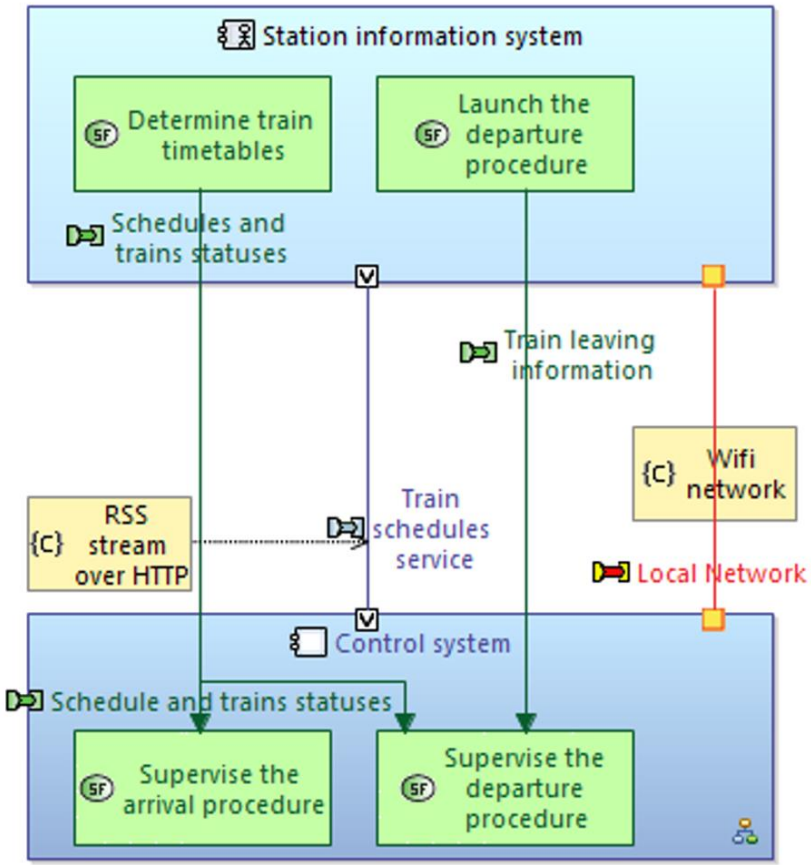
Perform Capability COMPROMISE Analysis

- Define the **essential characteristics** **necessary for the fulfillment of each operational capability** (the problem space), to uncover different alternative orientations likely to satisfy these required capabilities as well as the **criteria for associated appreciation and choice** (the solution space), and to compare these orientations to find the one(s) exhibiting the best compromise between the desirable characteristics.
- These parameters may concern the **functional contribution** **and the expected performance of the system**, obviously, but much further: organization, doctrines, procedures and users' roles, human factors, skills and training, logistics footprint and deployment conditions, hosting facilities, etc. **Quantitative and qualitative metrics** should be defined to evaluate the satisfaction conditions for each of these parameters.
- Capability analysis considers **much more general aspects** than the functional issues: as the client organization, organizational operating principles, roles and responsibilities, nature and infrastructure capacity, safety, human factors and users' skills/training, logistics, acquisition and operation costs, but also the potential complexity and implementational risks.



Perform Functional/non-Functional Need Analysis

- The intent is to **formalize the functional needs allocated to the system**, and to identify constraints, namely non-functional, to which it will have to respond through its use under operational conditions
- Assess the operational capabilities to which the system will have to contribute, taking the preliminary capability trade-off analysis (of "system capabilities") into account - only **needs-related considerations** should be included in this perspective dedicated to the expression of the system needs as required by users
- In the event that actors or external systems are imposed by the client (or the state of the art) and exhibit a complex or critical level of interactions with the system, it is recommended to carry out minimal functional and non-functional analyses for these external systems or actors, and to compare them with the SA, to ensure the compatibility between the two. At this point, an **analysis of available interfaces is desirable**, to verify that planned functionalities and interactions will be possible.
- Another way to address the needs functional analysis consists of implementing each functional requirement into a few functions and exchanges between them (often the verbs of the requirement), the manipulated data (the names) and actors or external systems..



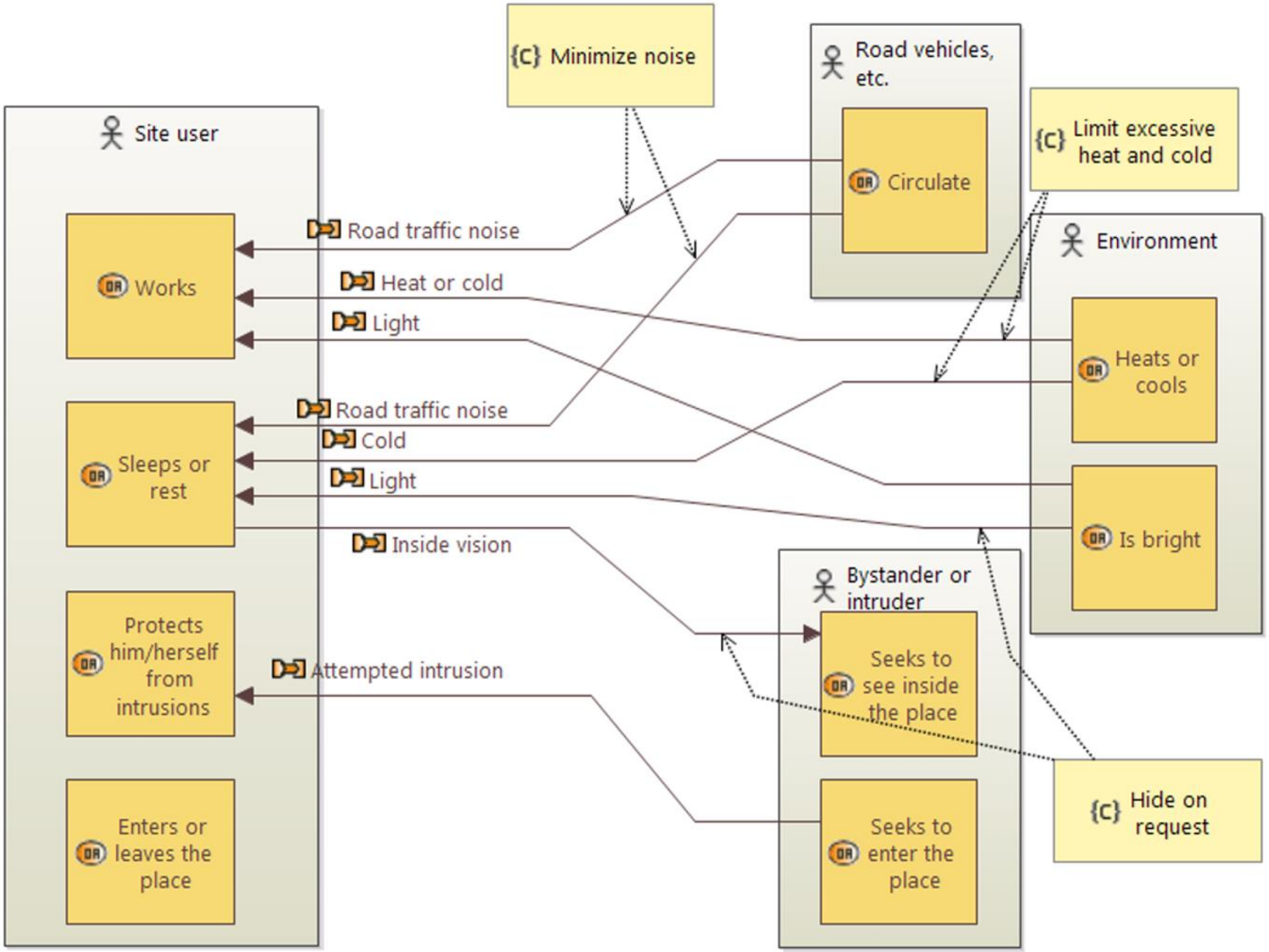
Formalize and Consolidate the System Needs

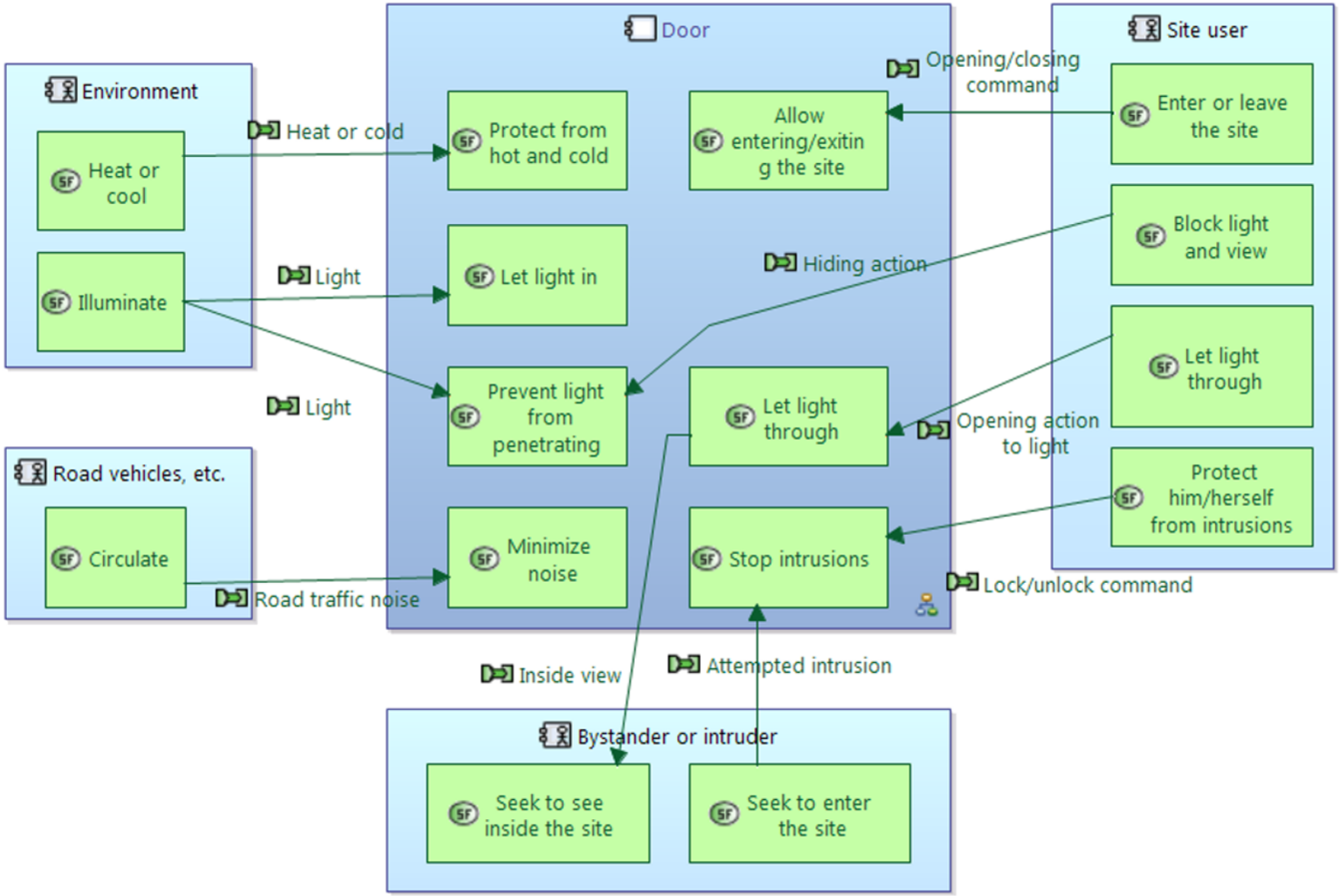
- The good understanding and consolidation of system needs rely on the three dimensions mentioned earlier, which are the **OA, requirements and the functional analysis** of the system need.
- It is through their *comparison* that consistency and completeness of the system need is assured: **Are all activities and operational processes correctly taken into account in the functional analysis? Are all functional requirements (or even nonfunctional) correctly captured? Is there any incompatibility between them?**
- It may even be the case that the functional needs analysis results in modifying the OA (e.g. changing an operator role for a more secure behavior, or reviewing the distribution of roles should an opportunity for system automation emerge); or alternatively, that the functional analysis reveals an inconsistency or something missing in the requirements.

09:21

Arcadia method – SYSTEM analysis summary

Capability Analysis	**essential characteristics necessary for the fulfillment of each operational capability**
Functional/non-Functional Need Analysis	**formalize the functional needs allocated to the system**
Formalize and Consolidate the System Needs	**OA, requirements and the functional analysis** of the system need


{C} Minimize noise
Road vehicles, etc.
Site user
Circulate
{C} Limit excessive heat and cold
Environment
Road traffic noise
Works
Heat or cold
Light
Heats or cools
Road traffic noise
Cold
Sleeps or rest
Light
Is bright
Inside vision
Bystander or intruder
Protects him/herself from intrusions
Attempted intrusion
Seeks to see inside the place
{C} Hide on request
Enters or leaves the place
Seeks to enter the place


Door
Environment
Heat or cool
Illuminate
Heat or cold
Light
Light
Protect from hot and cold
Let light in
Prevent light from penetrating
Allow entering/exiting the site
Let light through
Minimize noise
Stop intrusions
Opening/closing command
Hiding action
Opening action to light
Lock/unlock command
Site user
Enter or leave the site
Block light and view
Let light through
Protect him/herself from intrusions
Road vehicles, etc.
Circulate
Road traffic noise
Inside view
Attempted intrusion
Bystander or intruder
Seek to see inside the site
Seek to enter the site

09:21

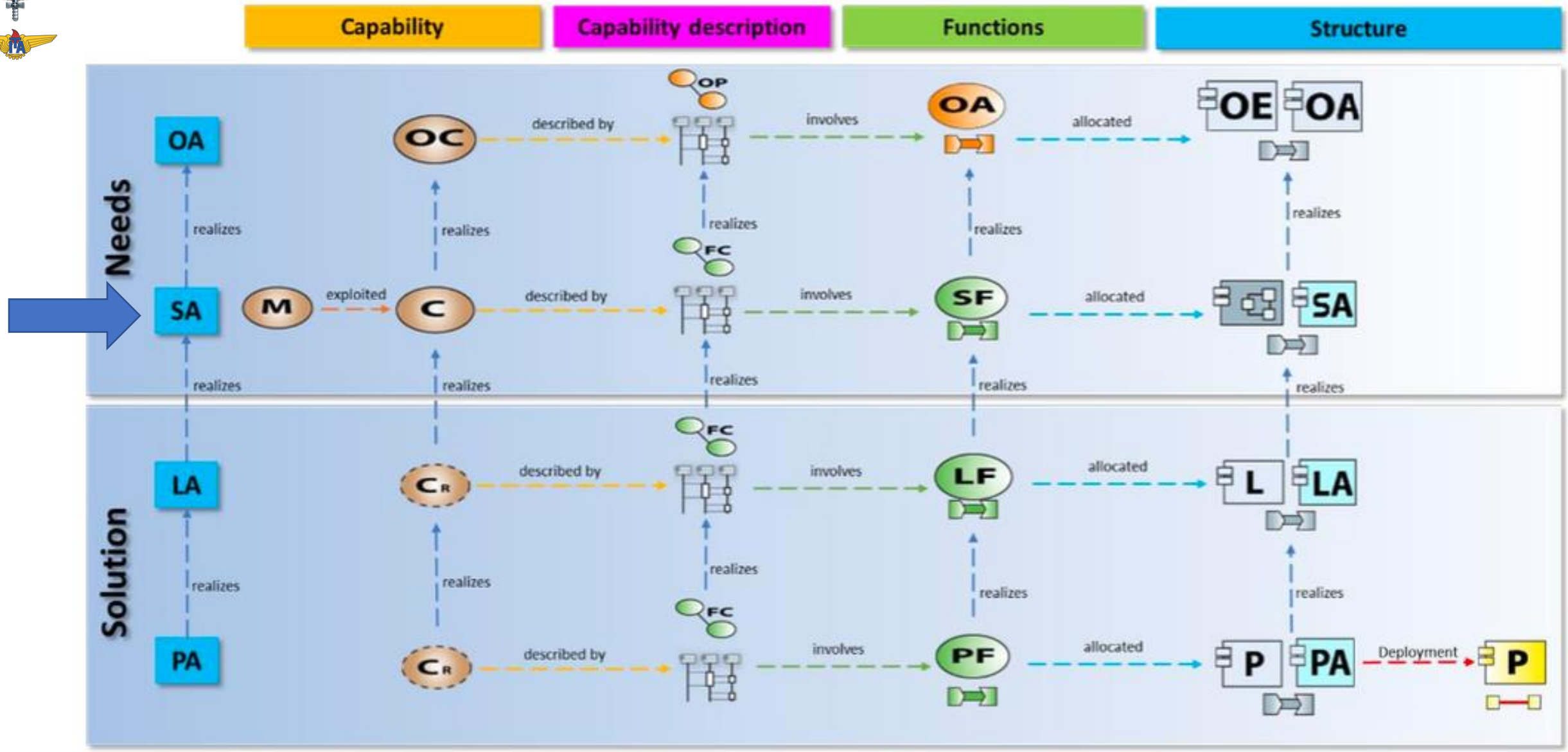


Figure 2.3: Arcadia ontology traceability

https://www.slideshare.net/HelderCastro3/mbse-with-arcadia-methodpdf-256664096

Arcadia layer	Requirements	Capability	Capability description	Functional	Structure	Modes and States	Data	Interfaces
	R-OA	OA1	OA2	OA3	OA4	M&S-OA5	D-OA6	I-OA7
Operational Analysis	Capture stakeholder requirements	Define Operational Capabilities	Define processes and scenarios	Define Operational Activities and interactions	Capture Operational Entities and Actors. Allocate Operational Activities to Operational Actors, Entities	Define operational modes and states	Define operational data model	Define interfaces and describe interfaces scenarios
	R-SA	SA1	SA2	SA3	SA4	M&S-SA5	D-SA6	I-SA7
System Analysis	Derive Stakeholder requirements and capture System requirements	Define System Missions and System Capabilities	Define Functional Chains and Scenarios.	Define System Functions. Define Functional Exchanges and components	Allocate System Functions to System and Actors	Define system modes and states	Define system data model	Define interfaces and describe interfaces scenarios Enrich Logical Scenarios.
	R-LA	LA1	LA2	LA3	LA4	M&S-LA5	D-LA6	I-LA7
Logical Architecture	Derive system requirements and Capture components requirements	Transition Capabilities Realization from system layer	Define Functional Chains and scenarios	Derive System Functions and define Logical Functions. Define Functional Exchanges and components.	Allocate Logical Functions to Logical Components	Define logical components modes and states	Define logical data model	Delegate System Interfaces and create Logical Interfaces. Enrich Logical Scenarios.
	R-PA	PA1	PA2	PA3	PA4	M&S-PA5	D-PA6	I-PA7
Physical Architecture	Derive logical requirements and capture physical requirements	Transition Capabilities Realization from logical layer	Define Functional Chains, Scenarios, and Physical Path	Derive Logical Functions and define Physical Functions. Define Functional Exchanges and components.	Define Physical Nodes and refine Behavioural Physical Components. Allocate Behavioural Components.	Define physical nodes modes and states	Define physical data model	Delegate Logical Interfaces and create Physical Interface. Enrich Physical Scenarios.

Table 3.2: Arcadia matrix activities

https://www.slideshare.net/HelderCastro3/mbse-with-arcadia-methodpdf-256664096

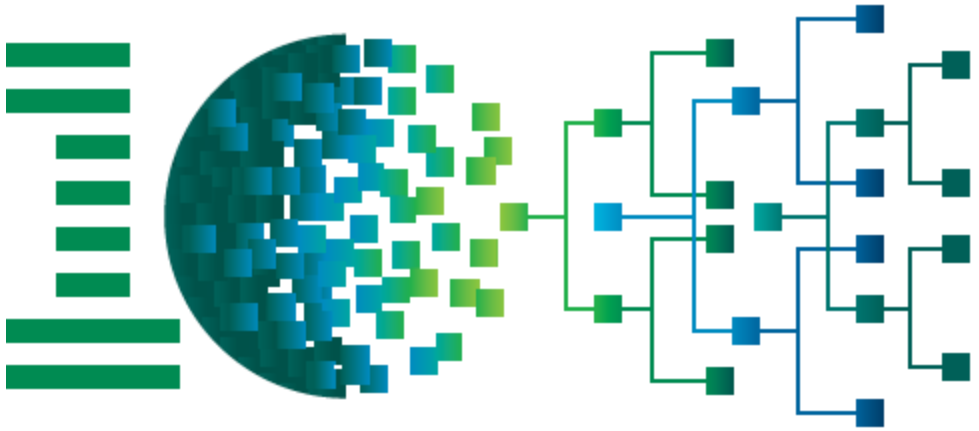
Arcadia layer	Requirements	Capability	Capability description	Functional	Structural	Modes and States	Data	Interfaces
	R-OA	**OA1**	**OA2**	**OA3**	**OA4**	**M&S-OA5**	**D-OA6**	**I-OA7**
Operational Analysis	No dedicated diagram	[OCB] Operational Capabilities	[OAS] Operational Activity Scenario [OPD] Operational Process Scenario [OES] Operational Entity Scenario	[OABD] Operational Activity Breakdown Diagram [OAIB] Operational Activity Interaction Blank	[OEBD] Operational Entities Blank Diagram [ORB] Operational Roles Blank [OAB] Operational Architecture Blank	[MSM] Modes and States	[CDB] Class Diagram	[IDB] Interface Definition Blank [CEI] Component External Interfaces [IS] Interface Scenario [CDI] Component Detailed Interface
	R-SA	**SA1**	**SA2**	**SA3**	**SA4**	**M&S-SA5**	**D-SA6**	**I-SA7**
System Analysis	No dedicated diagram	[MCB] Mission and Capabilities Blank [CC] Contextual Capability	[FS] System Functional Scenario [ES] System Entity Scenario [SFCD] System Functional Chain Description	[SFBD] System Functional Breakdown Diagram [SDFB] System Data Flow Blank	[CSA] Contextual System Actor [SAB] System Architecture Blank	[MSM] Modes and States	[CDB] Class Diagram	[IDB] Interface Definition Blank [CEI] Component External Interfaces [IS] Interface Scenario [CDI] Component Detailed Interface
	R-LA	**LA1**	**LA2**	**LA3**	**LA4**	**M&S-LA5**	**D-LA6**	**I-LA7**
Logical Architecture	No dedicated diagram	[CRB] Capabilities Realization Blank [CRI] Contextual Capability Realization Involvement	[FS] Logical Functional Scenario [ES] Logical Entity Scenario [LFCD] Logical Functional Chain Description	[LFBD] Logical Functional Breakdown Diagram [LDFB] Logical Data Flow Blank	[LCBD] Logical Component Breakdown Diagram [LAB] Logical Architecture Blank	[MSM] Modes and States	[CDB] Class Diagram	[IDB] Interface Definition Blank [CEI] Component External Interfaces [IS] Interface Scenario [CDI] Component Detailed Interface
	R-PA	**PA1**	**PA2**	**PA3**	**PA4**	**M&S-PA5**	**D-PA6**	**I-PA7**
Physical Architecture	No dedicated diagram	[CRB] Capabilities Realization Blank [CRI] Contextual Capability Realization Involvement	[FS] Physical Functional Scenario [ES] Physical Entity Scenario [PFCD] Physical Functional Chain Description	[PFBD] Physical Functional Breakdown Diagram [PDFB] Physical Data Flow Blank	[PCBD] Physical Component Breakdown Diagram [PAB] Physical Architecture Blank	[MSM] Modes and States	[CDB] Class Diagram	[IDB] Interface Definition Blank [CEI] Component External Interfaces [IS] Interface Scenario [CDI] Component Detailed Interface

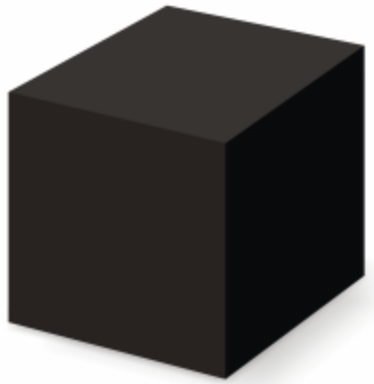
Table 3.3: Arcadia diagrams matrix

https://www.slideshare.net/HelderCastro3/mbse-with-arcadia-methodpdf-256664096

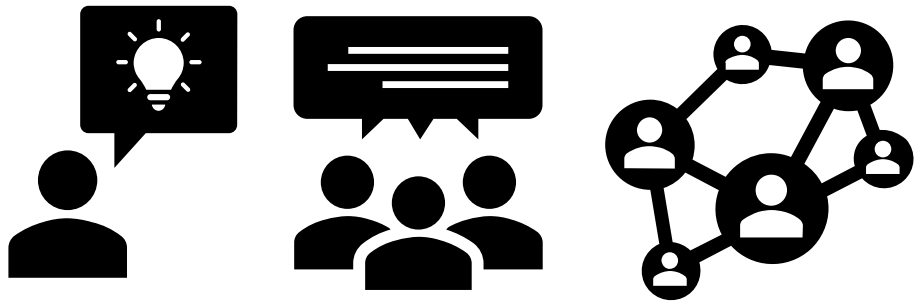
ARCADIA SYSTEMIC CONCEPTS:

- **System**: organized group of elements that function as a **unit (black box)** and **respond to the needs of the users**. The System owns **Component Ports** that allow it to interact with the external Actors;





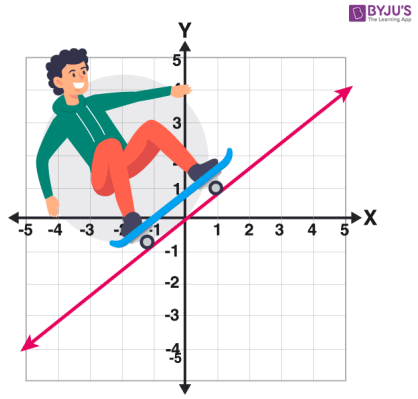
- **Actor**: any element that is **external to the System** (human or nonhuman) that **interacts with it**. (for example Pilot, Test operator, etc.);

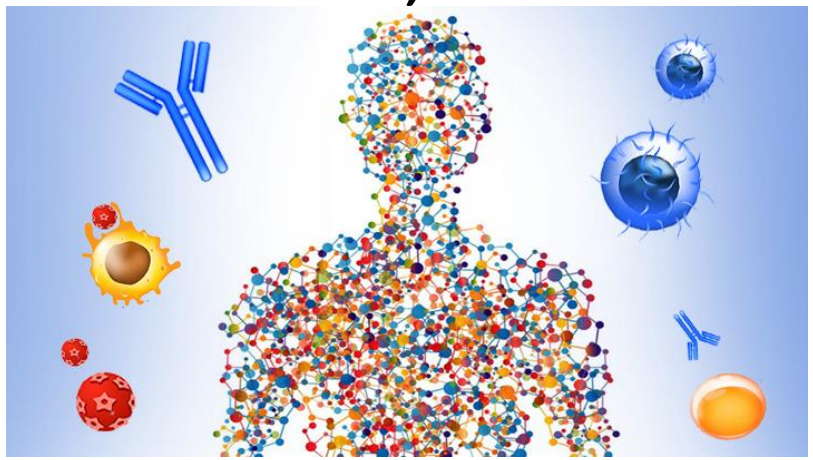


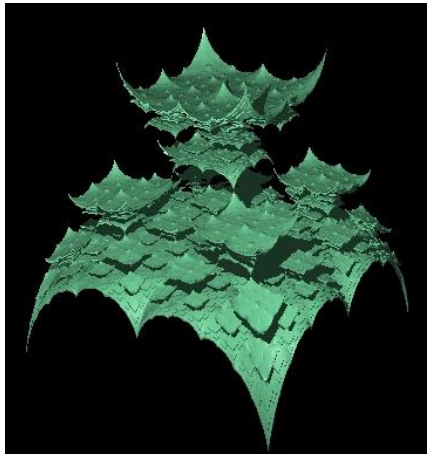
09:21

- **System Capability**: capability of the System to provide a highlevel **service** allowing it **to carry out an operational objective** (*for example provide meteorological data, etc.);*

- **Function**: **behavior** or **service provided** by the System or by an Actor (for example detect a threat, measure altitude, etc.). A Function owns **Function Ports** that allow it to communicate with the other Functions. A Function can be split into subfunctions;

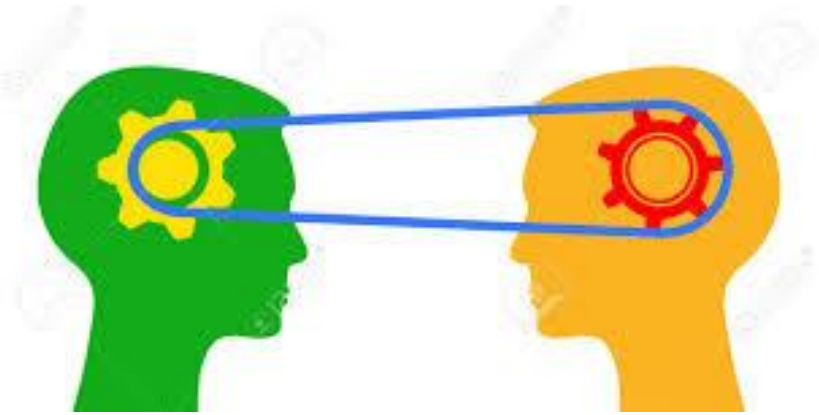


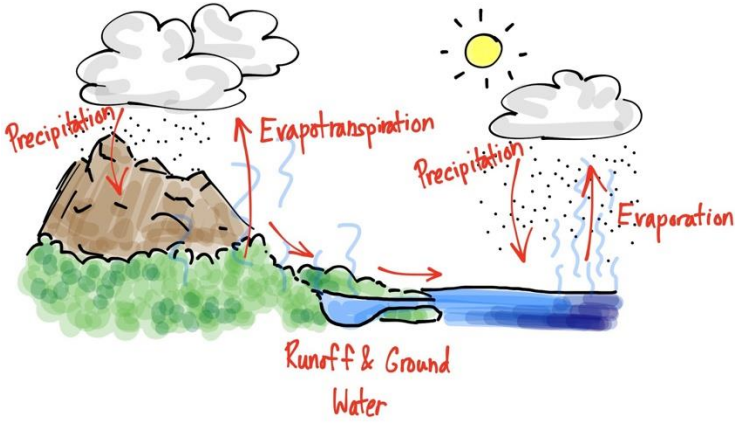




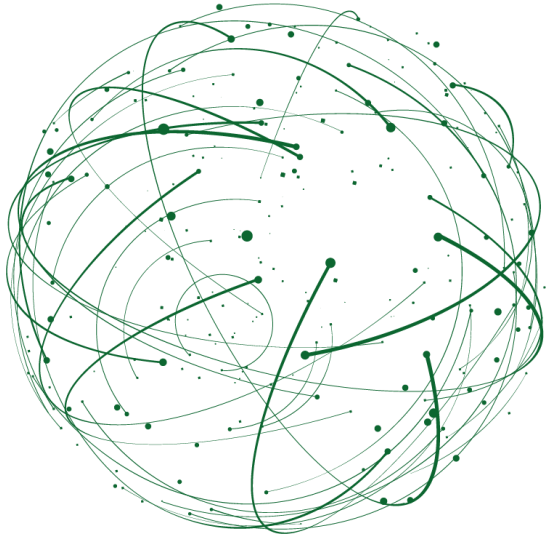
09:21

- **Functional Exchange**: **unidirectional exchange of information or of matter** between two Functions, linking two **Function Ports**;

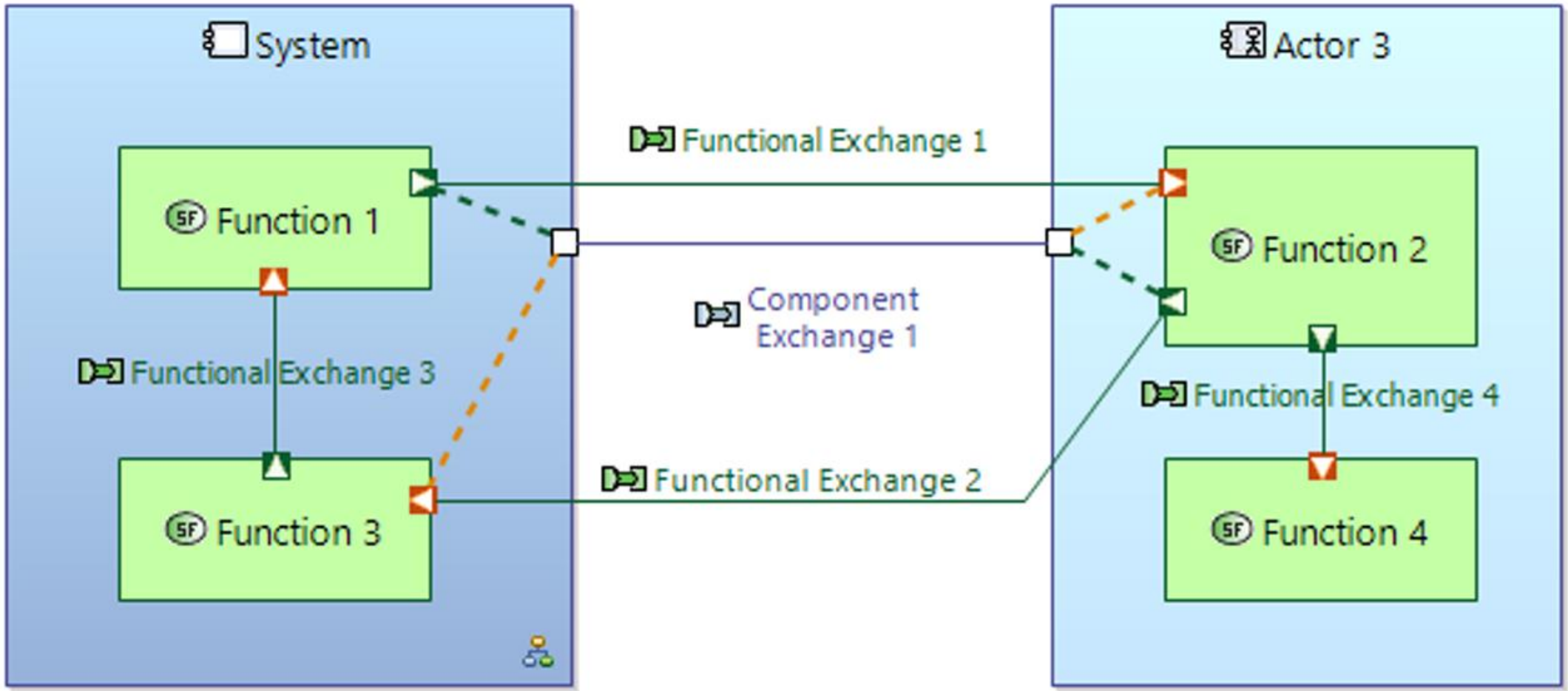




- **Component Exchange**: **connection** between the System and one of its external Actors, allowing **circulation of Functional Exchanges**;

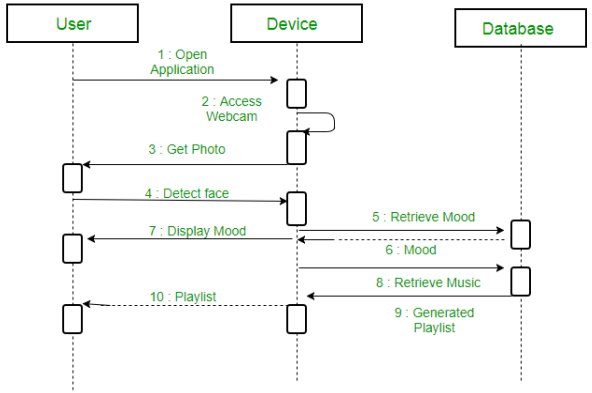


09:21


System
Actor 3
Functional Exchange 1
Function 1
Function 2
Component
Exchange 1
Functional Exchange 3
Functional Exchange 4
Functional Exchange 2
Function 3
Function 4

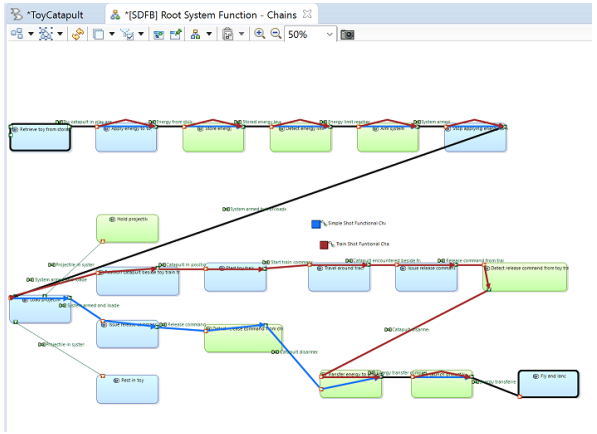
09:21

- **Scenario**: dynamic occurrence describing how the System and its Actors **interact** in the context of a System Capability. It is commonly represented in the form of a sequence diagram, with the vertical axis representing time;



09:21

- **Functional Chain**: element of the model that **enables a specific path to be designated among all possible paths** (using certain Functions and Functional Exchanges). This is particularly useful for assigning constraints (latency, criticality, etc.), as well as organizing tests.



09:21

SYSTEM DIAGRAMS

Transition From Operational Activities

Perform an automated transition of Operational Activities

Create a System Functions / Operational Activities Traceability Matrix

Initialization and automated update of the system analysis according to the breakdown of operational activities.

The initialization and automated updated of the system actors can also be automatically performed from selected operational entities / actors.

The transition tools create a first 1-1 traceability mapping between System Analysis and Operational Analysis. Use dedicated traceability matrices to modify the traceability relationships.

Define Actors, Missions and Capabilities

Perform an automated transition of Operational Capabilities

Contextually create new System Actors from Operational Entities / Actors

Contextually create new System Capability or Mission from Operational Capability

[CSA] Create a new Contextual System Actors diagram

Create a new Mission and / or Capability Blank diagram

Create a System Actors / Operational Entities Traceability Matrix

Identify the boundaries of the system : who which are the actors, which are their goals?

Missions give a global view upon the system main business goals and usages.

Capabilities provide a more operational and finer-grained enlightenment, directly related to customer requirements.

Capabilities are meant to be illustrated with scenarios.

Refine System Functions, describe Functional Exchanges


[SFBD] Create a new Functional Breakdown diagram


[SDFB] Create a new Functional Dataflow Blank diagram

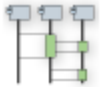

[FS] Create a new Functional Scenario

Enrich and details the functional breakdown with new system functions.

Describe the data flows between system functions and identify specific functional chains.

Allocate System Functions to System and Actors


[SAB] Create a new System Architecture diagram


[ES] Create a new Exchange Scenario

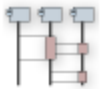
The system and the actors are responsible for implementing the system functions. Manage these allocations using an architecture diagram and deduce component exchanges implementing the functional exchanges.

Create dataflows scenarios to illustrate the functional exchanges between the system and the actors.

Define Interfaces and describe Interface Scenarios


[CDI] Create a new Contextual Detailed Interfaces diagram on the System

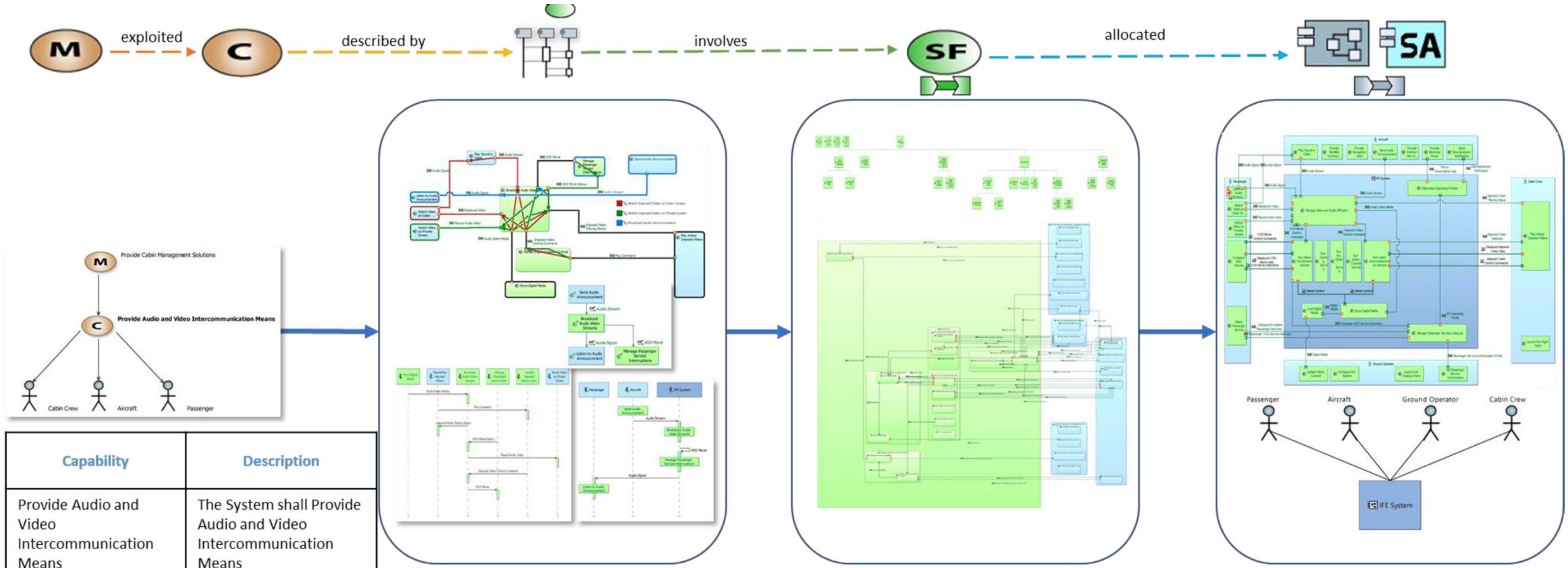

[CEI] Create a new Contextual External Interface diagram on the System


[IS] Create a new Interface Scenario

Detail the interfaces of the system as well as the ones of the actors, thus drawing the boundary of the system.

Describe scenarios in order to specify the dynamical behavior of the system.

Defining the interaction sequences and identifying the interfaces are two very tight and iterative activities



Capability diagrams	Capability description diagrams	Functional diagrams	Structural diagrams
[MCB] Mission and Capabilities Blank [CC] Contextual Capability	[FS] System Functional Scenario [ES] System Entity Scenario [SFCD] System Functional Chain Description	[SFBD] System Functional Breakdown Diagram [SDFB] System Data Flow Blank	[CSA] Contextual System Actor [SAB] System Architecture Blank

Figure 23: System Analysis model elements and diagrams traceability flow

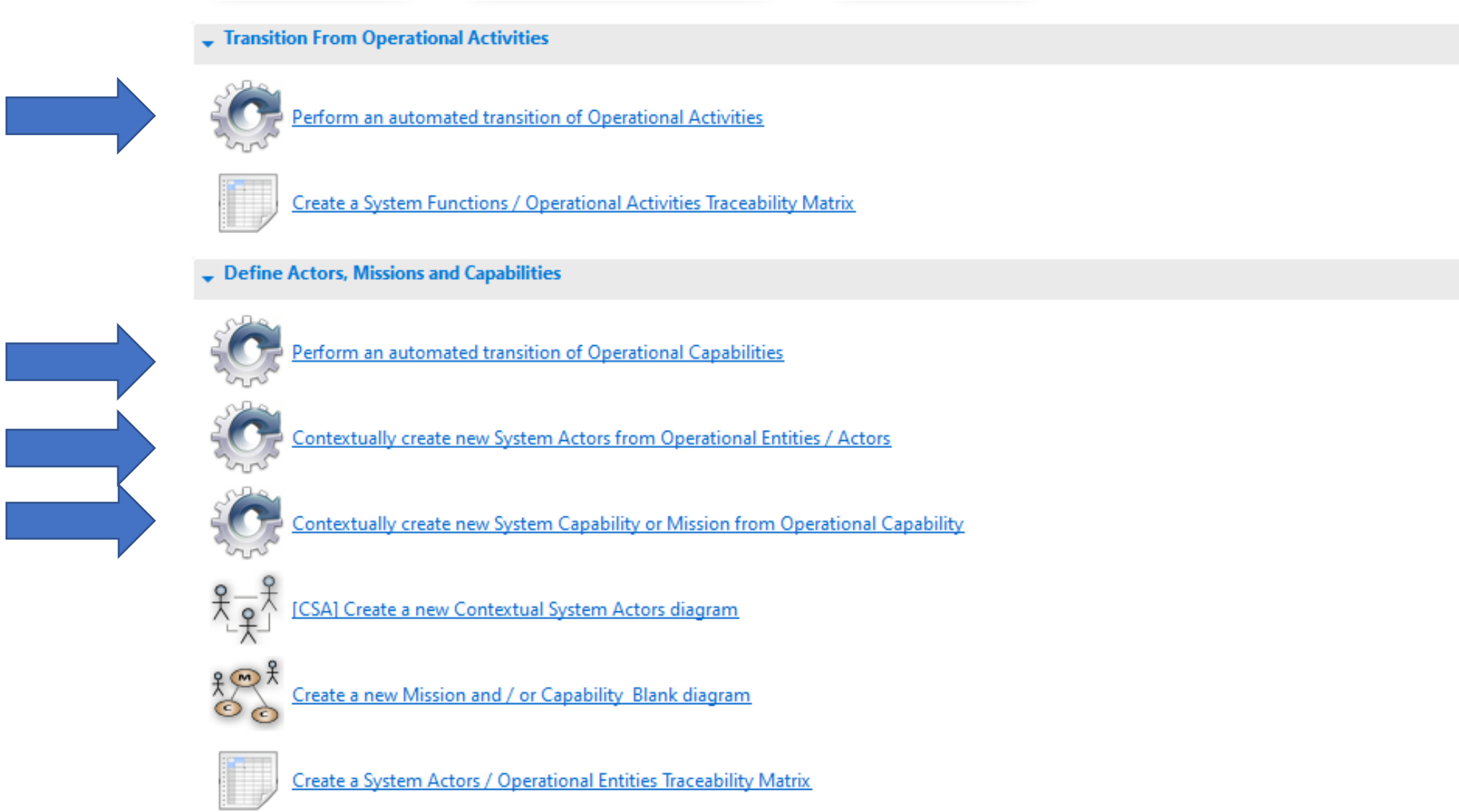
sA STEPS EXAMPLE

09:21

“Steps” disclaimer

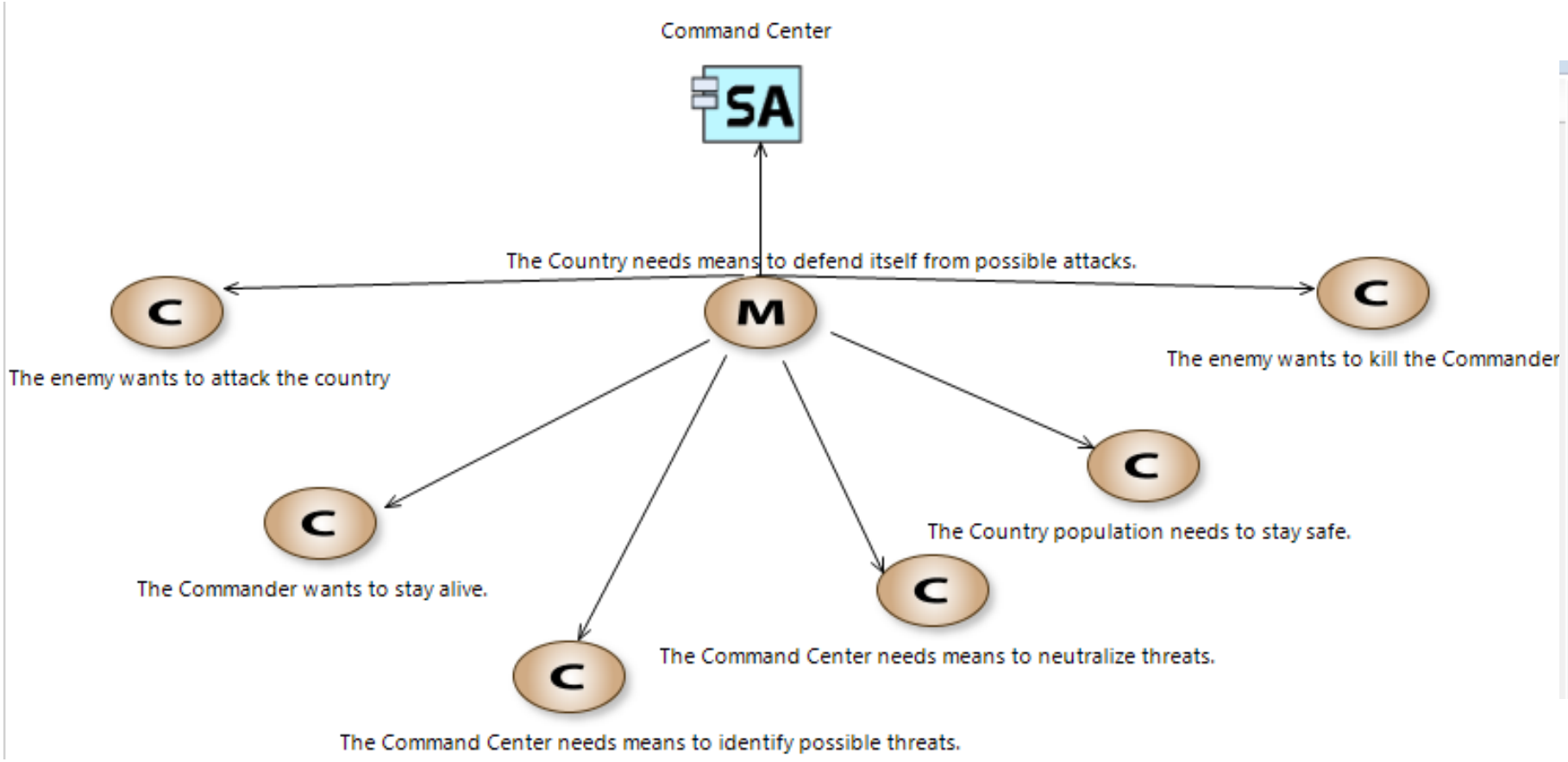
- These steps **are just to be didact**, and incrementally describe each diagram of this level.
- You can follow any sequence you might want (or be defined by your organization) to describe the model.
- I personally, in daily activities, go straight to the architecture diagram (SAB) and create the others as needed.
- It is important that you **iterate as much as you need/can**, to cover this level to understand the problem space.
- Beyond what Arcadia Method describes, you will have to use other approaches to collect/analyze data.

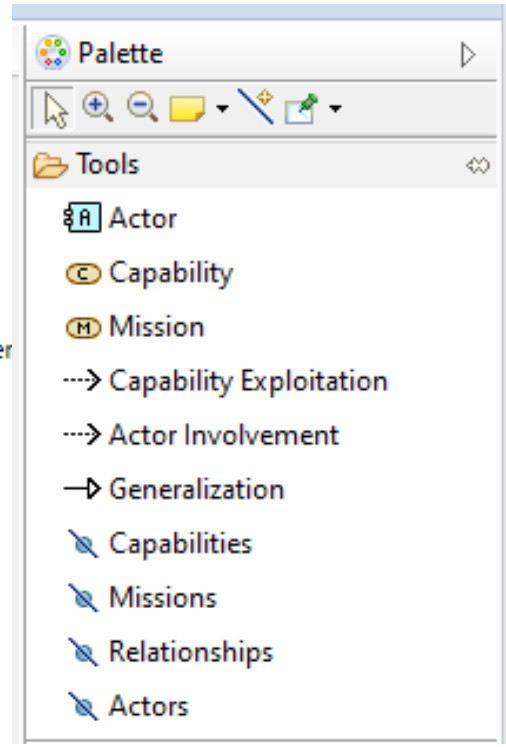
IMPORT DECISION FROM OA



09:21

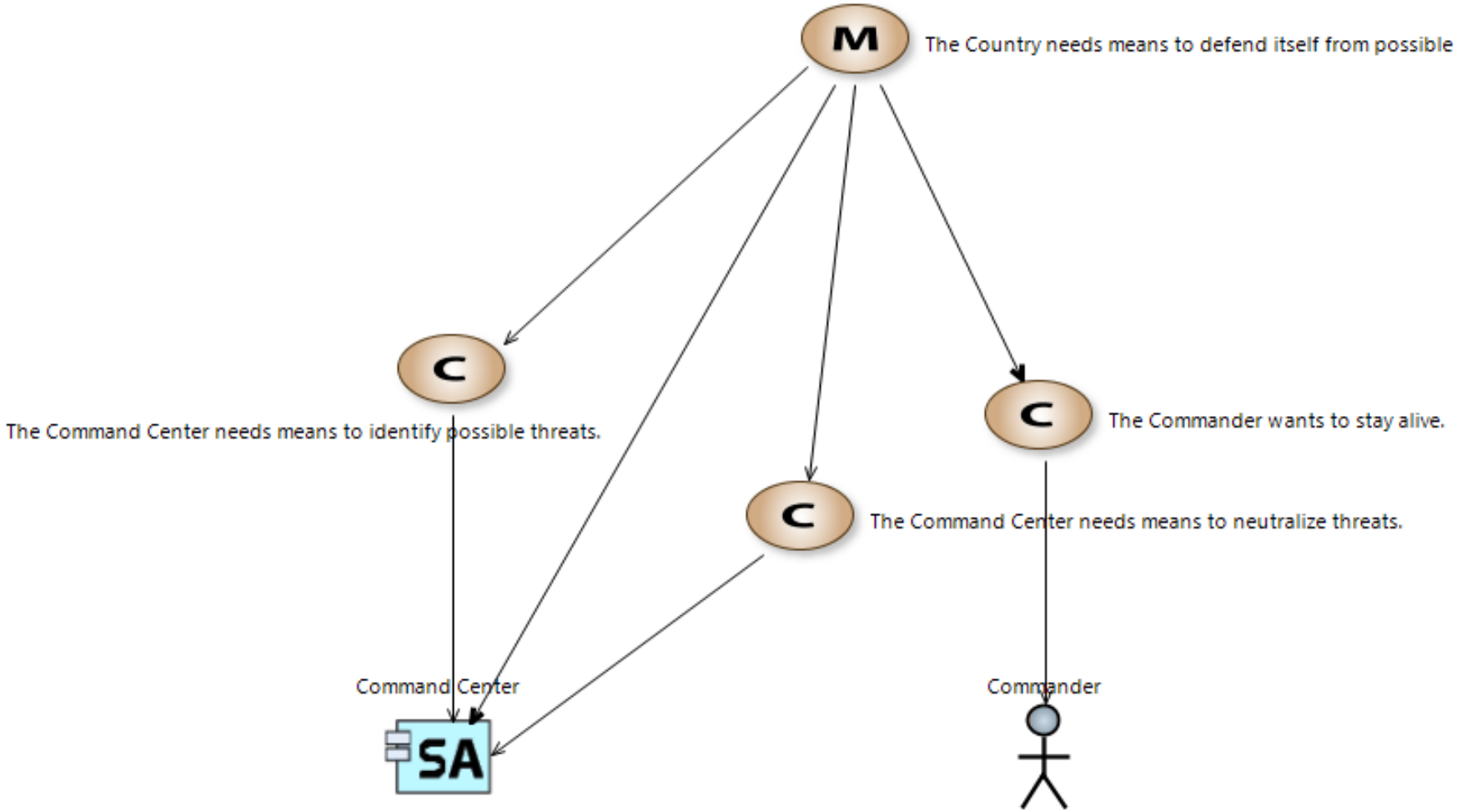
[MB] Mission (identify the mission related to the capability and the actors)

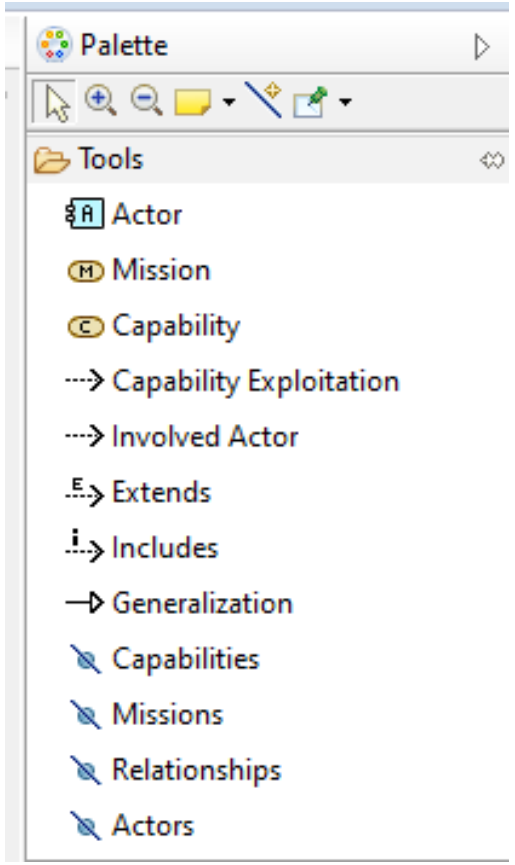




09:21

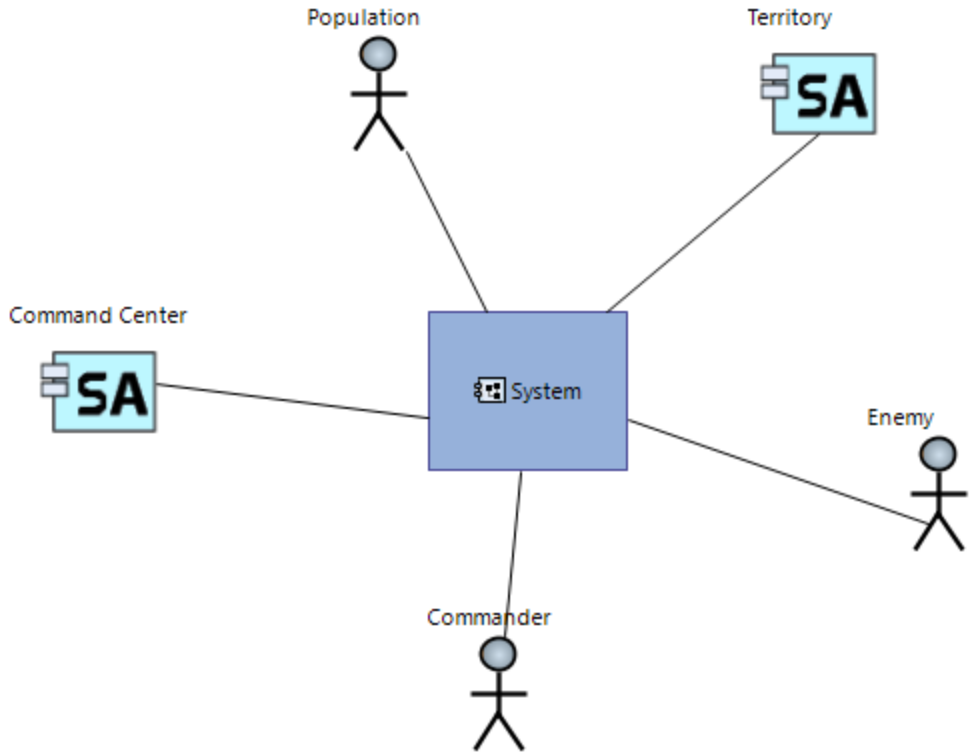
[MBC] Mission Capabilities

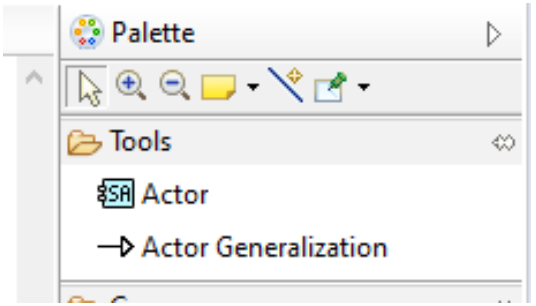




09:21

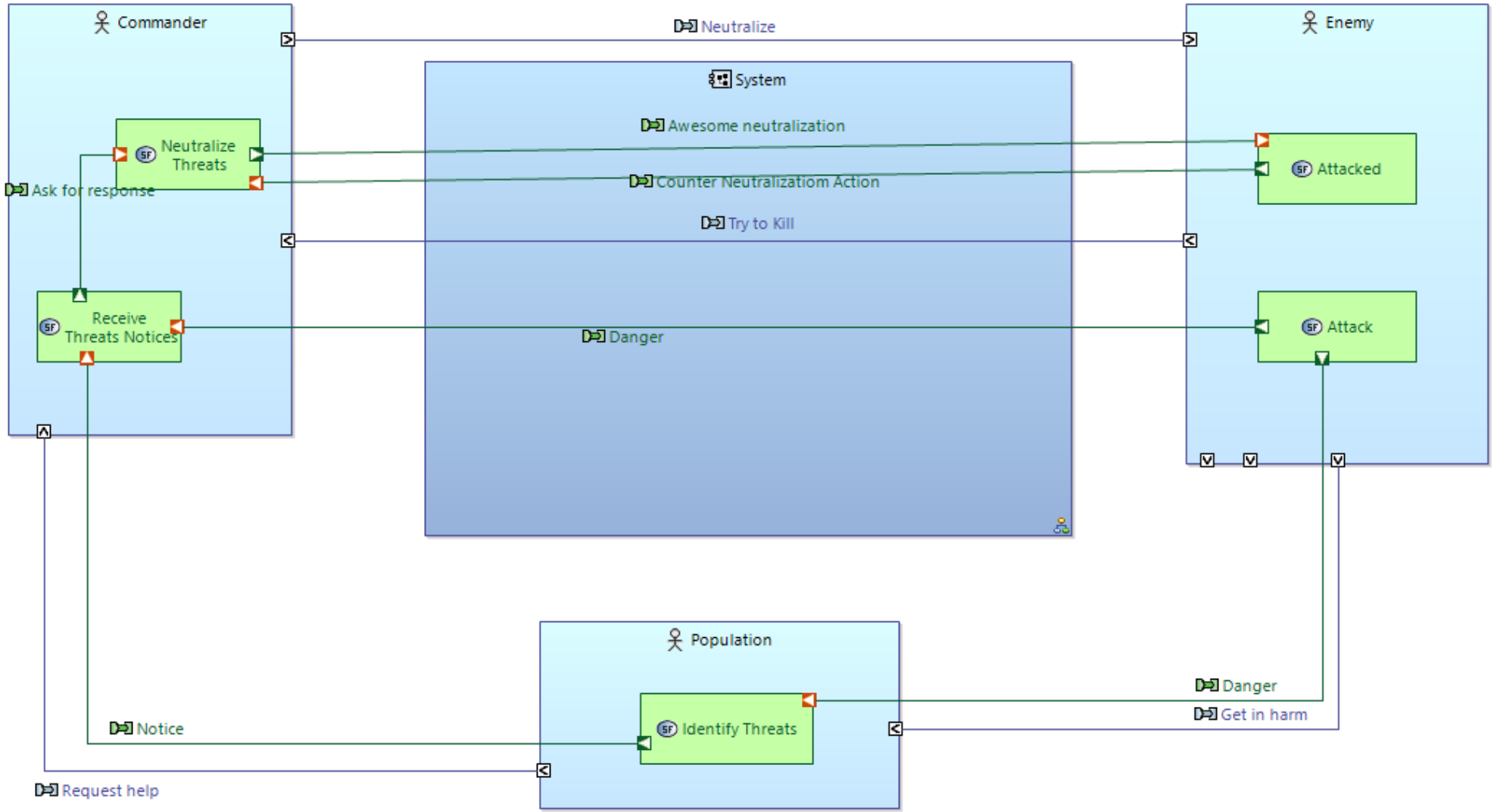
[CSA] Contextual System Actors

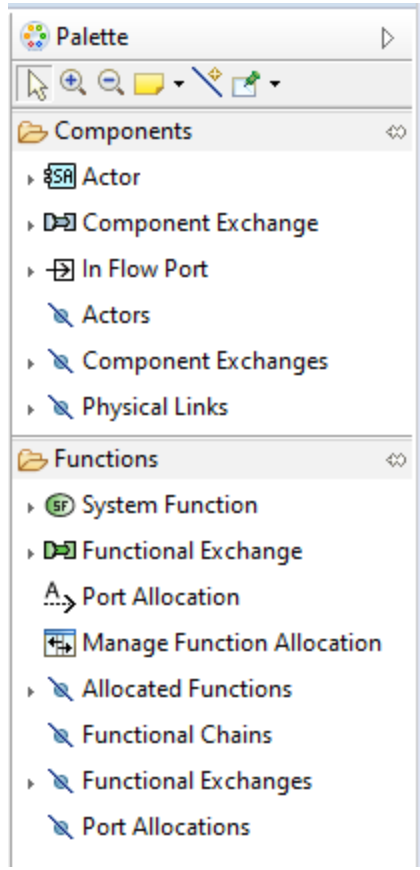




09:21

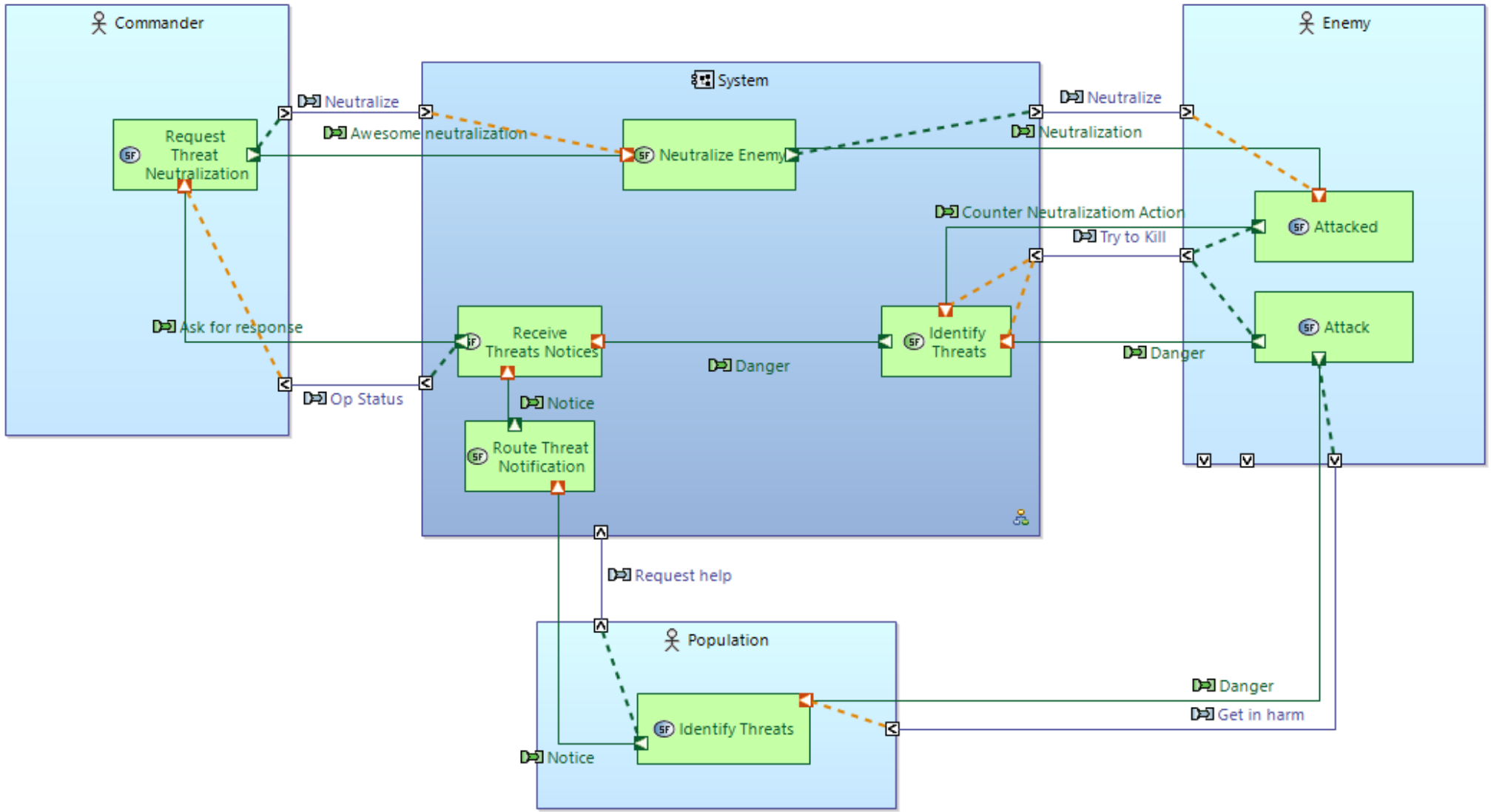
[SAB] System Architecture <START>





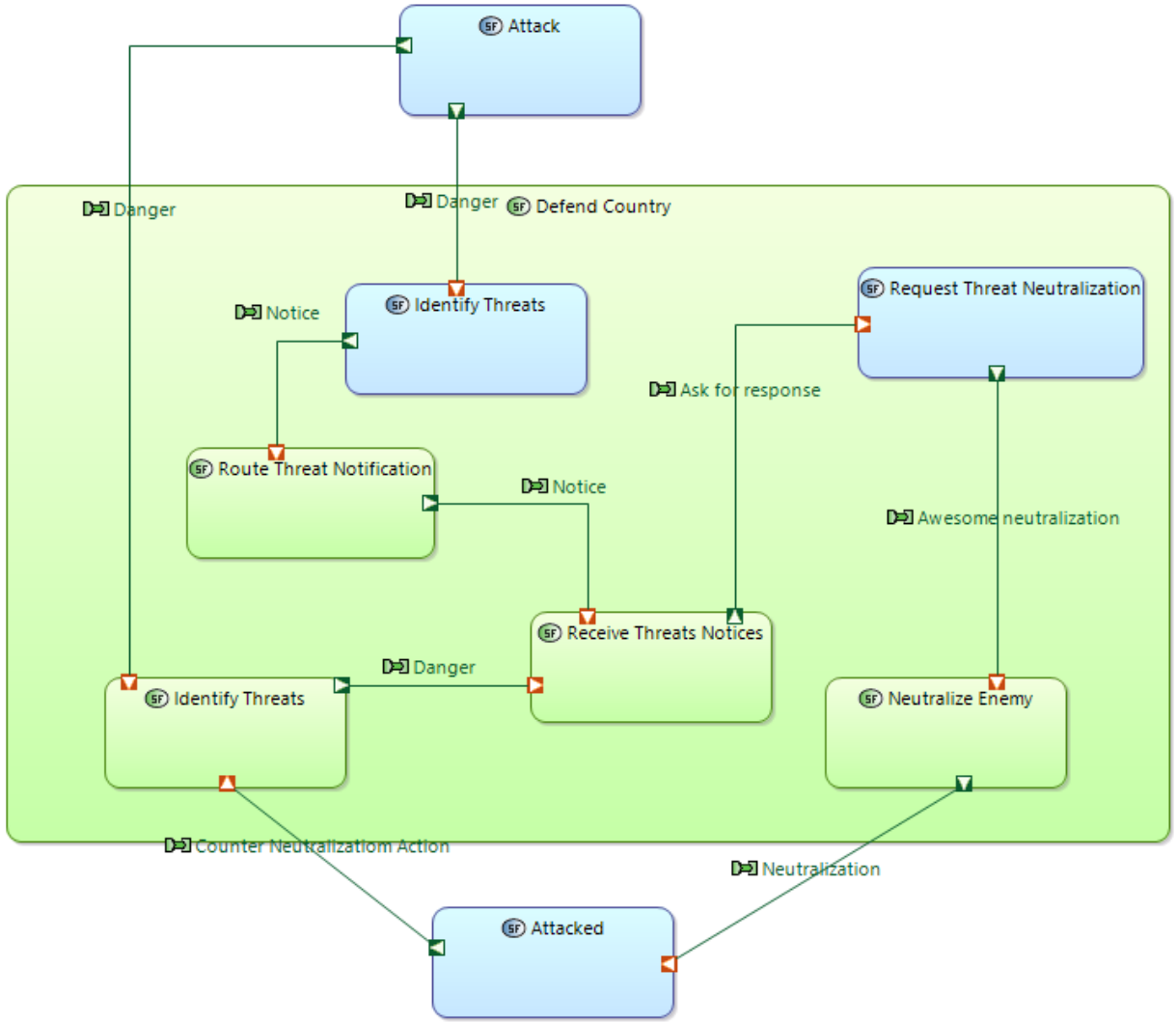
09:21

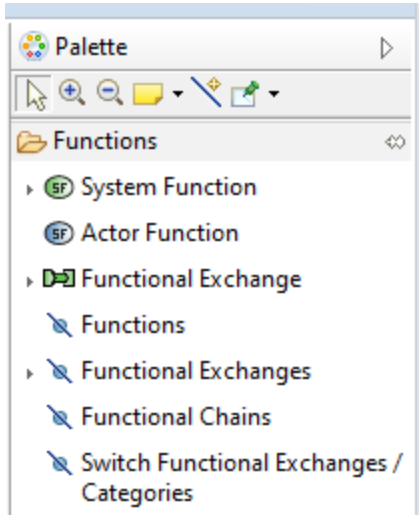
[SAB] System Architecture <END>



09:21

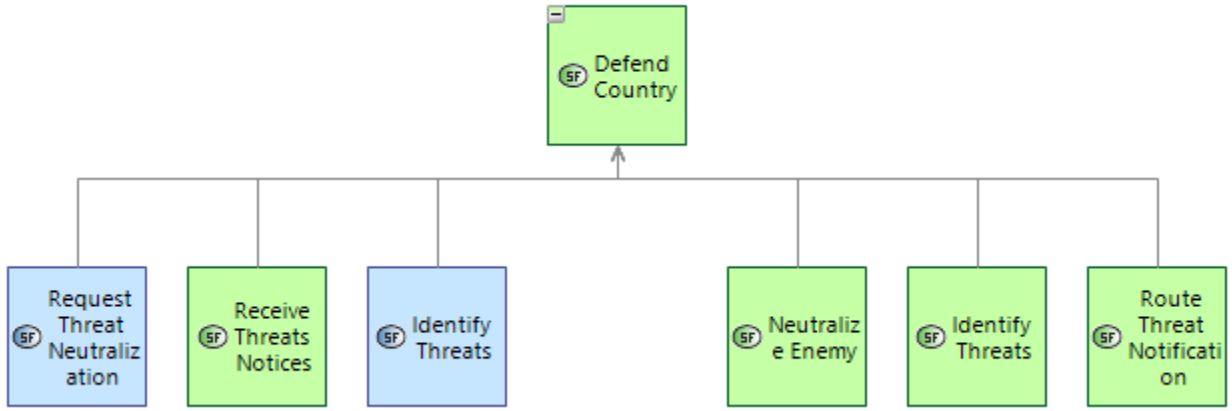
[SDFB] System Dataflow

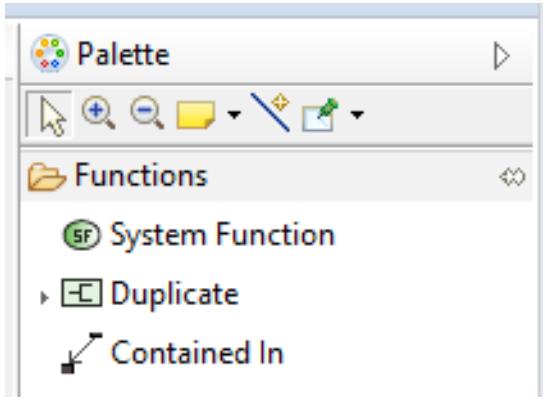




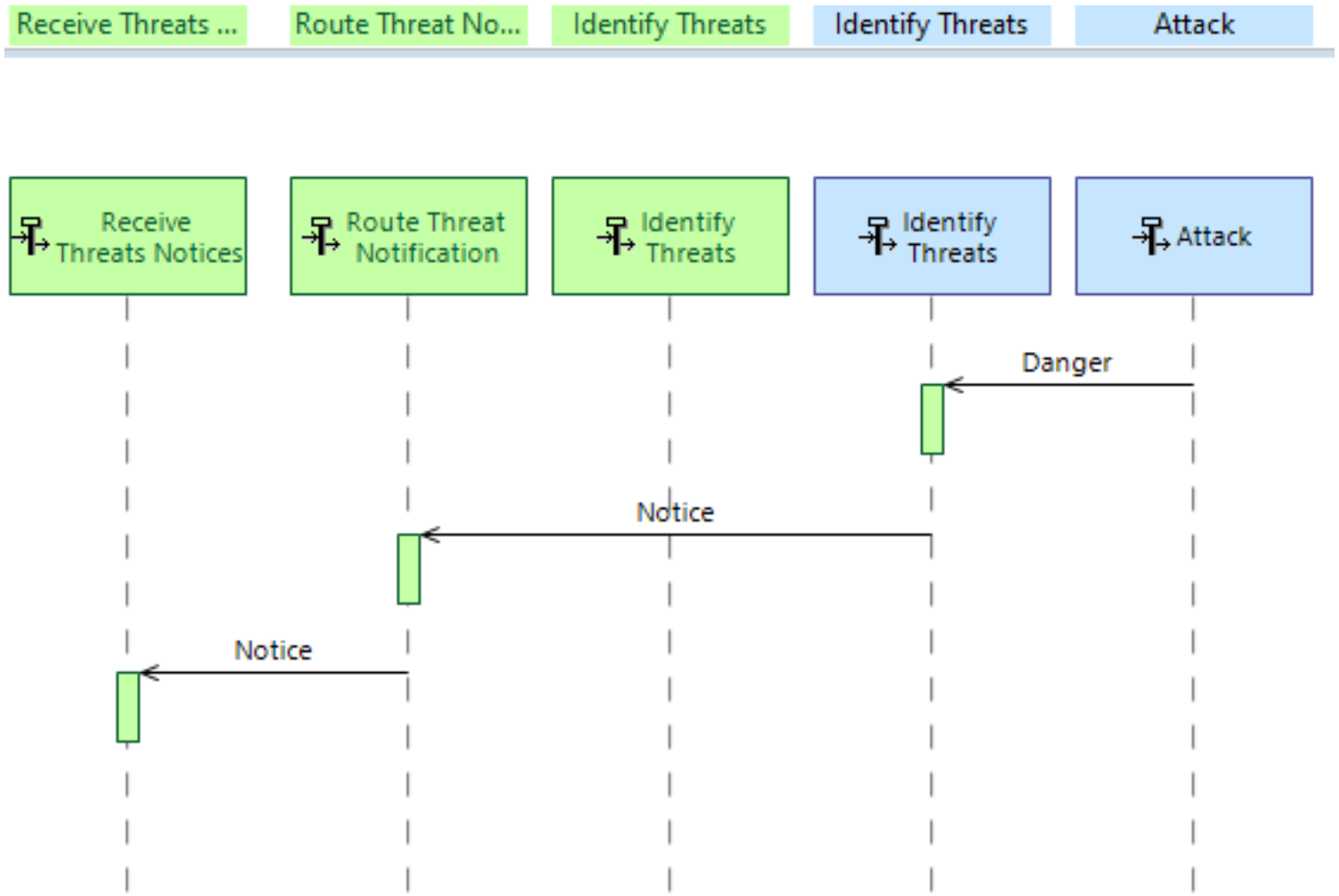
09:21

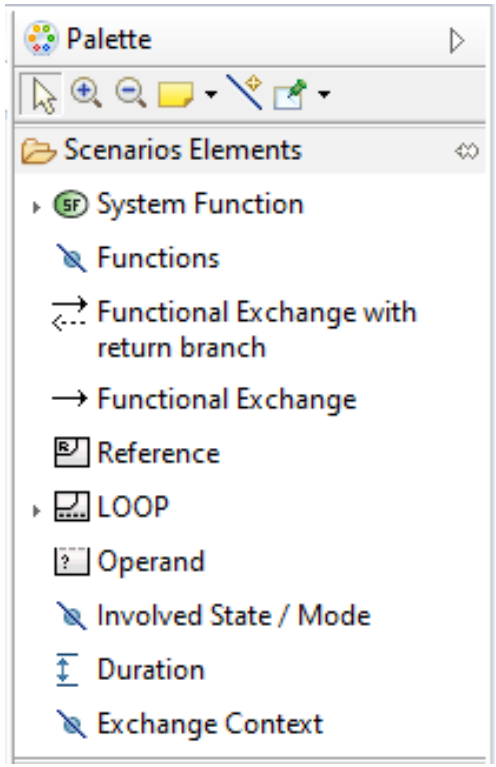
[SFBD] System Functional Breakdown





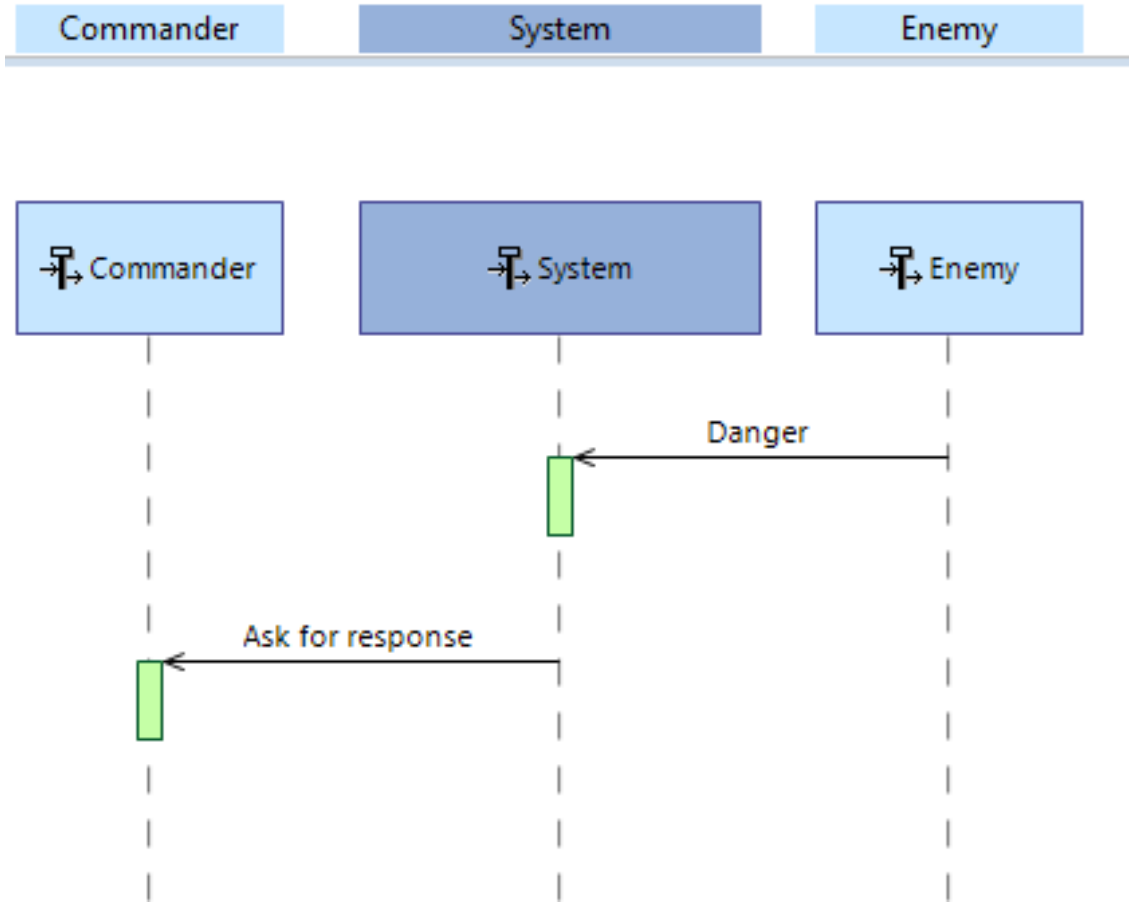
[FS] Functional Scenario

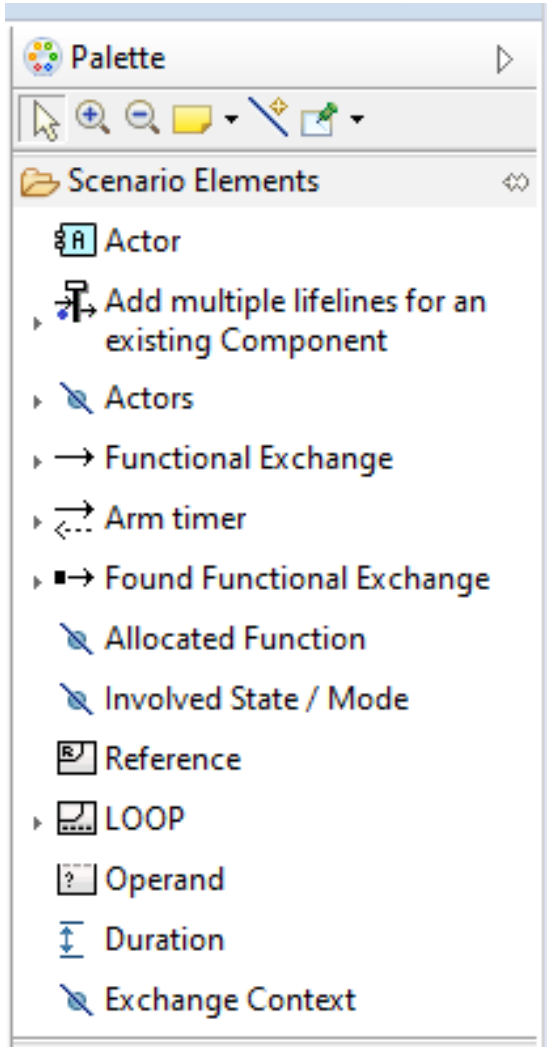




09:21

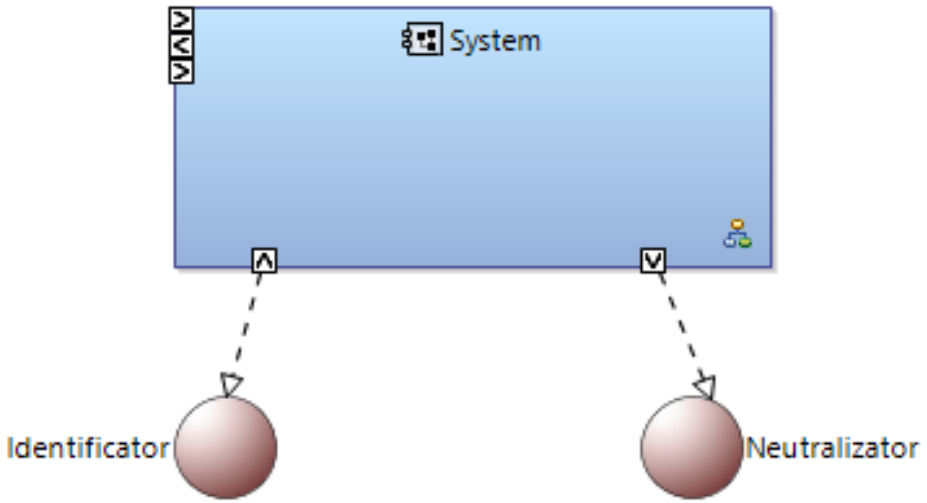
[ES] Exchange Scenario

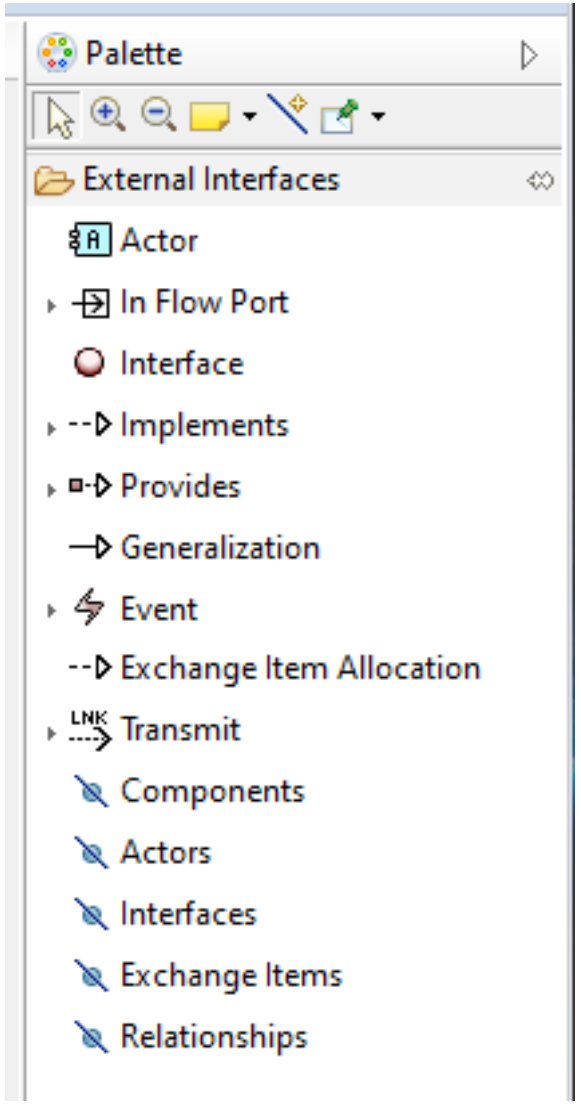




09:21

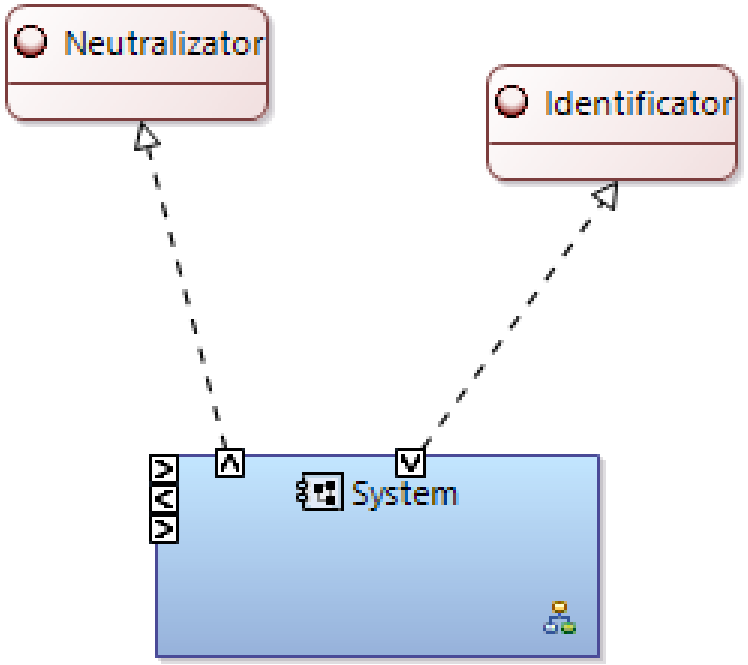
[CEI] Contextual External Interface

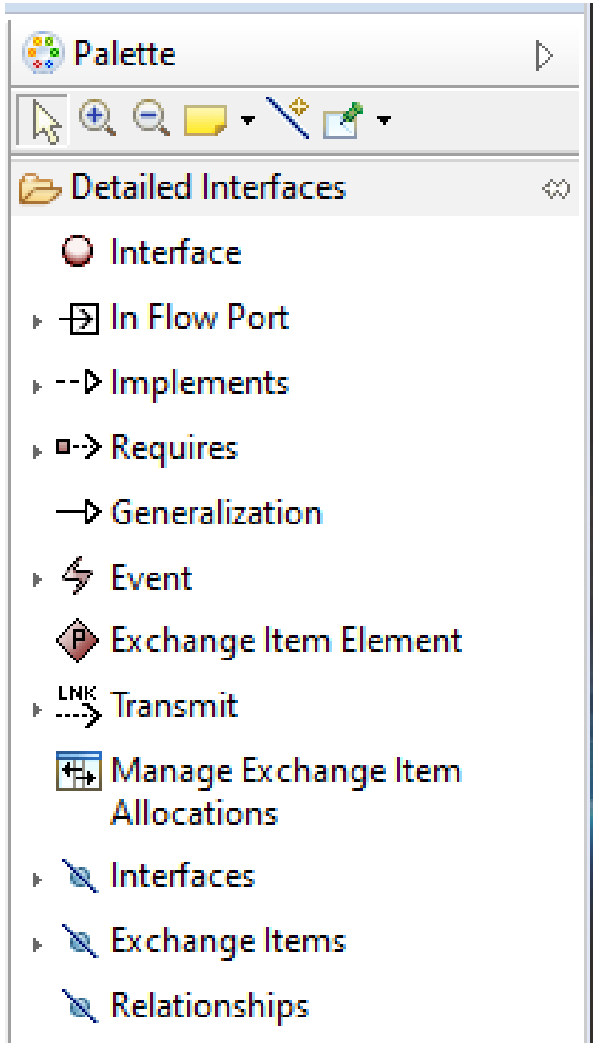




09:21

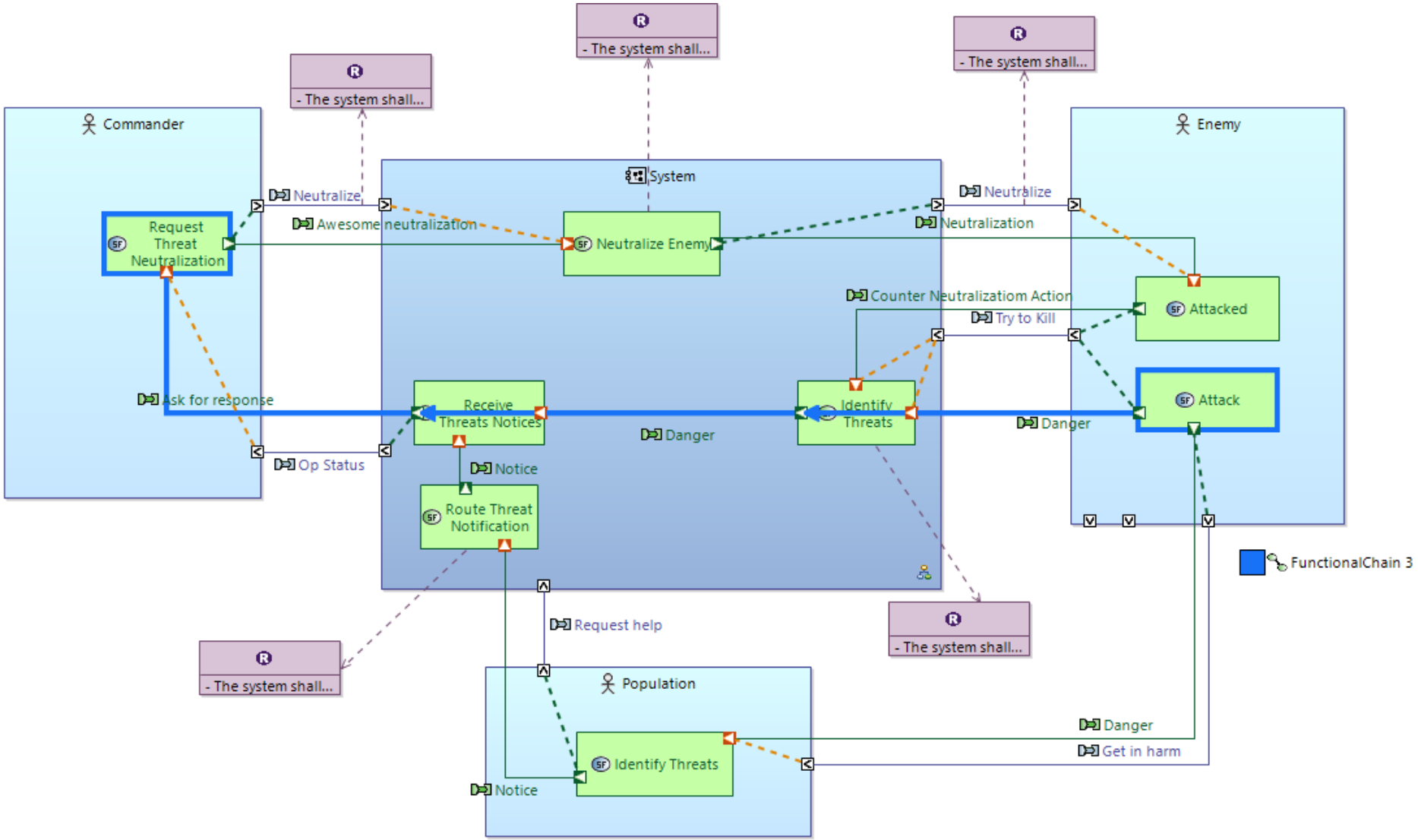
[CDI] Contextual Detailed Interface





09:21

[SAB] System Architecture with Requirements



09:21

Final Considerations

Final considerations

- System Analysis answers:
  - ***What the system has to accomplish for the users***
- The system is **only a black box** that exposes the interface functions.
- Focus on consolidate the needs that are accomplished by the system ➔ emergence
- Maps the system main functions that answers to the stakeholder needs.

Atividades para a próxima aula

- Fazer a etapa da Intervenção
- Apresentar o que o sistema tem que realizar para atender à demanda.
- Apresentar o modelo da Análise do Sistema
  - Características mínimas: manter apenas 2 stakeholders (justificar priorização), propor a missão do sistema e derivar 2 capacidades para o sistema, dizer que o sistema terá 2 funções, decomposição (detalhamento) de 1a dessas funções em 4 sub-funções, 1 diagrama de árvore funcional, 1 diagrama do arranjo interno dessa função decomposta, explicar a rastreabilidade entre as Atividades Operacionais e as Funções do Sistema (o que essas funções resolvem no problema), criar uma máquina de estado do sistema (3 estados), 1 diagrama de arquitetura, 1 diagrama de interface, descreva a troca que passa por uma interface no diagrama de classes, proponha 4 reqs e rastreie com as funções e propriedades desejadas, construa uma cadeia funcional para fazer o verificação de um requisito, e descreva com o diagrama de sequência os eventos ideiais que o sistema deve realizar.