



IEA-S – DEPARTAMENTO DE SISTEMAS ESPACIAIS
(SPACE SYSTEMS DEPARTMENT)

ARQUITETURA CONCEITUAL

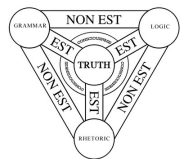


[TE-265 ENGENHARIA DE SISTEMAS BASEADA EM MODELOS] [2023]

AULA	TEORIA	INDIVIDUAL	GRUPO	AULA	TEORIA	GRUPO
1 <i>06-Mar</i>	Introdução e Apresentação de Engenharia de Sistemas	Resumo dos princípios	Definição do grupo. Montagem de apresentação do tema.	9 <i>08-May</i>	Introdução ao Arcadia e Análise do Contexto	Análise do Contexto e apontamento de necessidades
2 <i>13-Mar</i>	Pitch: Temas Frameworks e Stakeholders	Trabalho sobre MBSE	Elicitar stakeholders	10 <i>15-May</i>	Pitch: Análise do Contexto - NOP Intervenção Sistemica	Intervenção Sistemica e requisitos da missão do sistema
3 <i>20-Mar</i>	Arquitetura e Funções. Coesão e acoplamento.	Exercícios de análise estruturada	Mapa de interações	11 <i>22-May</i>	Pitch: Análise do Sistema - ROP OPM e Exploração de Alternativas	Montagem de Alternativas
4 <i>27-Mar</i>	Ciclo de Vida e CONOPs		Ciclo de Vida e CONOPs	12 <i>29-May</i>	Pitch: Alternativas Arquitetura Conceitual e desdobramentos	Arquitetura Conceitual e requisitos do sistema
5 <i>03-Apr</i>	Pitch: Descrição livre da captura do problema Requisitos	Exercícios de correção de requisitos	Requisitos dos stakeholders	13 <i>05-Jun</i>	Pitch: Arquitetura Conceitual - RTLI Revisita de Requisitos e o processo de Verificação e Validação	Apontamento de etividades de verificação
6 <i>10-Apr</i>	Modelagem Estrutural da Arquitetura	Exercícios de fixação	Relatório e Gravação de 5min com explicação	14 <i>12-Jun</i>	Pitch: Propostas de Verificação Arquitetura Concreta e Carta Morfológica	Arquitetura Concreta com decisões tecnologicas
7 <i>17-Apr</i>	Modelagem de Comportamento da Arquitetura	Exercícios de fixação		15 <i>19-Jun</i>	Pitch: Arquitetura Final - Especificações Revisão e desdobramentos para especialidades	Relatório e Gravação de 5min com explicação.
8 P1 - Questões conceituais e Mini-Case <i>24-Apr</i>				16 P2 - Apresentação final da relação entre as etapas <i>26-Jun</i>		
				EXAME Graduação (grupo): Desenvolvimento de um mini-case de um subsistema usando o Capella		
				<i>03-Jul</i>	Pós-graduação (grupo): Entrega de um artigo (Formato do SIGE) descrevendo seu case e atividades.	
				<i>17-Jul</i>		



CONCEPTIO FRAMEWORK



DCA-400-6

Problem



Capella
Open Source MBSE Solution



Soft
Operational
Research

Problem
Structuring
Methods

Context Analysis



Problem Framing

Operational
Analysis

Problem
Simulation

SIL
PIL

Hard
Operational
Research

Formalized Needs

Systemic Intervention

Seed

System Analysis

Concept
Simulation

SIL
PIL

System Element Requirements

Conceptual Alternatives



Ideation

Solution Neutral
Architecture

Logical Architecture

Concept
Simulation

SIL
PIL

Flowed-down Requirements

Realized Architecture

Design Architecture

Physical
Architecture

Product
Simulation

SIL
PIL
HIL

Specific
Discipline
Analysis

Multilevel and Multiphase

Multi Criteria
Decision
Analysis

Solution

PNOP

NOP

ROP

RTLI



DOMÍNIO DA SOLUÇÃO



T2 – CONSIDERE TODO O PROBLEMA, TODA A SOLUÇÃO E TODO O CICLO DE VIDA

- A Engenharia de Sistemas se preocupa com todo o problema e toda a solução, incluindo **como o "sistema de intervenção" irá interagir com seu ambiente** como parte de um sistema maior quando for implantado, e todos os sistemas e serviços de suporte necessários para estabelecer e manter a eficácia do sistema durante todo o seu ciclo de vida até uma eventual eliminação satisfatória.
- Precisamos considerar **todo o ciclo de vida de toda a solução, incluindo todos os sistemas de suporte (enabling systems) que acompanham o sistema de interesse**



- Visualizando o **problema como um sistema**,
- compreender como as **interdependências entre os elementos no espaço do problema criam os "sintomas do problema" e como o "sistema de intervenção"** pode aliviar os sintomas do problema
- **compreender as interações e interdependências** das partes interessadas e **estabelecer o propósito e os critérios de sucesso** acordados
- **antecipar e minimizar potenciais consequências** adversas ou não intencionais do sistema de intervenção
- **varredura e detecção precoce de comportamento anômalo** e consequências não intencionais – nem tudo pode ser antecipado de antemão



SOLUTION SPACE

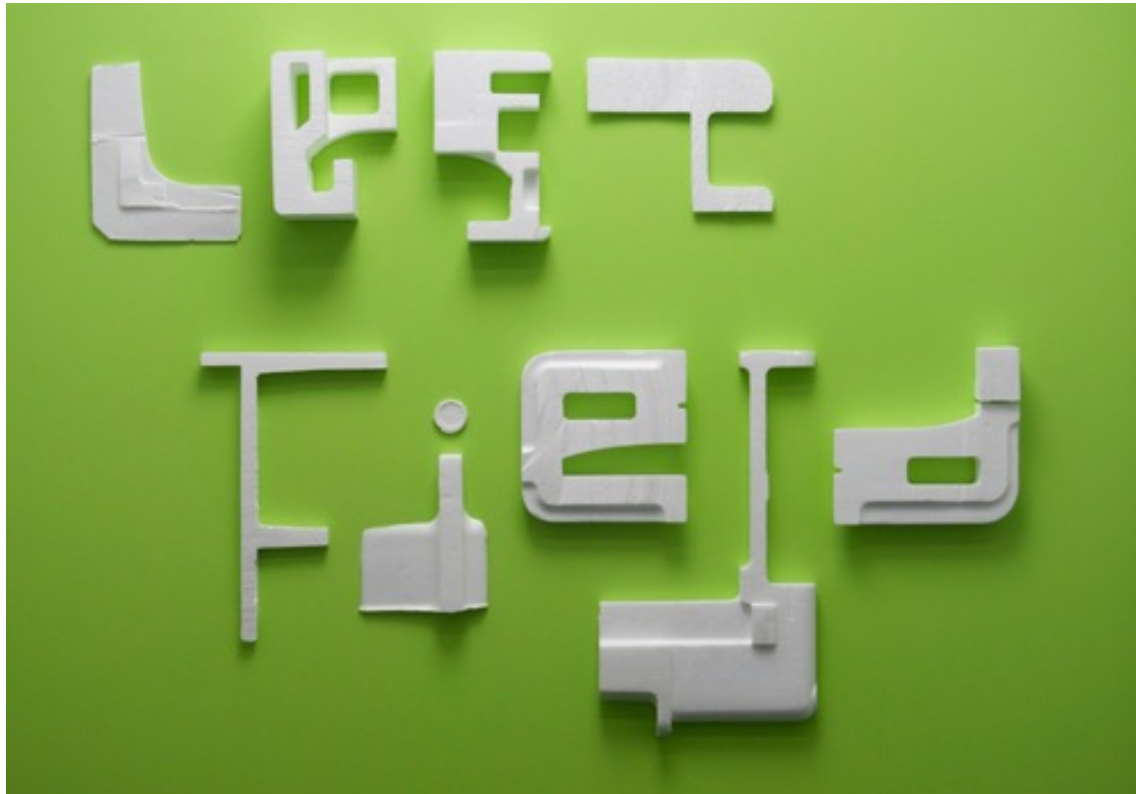
- No espaço da solução, a abordagem SE envolve
- Identificar **abordagens de solução** potenciais,
- selecionar uma **abordagem adequada** com base em evidências e julgamento de especialistas, guiada pelo propósito e **levando em conta os níveis de risco, incerteza e mudança**;
- **definir a solução, as partes componentes e suas propriedades**, e os produtos e serviços necessários para projetar, desenvolver, testar, implantar, usar, avaliar, dar suporte, evoluir e, eventualmente, aposentar e descartar o sistemas

Table 4. Applying SE Tools to the Solution Design

Tool	Solution Design Application
Understand and define the problem	Establish a clear, agreed definition and scope criteria for the innovation system. Understand how every part of the solution contributes to those the purpose.
Whole systems, which spans whole lifecycle	The solution must be a product, process, service or experience. Consider all aspects of the solution including making systems and services required for a successful system that is fit for purpose.
Understand and manage interdependencies	Systems Engineering focuses not on the design of the parts, but on the required properties of the parts and how they need to interact to create the required properties, capabilities, behavior and outcomes of the whole solution. Understand how key attributes of the system must be shaped the "System Building" of conceptual design, analysis and design synthesis.
Adapt the parts to serve the purpose of the whole	Parts must exist to be sub-optimal, and all the sub-parts interdependent, so that they work harmoniously to support the system.
System Engineering at multiple levels	Several layers of the solution must have parts that are worth making as systems in their own right.
How decisions are made and managed	How a system is designed based on evidence, guided by purpose, and taking into account the levels of risk, uncertainty and change. Take decisions at the "best possible moment" using rational judgement. If evidence is not available or difficult, then use judgement.
Uncertainty, change risk, opportunities and expectations	Keep system open until at least one is definitely suitable. Acknowledge when there is no solution, when the solution is not feasible and when the solution is not the best. Keep changes, things that are adaptable and evolutionary learning, through participants' experiences to learn.
Structure and coherence	Apply equal attention to behavioral/operational uses of the system as to the physical and technical uses. Develop both logical and physical architectures.
Feedback	Understand and carefully manage the feedback loop in the solution and in the innovation process. The system must be designed to allow the system to evolve. When the system is designed, the system must be designed to allow the system to evolve.
Value	Focus on measuring value. When multiple measures are used, make sure the value proposition of the solution remains compelling.
Responsible and sustainable	Keep standards high for the system. "Are we building the right system, with the right the problem?" and the final "will it work as expected in all relevant situations?"
Respect the people	Consider the implications of different solution configurations and supporting the system. Consider the cultural uses of the system as well as the technical uses. Consider the social, cultural and economic uses of the system. Consider the social, cultural and economic uses of the system. Consider the social, cultural and economic uses of the system.



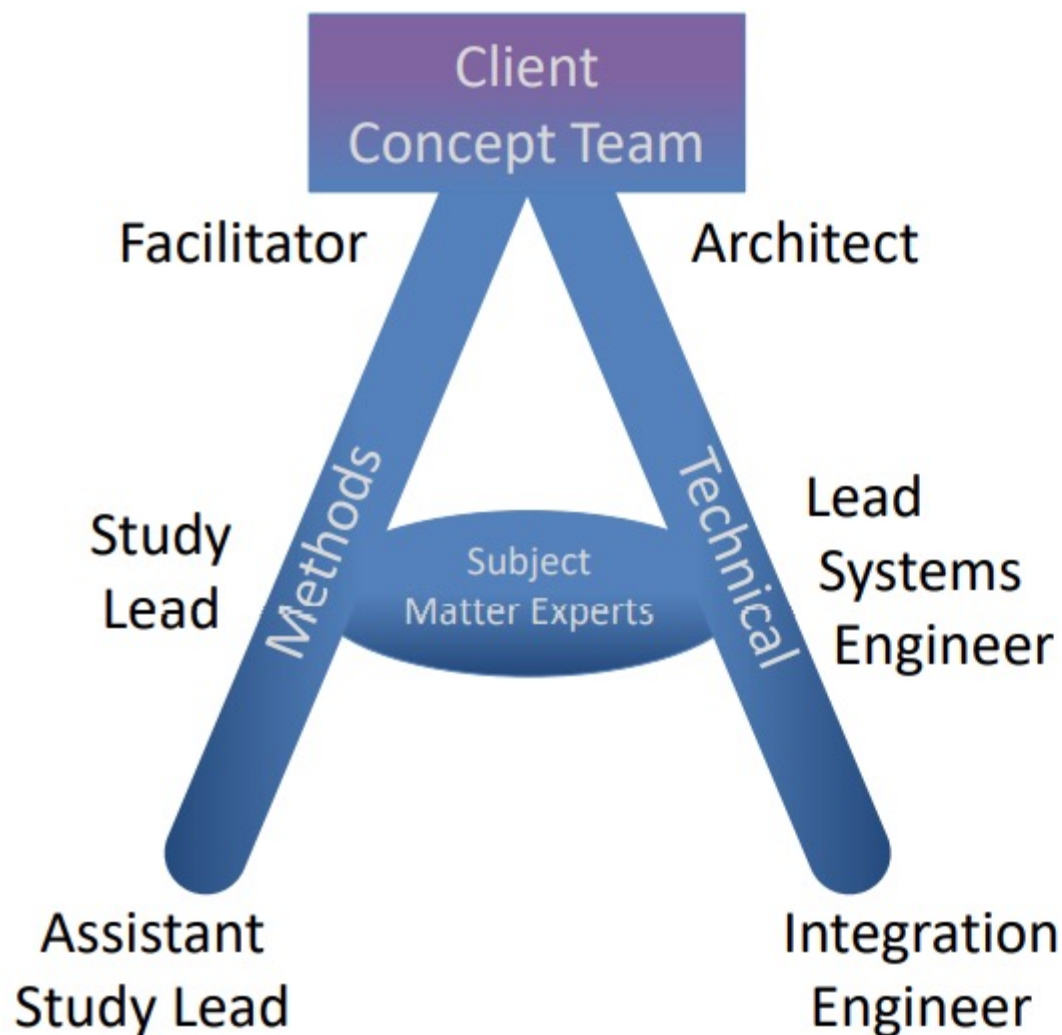
ESSA FASE É FEITA POR ESTRUTURADORES DE ARQUITETURA... EXEMPLO É O LEFT FIELD DO JPL





JPL Innovation Foundry

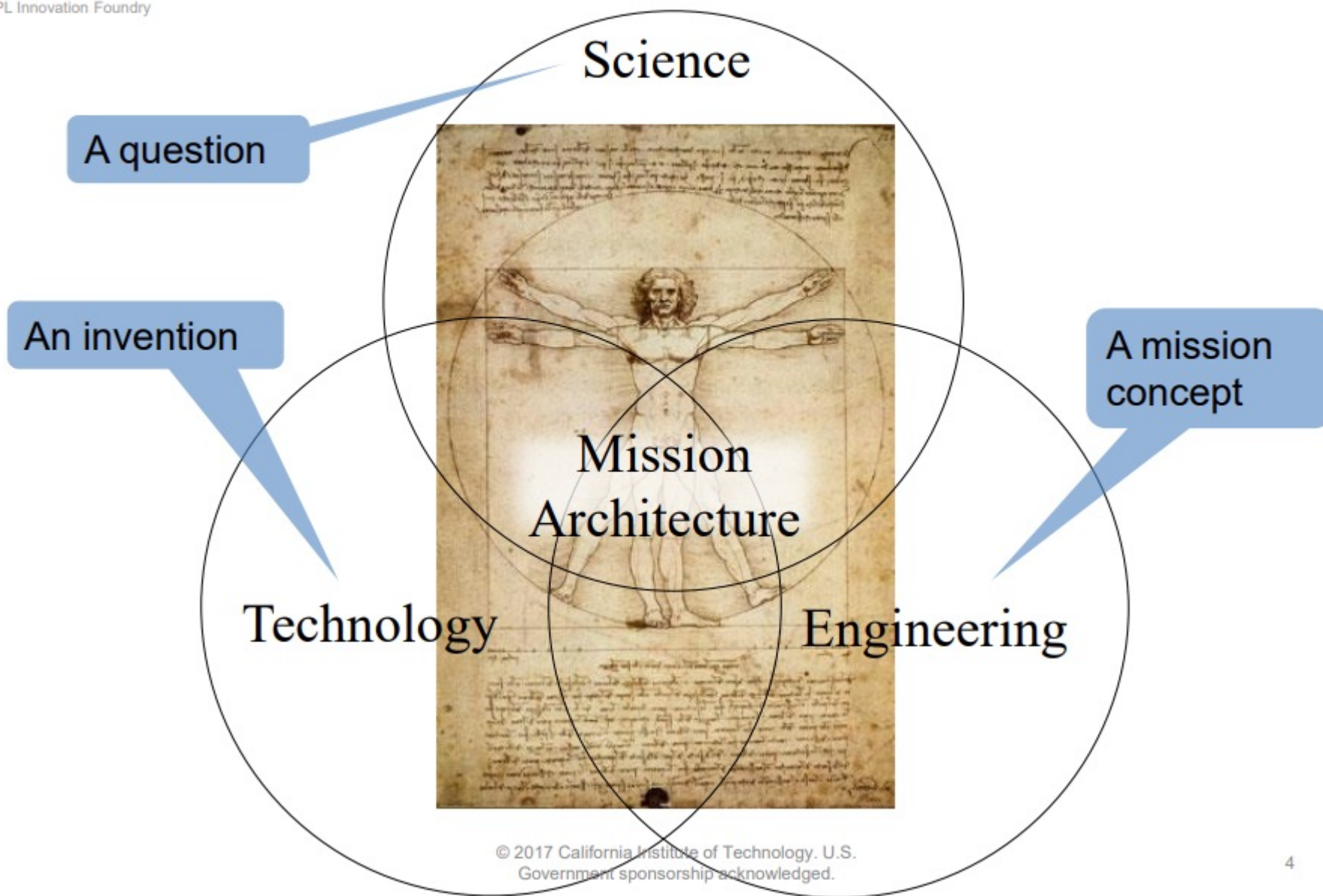
The “A-Frame”



- Each A-Team study has a 3-6 person “A-Frame Team” from two points of view:
 - Innovative Methods
 - Technical Expertise
- Additional subject matter experts are brought in as needed (customized)
- The client may also add members from the Concept Team

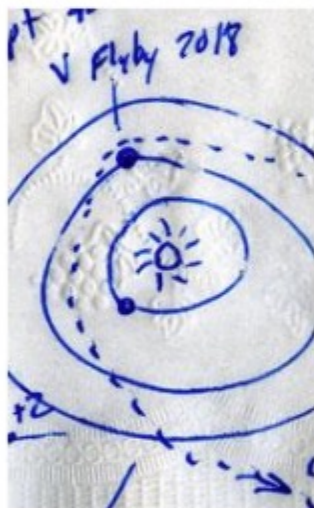


Every mission starts with a spark

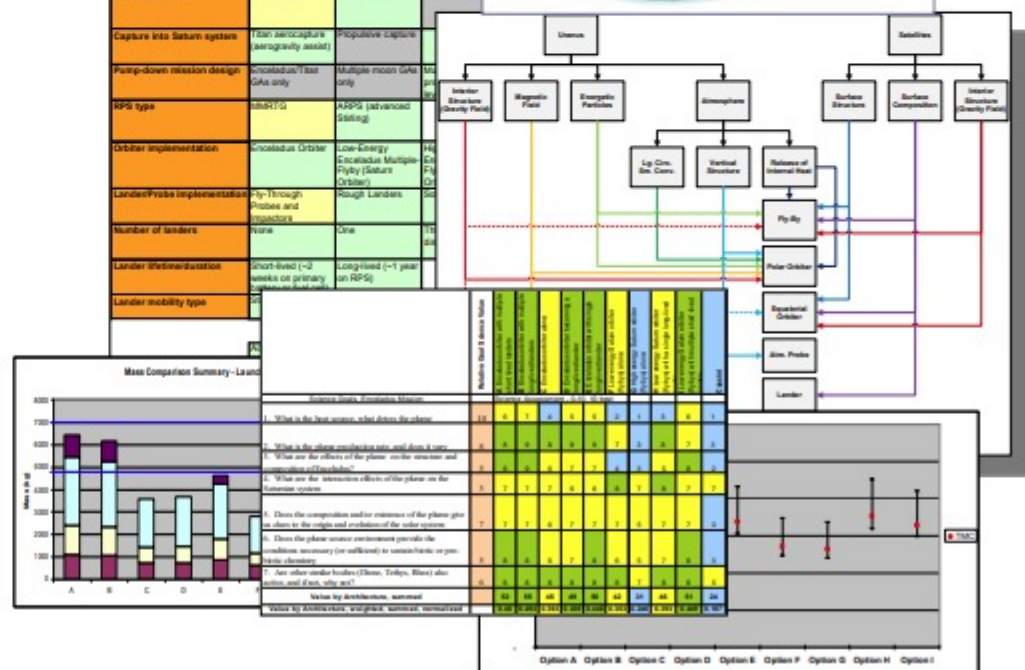
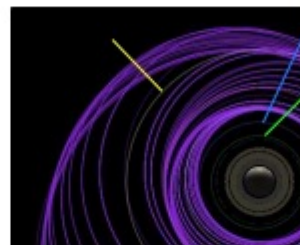




...then the concept is developed



or



One person's
concept is another's
doodle...

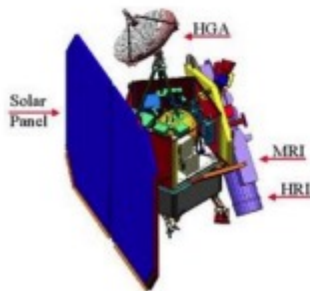


JPL Innovation Foundry

A-Team Methods

CML 1

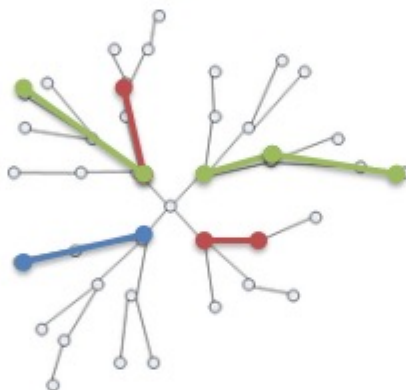
Capturing ideas and linking associated ideas



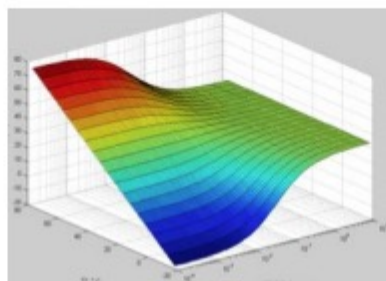
Research, bringing in previous studies

CML 2

Testing assumptions, relationships, and links

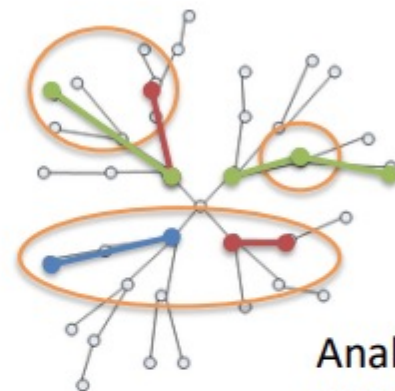


Analyzing feasibility, finding FOMs and thresholds

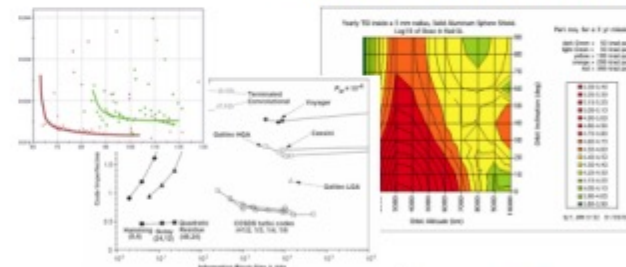


CML 3

Building seed science cases and concept architectures



Analysis and trade space exploration



Rapid prototyping of concepts





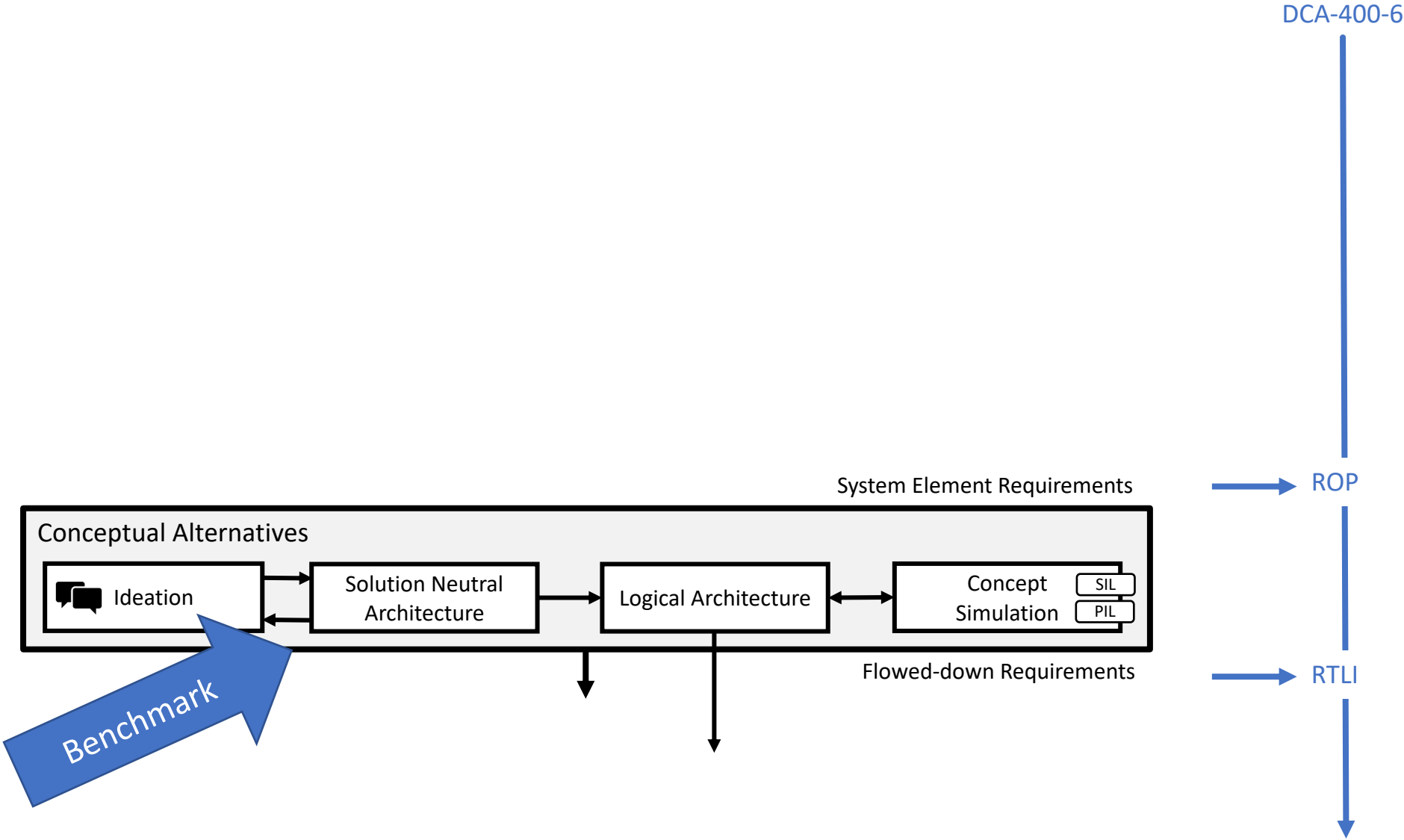
DECISION MATRIX – PUGH’S METHOD

		Alternatives			
		Wt	Vendor 1	Vendor 3	Vendor 4
Criteria	Cost	.30	4	4	4
	Response time	.17	3	3	5
	Training time	.17	2	4	5
	Ease of use	.17	1	4	4
	Strong team	.10	3	4	
	Team experience	.10	3	4	
Total					
Weighted total					

		Alternatives			
		Wt	Vendor 1	Vendor 3	Vendor 4
Criteria	Cost	.30	Datum	+	S
	Response time	.17		+	+
	Training time	.17		-	S
	Ease of use	.17		+	+
	Strong team	.10		-	-
	Team experience	.10		S	-
Pluses		1.0		3	2
Minuses				2	2
Overall total				+1	0
Weighted total				+.37	+.14

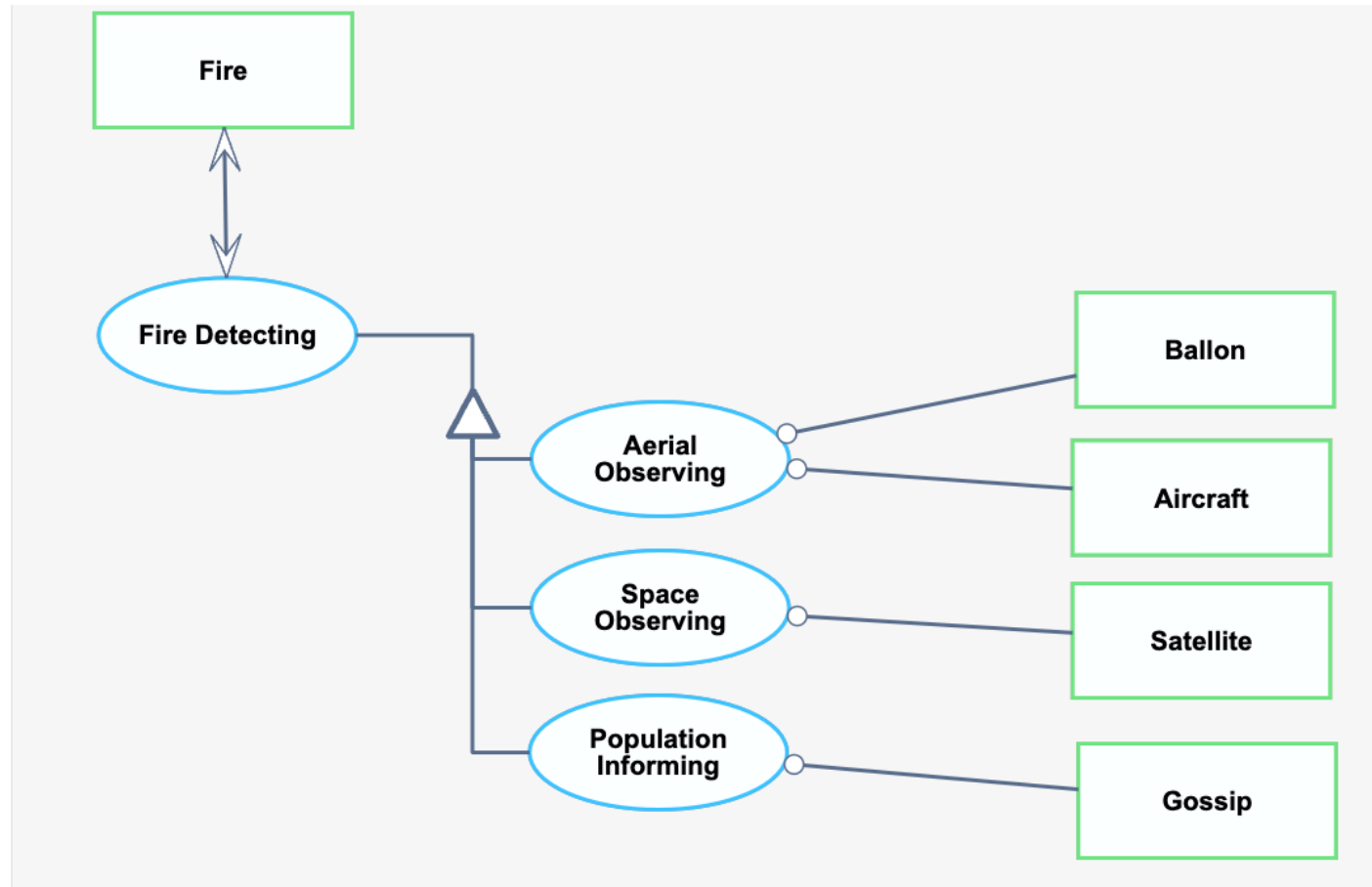
- Meios de **pontuar cada conceito alternativo** em sua capacidade de atender a um conjunto de critérios (MoEs).
- Comparando as pontuações, você desenvolve uma visão sobre as melhores alternativas e as informações mais úteis para tomar sua decisão.

Figure 2.3: The two forms of a Decision Matrix.



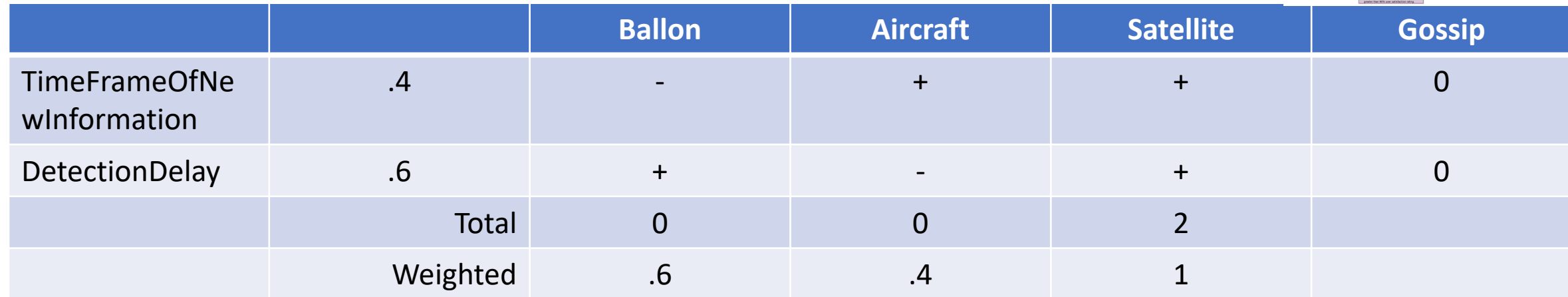


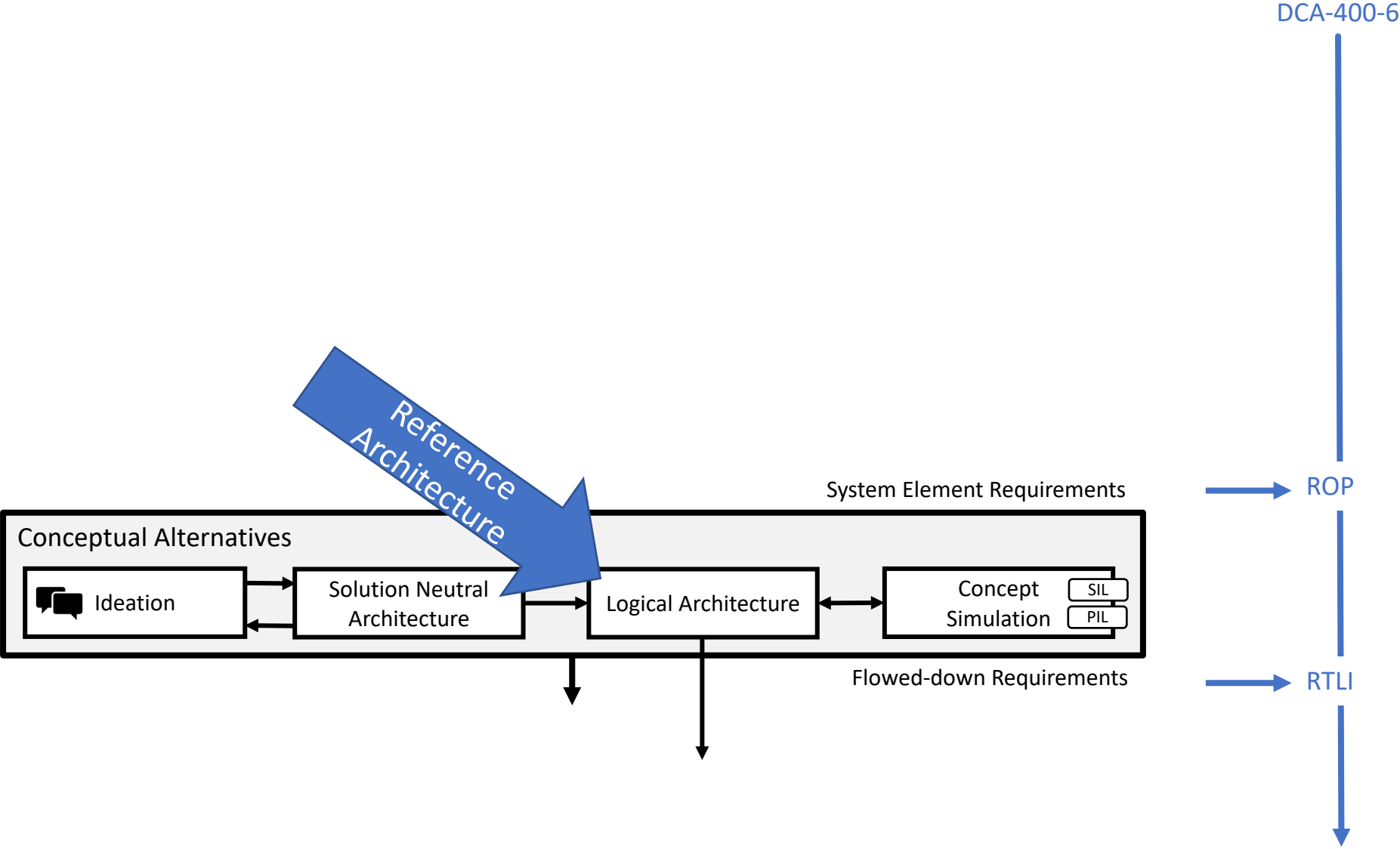
INTENTION EXPLORATION

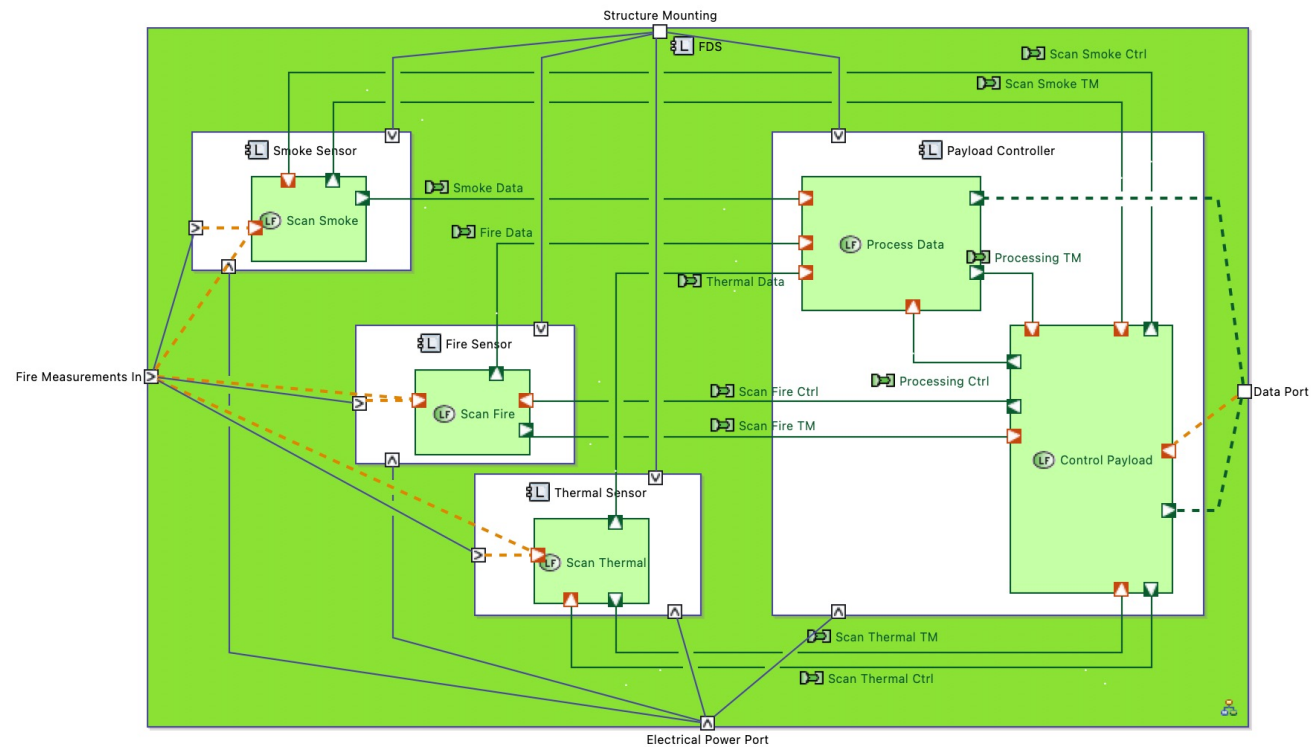
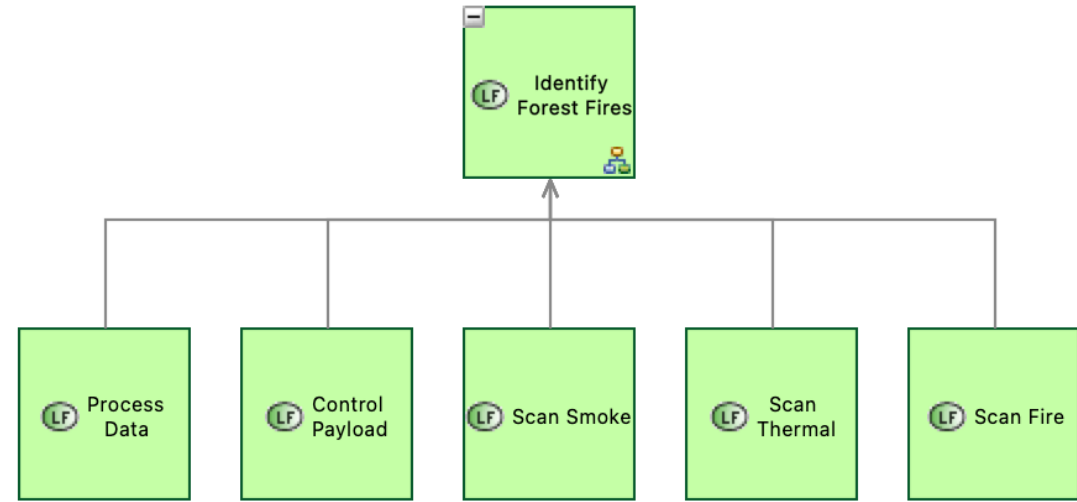
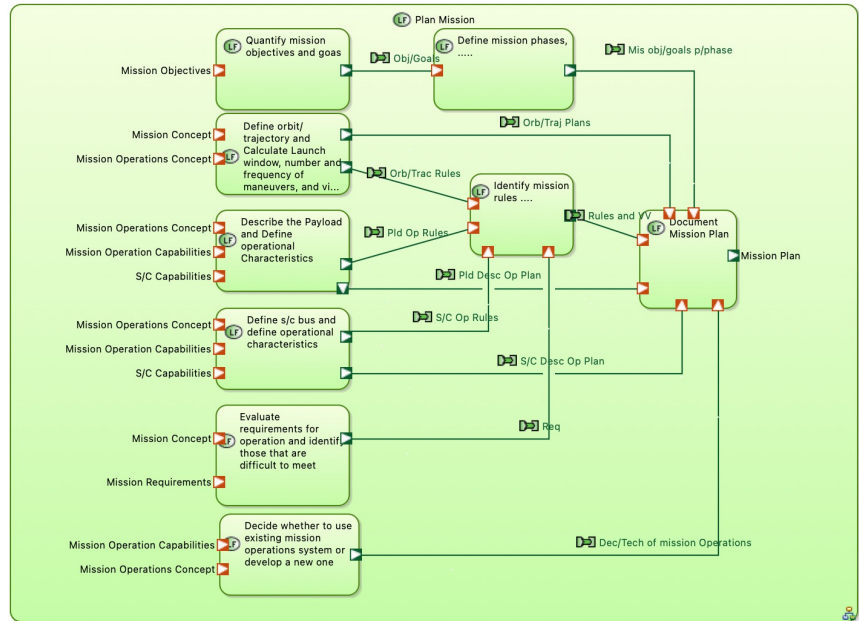
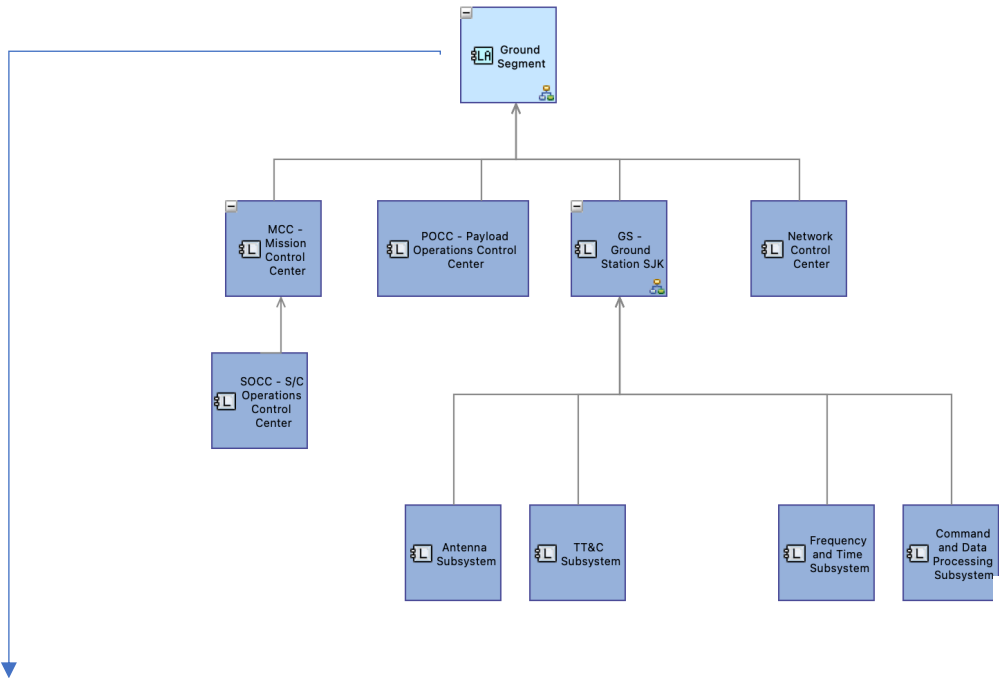


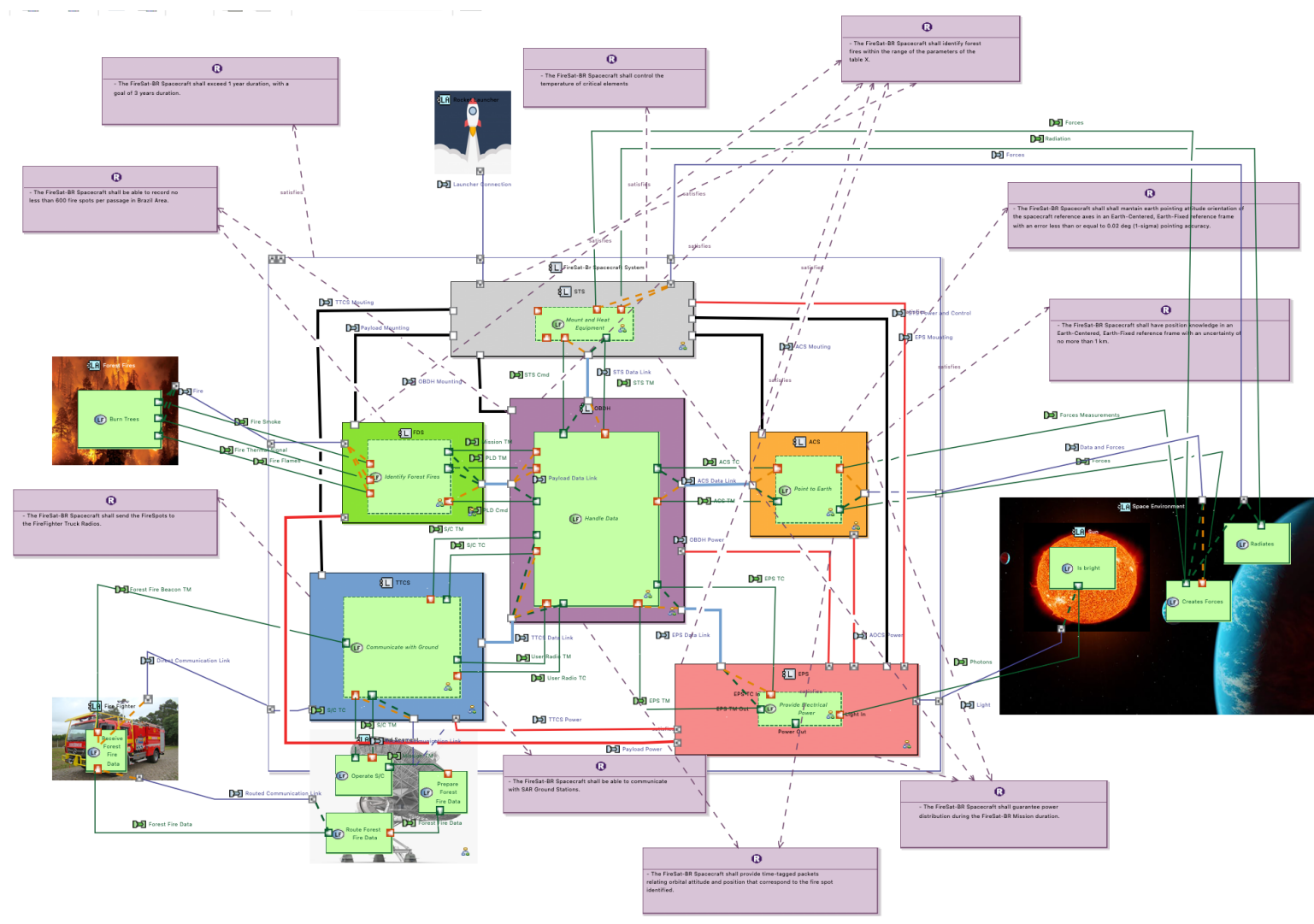
OPL

1. Aerial Observing, Population Informing and Space Observing are Fire Detecting.
2. Fire Detecting affects Fire.
3. Aerial Observing requires Aircraft and Ballon.
4. Space Observing requires Satellite.
5. Population Informing requires Gossip.





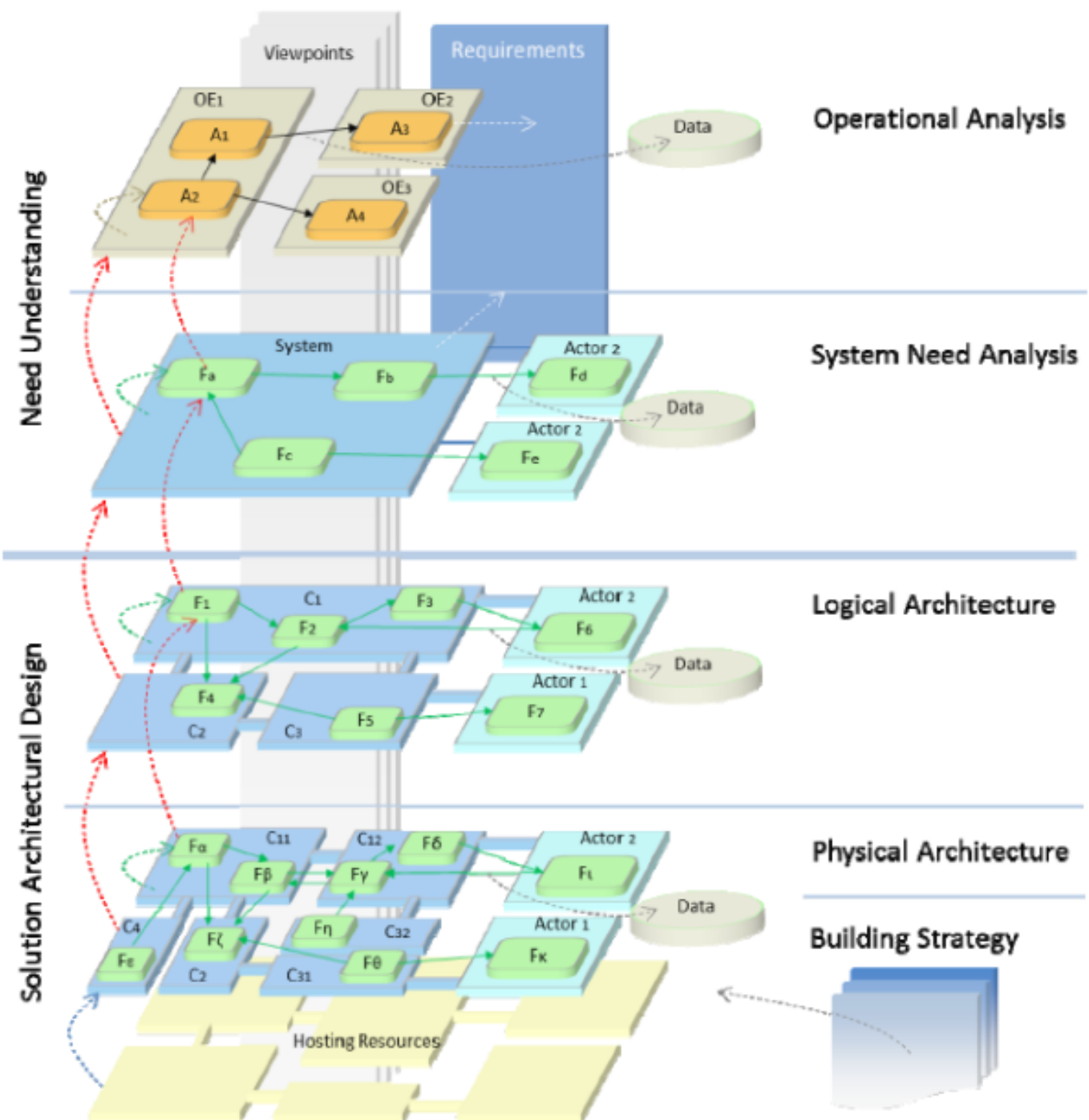




- Logical Architecture
- Capella Module
 - System Requirements
 - [SYS-XXX] The FireSat-BR Spacecraft shall guarantee power
 - [IE PUID] SYS-XXX
 - [Rationale] null
 - [VV Method] null
 - [VV Success Criteria] null
 - [VV Phase] null
 - [VV Procedure] null
 - [VV Report] null
 - [SYS-XXX] The FireSat-BR Spacecraft shall maintain e
 - [SYS-XXX] The FireSat-BR Spacecraft shall be able to recor
 - [SYS-XXX] The FireSat-BR Spacecraft shall be able to comr
 - [SYS-XXX] The FireSat-BR Spacecraft shall exceed 1 year d
 - [SYS-XXX] The FireSat-BR Spacecraft shall control the tem
 - [SYS-XXX] The FireSat-BR Spacecraft shall send the FireSp
 - [SYS-XXX] The FireSat-BR Spacecraft shall identify forest f
 - [SYS-XXX] The FireSat-BR Spacecraft shall provide time-ta
 - [SYS-XXX] The FireSat-BR Spacecraft shall have position k



ARQUITETURA FUNCTIONAL/CONCEITUAL LOGICAL ARCHITECTURE





O QUE HÁ NA ARQUITETURA CONCEITUAL (LOGICAL ARCHITECTURE - LA)?



LA

“Como o sistema funcionará para atender às expectativas”

- Em resposta à necessidade expressa pelas duas perspectivas anteriores, ele permite as primeiras grandes opções de design de solução, primeiro através de uma **análise funcional interna do sistema: descreve as funções a serem executadas e montadas para implementar as funções de serviço identificadas na fase anterior.**
- Prossegue com a **identificação dos componentes que implementam estas funções** da solução, integrando as restrições desejadas.



- O nível de Arquitetura Conceitual visa **identificar os possíveis componentes funcionais** dentro do Sistema (“Como o sistema funcionará para atender às expectativas”), suas relações e seu conteúdo, **independentemente de quaisquer considerações de tecnologia ou implementação**.
- Em seguida, deve ser realizada uma análise funcional do sistema: **devem ser identificadas as subfunções necessárias para executar as Funções do Sistema escolhidas durante a fase anterior**; então, **uma divisão em componentes para os quais essas subfunções internas serão alocadas deve ser determinada**, ao mesmo tempo em que integra as restrições não funcionais.



- A definição de LA (uma atividade muitas vezes – e erroneamente – designada "arquitetura lógica" por conveniência, e também chamada de arquitetura de referência) **consiste principalmente em uma comparação entre as necessidades expressas em perspectivas anteriores, uma análise funcional** descrevendo o comportamento do sistema escolhido para satisfazer os requisitos, e uma **análise estrutural destinada a identificar os componentes** que constituirão o sistema, tendo em conta as condicionantes escolhidas e os princípios estruturantes.
- A LA é, portanto, **uma primeira visão geral**, detalhada até o nível desejado pela organização, de alguma forma uma abstração, do que será a arquitetura do Sistema.



AS PRINCIPAIS ATIVIDADES PARA A DEFINIÇÃO DA ARQUITETURA SÃO AS SEGUINTEs:

1. definir os **fatores que impactam os** pontos de vista da arquitetura e da análise;
2. Definição dos **princípios de comportamento do sistema**;
3. para **construir alternativas** de estruturação de sistemas baseados em componentes;
4. para selecionar a **alternativa de arquitetura que oferece o melhor compromisso**.



1. DEFINIR OS FATORES QUE IMPACTAM OS PONTOS DE VISTA DA ARQUITETURA E DA ANÁLISE

- Qualquer arquitetura **adequadamente projetada** **satisfaz várias expectativas e restrições**, que restringem e influenciam ou mesmo direcionam sua definição, e cuja **satisfação deve ser verificada** o mais cedo possível para minimizar possíveis custos de retrabalho.
- Esses fatores que restringem a arquitetura dependem em grande parte de cada domínio. Como exemplos, citamos: serviços e custos entregues, desempenho esperado, segurança das operações, privacidade, facilidade de manutenção, duração da vida, pegada energética ou logística, disponibilidade, política do produto, escalabilidade, mas também considerações mais "estéticas", como a satisfação do cliente.



- No caso do sistema de controle de tráfego, o primeiro **fator de impacto é, obviamente, a segurança dos bens e das pessoas.**
- Um factor adicional envolve os operadores de rede, a sua **formação e as competências exigidas, o âmbito da sua responsabilidade e o papel que lhes deve ser atribuído.**
- Devemos também ter em conta factores como as **condições ambientais, a duração da vida, as restrições de logística e manutenção.**
- No caso do sistema de controle de tráfego, ainda temos: **a taxa de confiabilidade, a probabilidade de falha do sistema, a capacidade de operar em caso de falha parcial de certos subsistemas; o número máximo elegível de operadores; faixas extremas de temperatura, umidade, etc.**



2. DEFINIÇÃO DOS PRINCÍPIOS DE COMPORTAMENTO DO SISTEMA

- O objetivo é formalizar os princípios do comportamento desejado do sistema, aos quais ele tem a responsabilidade de responder durante sua operação.
- *Um erro comum consiste em considerar o comportamento da solução como um simples refinamento da expressão funcional anterior da necessidade em um nível mais fino de detalhe. O design da solução é muito mais do que isso: é uma tomada de decisão em conta das restrições, nomeadamente o esforço de ser "criativo" para estruturação de um comportamento que satisfaz a necessidade (e que não a refina), detalhando os processos e etapas desde as solicitações do sistema, até a prestação de serviços, resultados ou saídas, levando em consideração as decisões de projeto.*

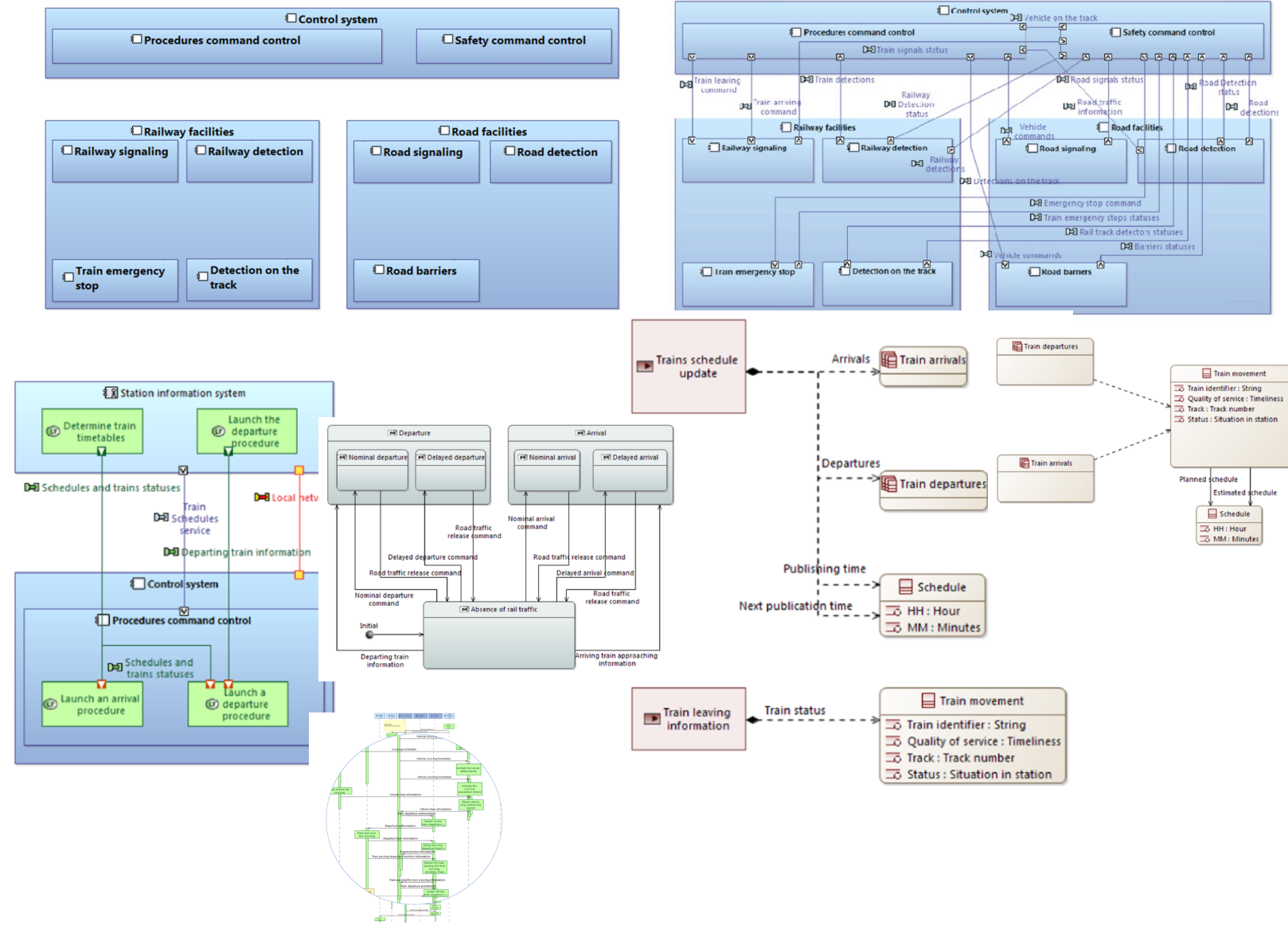


- 1 – **identificar e formalizar** necessidades capturados. (rastreando para SA)
- 2 – busca de **possíveis funções já existentes na LA** que também possam desempenhar um **papel para resolver as necessidades**. (minimizar funções)
- 3 – verificar os **limites das funções** para alcançar o que se espera delas.
 - Cenários/cadeias darão luz às decisões de projeto ou à escolha de linhas de produtos.
- 4 – construir uma **descrição global** usando os elementos comportamentais (cenários/máquinas de estado).



3. CONSTRUIR ALTERNATIVAS DE ESTRUTURAÇÃO DE SISTEMAS BASEADOS EM COMPONENTES

- Descrever estrutura preliminar em componentes e subcomponentes.
- Definir as relações entre os componentes e
- definir como são as interfaces dos componentes.
- Instanciar/organizar as funções e definir a dinâmica dos componentes.





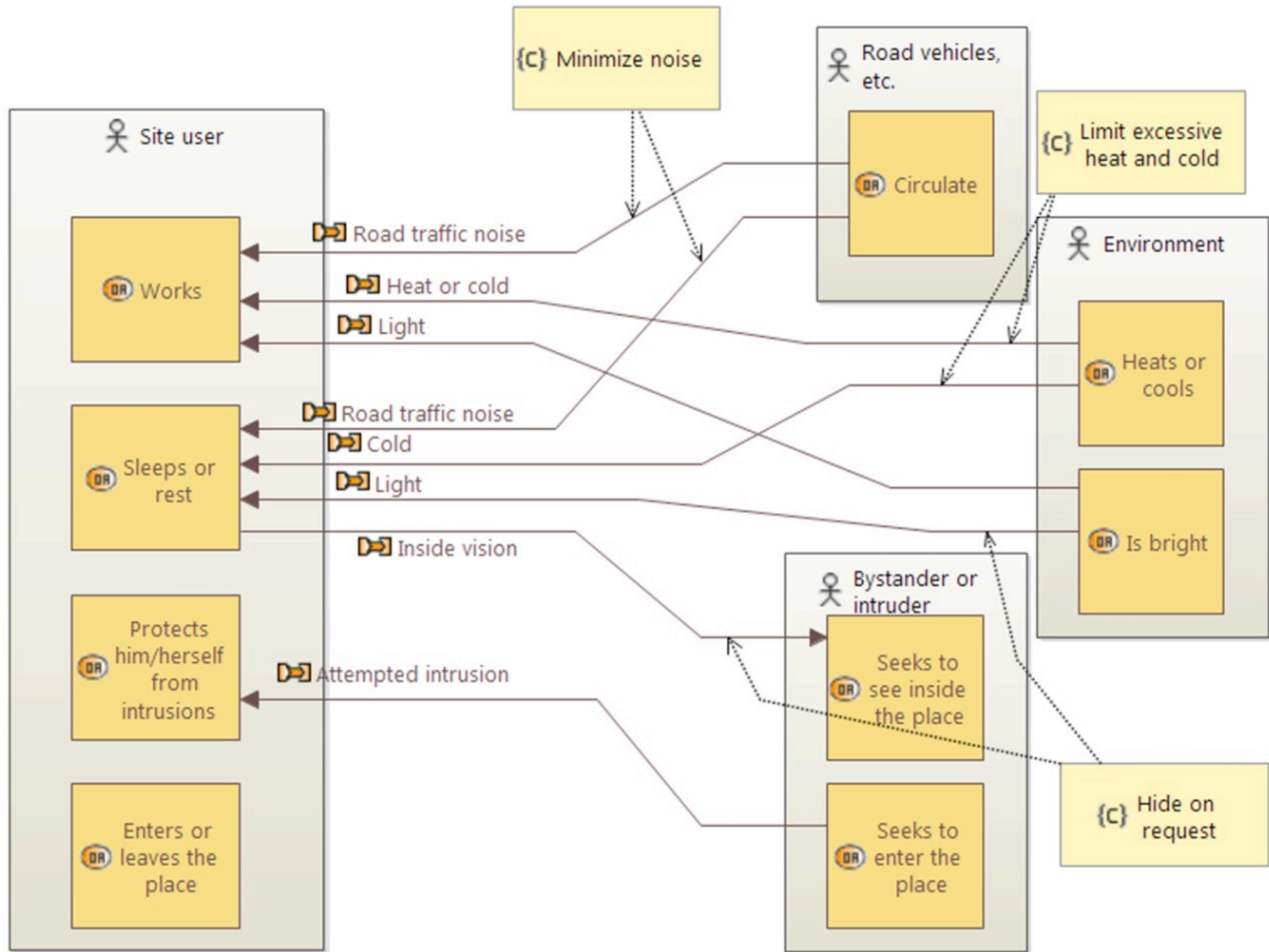
4. SELEÇÃO DA ALTERNATIVA DE ARQUITETURA QUE OFERECE O MELHOR TRADE-OFF

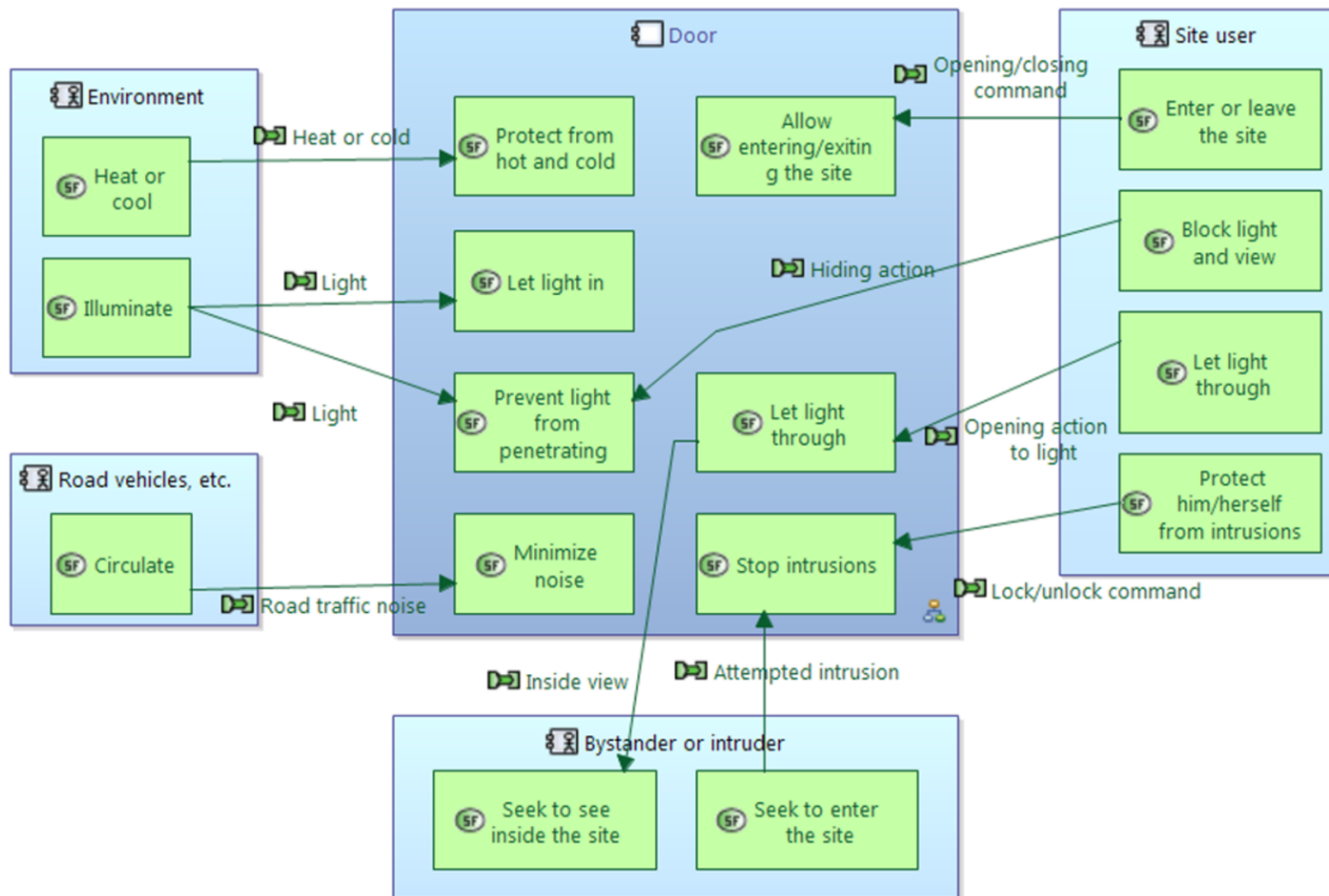
- O objetivo desta atividade é **encontrar entre as arquiteturas candidatas que representam o melhor trade-off em relação a todos os pontos de vista em consideração**, e justificar sua conformidade com a necessidade.
- Cada alternativa foi, em princípio, **avaliada com base nos principais pontos de vista que a impactaram** – e sua importância relativa – durante sua definição; as não conformidades foram eliminadas, mas como a avaliação raramente é binária, trata-se, portanto, de **comparar os "méritos" de cada candidato em uma análise multicritério**, da qual análises de pontos de vista, prioridades e métricas previamente identificadas são elementos-chave.

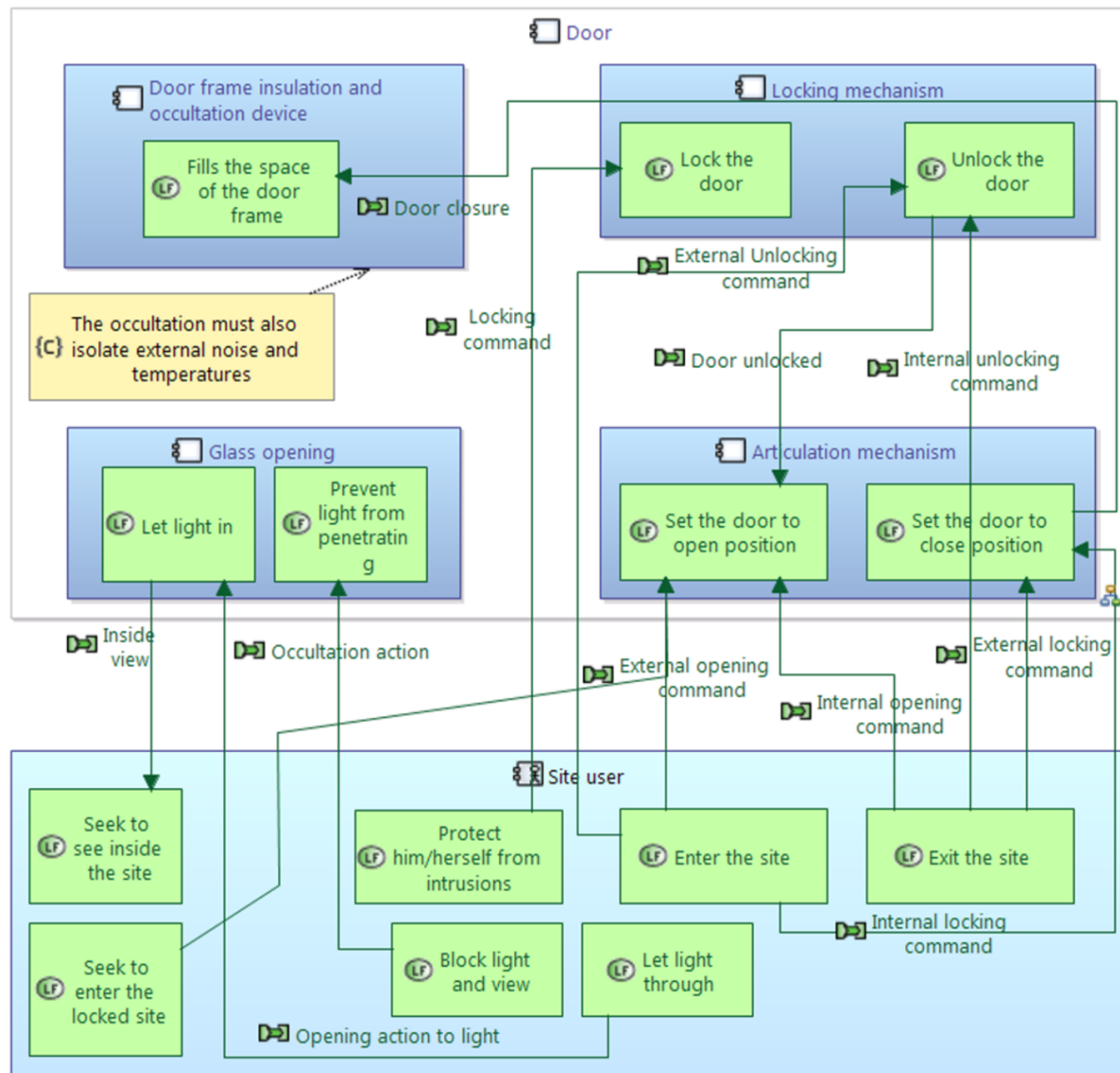


ARCADIA METHOD – LOGICAL ARCHITECTURE ANALYSIS SUMMARY

Define the factors impacting the architecture and analysis viewpoints	satisfies a number of expectations and constraints of various kinds, which constrain and influence or even direct its definition, and whose satisfaction should be verified as early as possible to minimize possible subsequent resumption costs.
Define the principles underlying the system behavior	non-functional, to which it has the responsibility to respond during its operation under operational conditions
Build component-based system structuring alternatives	The system is broken down into principle components called logical components. The term “component” is understood here in the general sense, as a constituent of the system at this level;
Select the architecture alternative offering the best compromise	find among previous candidate architectures the one that represents the best trade-off with respect to all viewpoints under consideration, and to justify its compliance to the need.







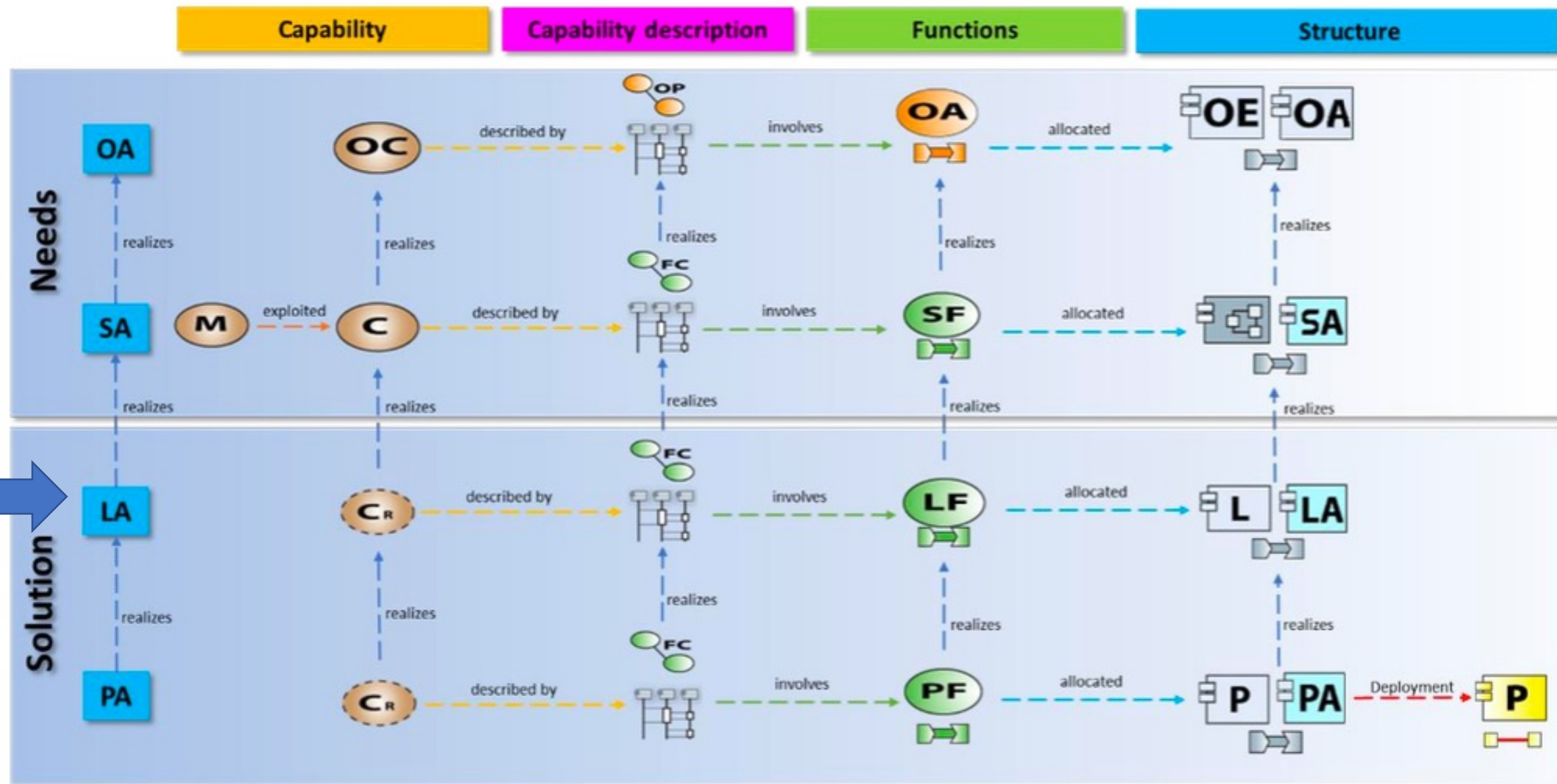


Figure 2.3: Arcadia ontology traceability










































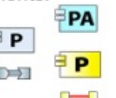





Arcadia layer	Requirements	Capability	Capability description	Functional	Structure	Modes and States	Data	Interfaces
Operational Analysis	R-OA	OA1	OA2	OA3	OA4	M&S-OA5	D-OA6	I-OA7
	Capture stakeholder requirements	Define Operational Capabilities	Define processes and scenarios	Define Operational Activities 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
	 					 	 	
System Analysis	R-SA	SA1	SA2	SA3	SA4	M&S-SA5	D-SA6	I-SA7
	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.
	 	 				 	 	
Logical Architecture	R-LA	LA1	LA2	LA3	LA4	M&S-LA5	D-LA6	I-LA7
	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.
	 					 	 	
Physical Architecture	R-PA	PA1	PA2	PA3	PA4	M&S-PA5	D-PA6	I-PA7
	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



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

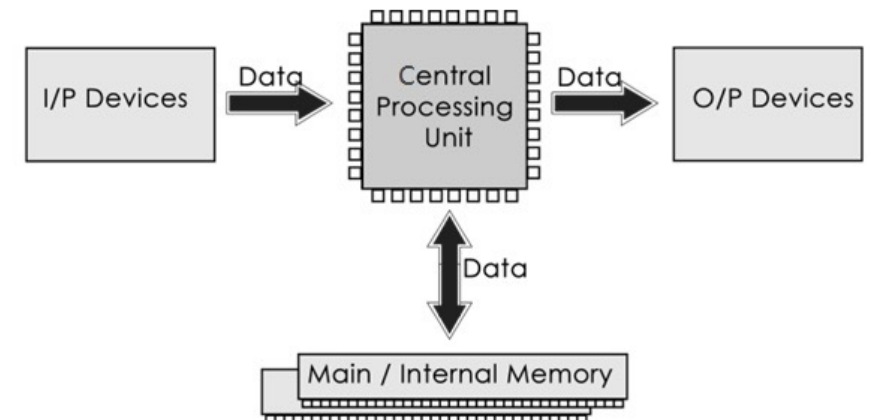
Table 3.3: Arcadia diagrams matrix



VOCABULÁRIO DA ARQUITETURA FUNCIONAL (LOGICAL ARCHITECTURE)

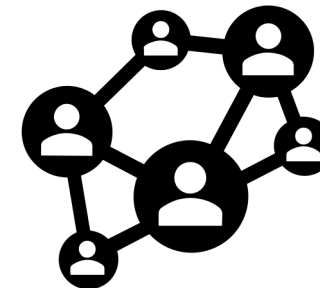
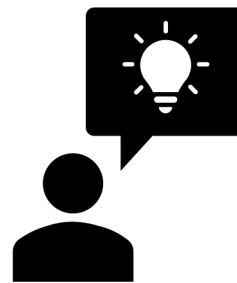
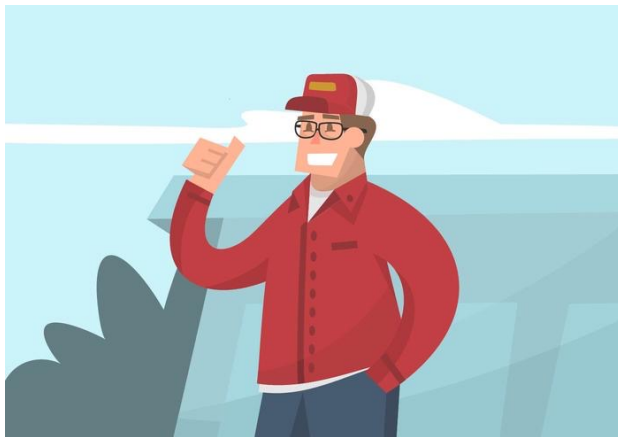


- **Componente lógico (Logical Component):** elementos estruturais dentro do Sistema, com “portas” que permitem a interação entre components e atores. Um componente lógico pode ter uma ou mais funções e pode ser dividido em subcomponentes.



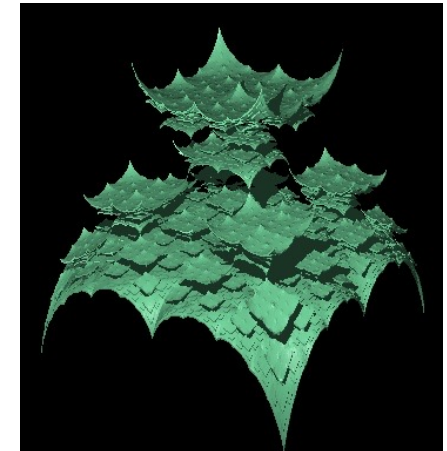
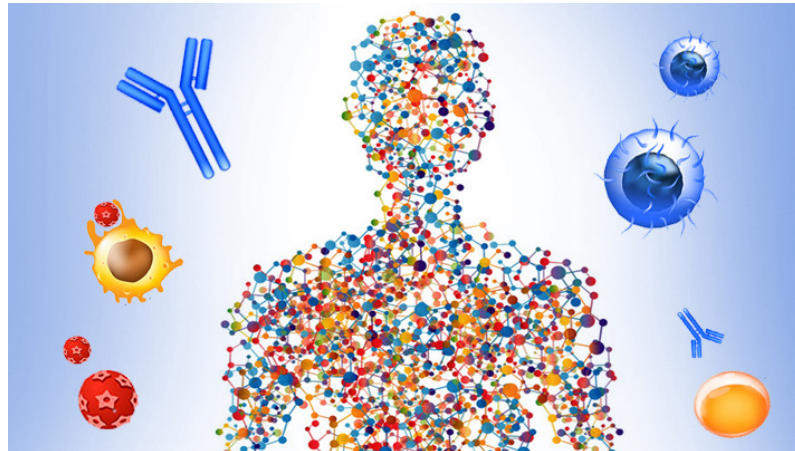
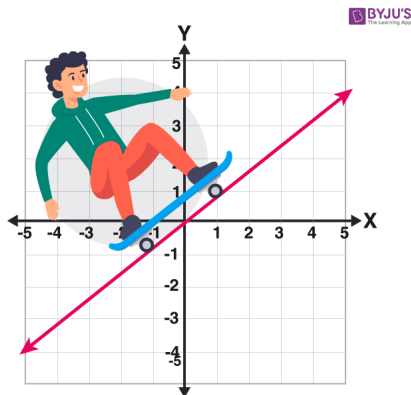


- **Atores (Logical Actor):** qualquer elemento externo ao Sistema (humano ou não humano) e que interaja com ele (por exemplo, Piloto, Operador de Manutenção, etc.).



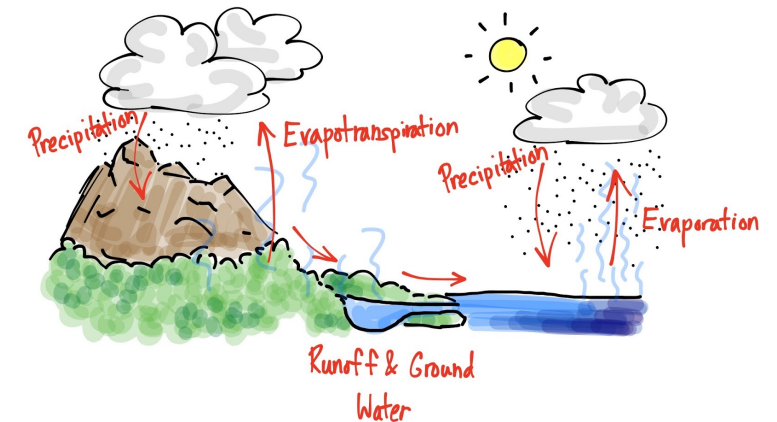
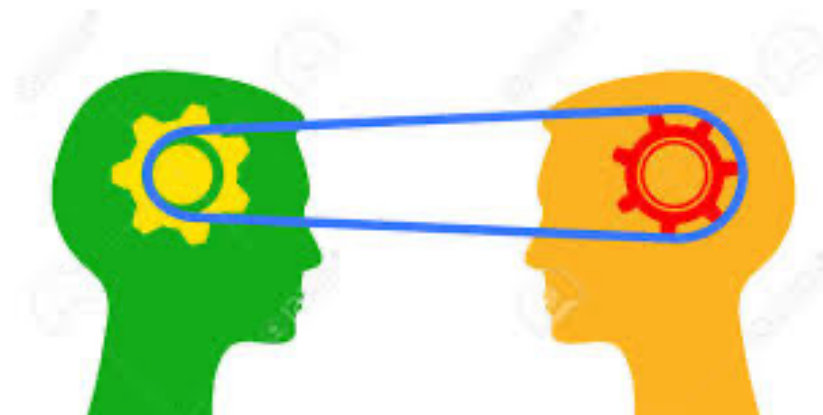


- **Funções (Logical Function):** comportamento ou serviço fornecido por um Componente ou por um Ator. Uma Função tem Portas que lhe permitem comunicar com as outras Funções. Uma função pode ser subdividida em subfunções;



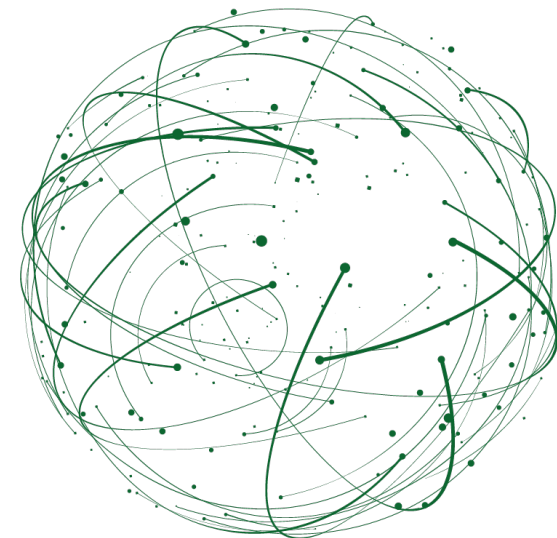


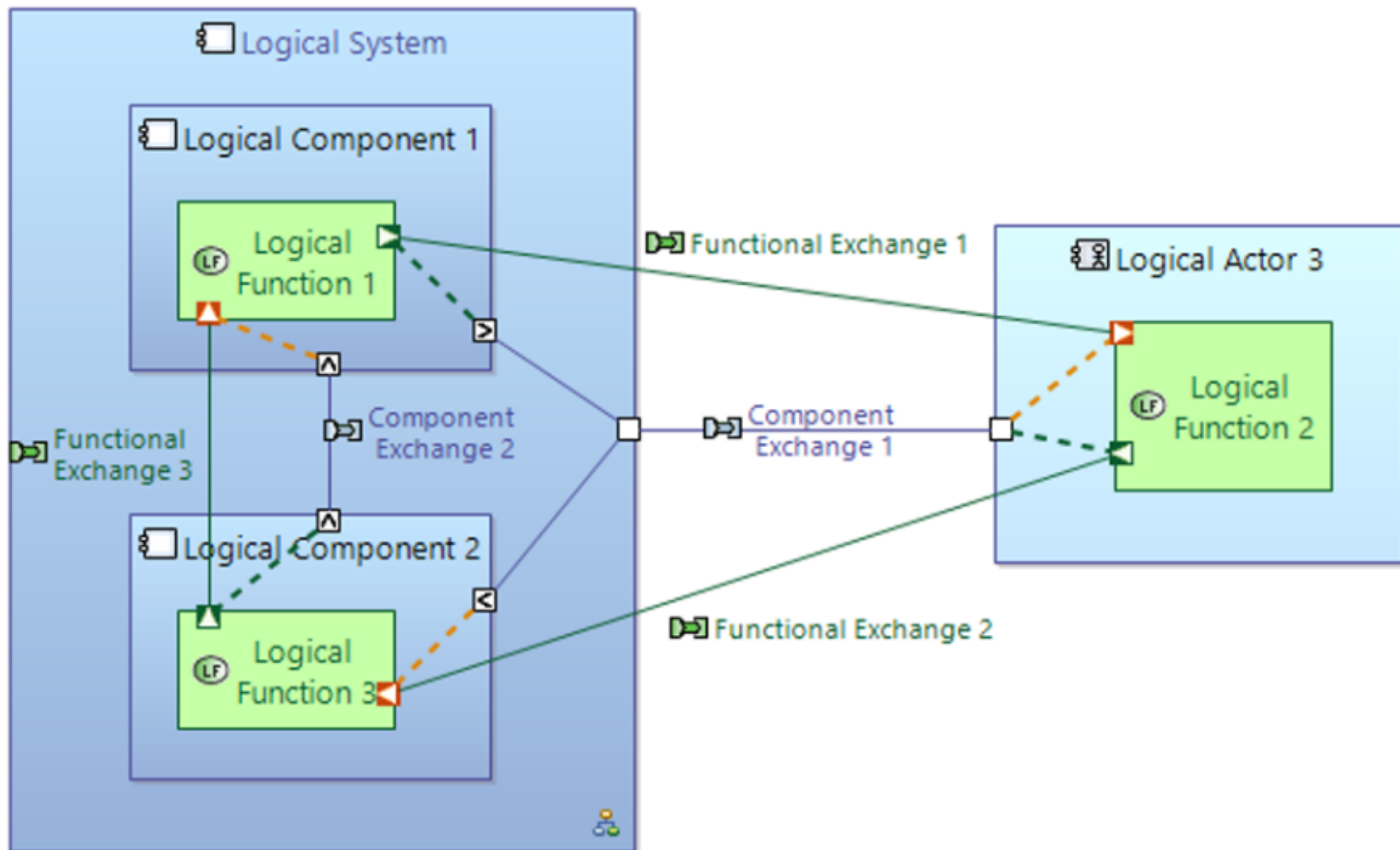
- **Troca Funcional (Functional Exchange):** uma troca unidirecional de informações, matéria ou energia entre duas funções, ligando duas Portas de Função;





- **Troca entre components (Component Exchange):** conexão entre os Componentes e/ou os Atores, permitindo a circulação das Trocas Funcionais;







DIAGRAMAS



▼ Transition from System Functions



[Perform an automated transition of System Functions](#)



[Create Traceability Matrix](#)

Inicialização e atualização automatizada das funções lógicas de acordo com as funções do sistema

As ferramentas de transição criam um primeiro mapeamento de rastreabilidade 1-1 entre a Arquitetura Lógica e a Análise do Sistema. Use matrizes de rastreabilidade dedicadas para modificar as relações de rastreabilidade.

▼ Refine Logical Functions, describe Functional Exchanges



[\[LFBD\] Create a new Functional Breakdown diagram](#)



[\[LDfB\] Create a new Functional Dataflow Blank diagram](#)



[\[FS\] Create a new Functional Scenario](#)

Enriqueça e detalhe o detalhamento funcional com novas funções lógicas.

Descrever os fluxos de dados entre funções lógicas e identificar funções específicas e cadeias funcionais.

▼ Define Logical Components and Actors



[Perform an automated transition of System Actors](#)



[\[LCBD\] Create a new Logical Component Breakdown diagram](#)



[\[LAB\] Create a new Logical Architecture diagram](#)

A inicialização e atualização automatizada dos atores lógicos pode ser realizada de acordo com os atores do sistema.

Use uma arquitetura ou diagrama de detalhamento para descrever os blocos de construção internos do sistema de um ponto de vista lógico.

Os componentes lógicos são destinados a interagir uns com os outros para atingir os objetivos funcionais do sistema.



▼ Allocate Logical Functions to Logical Components



[\[LAB\] Create a new Logical Architecture diagram](#)



[\[ES\] Create a new Exchange Scenario](#)



[Create a new allocation Logical Component / Logical Function Matrix](#)

▼ Delegate System Interfaces and create Logical Interfaces



[\[CII\] Create a new Contextual Internal Interface diagram on the Logical System Component](#)

▼ Enrich Logical Scenarios



[Perform an automated transition of System Analysis Capabilities](#)



[\[IS\] Create a new Interface Scenario](#)

Os componentes lógicos são responsáveis pela implementação das funções lógicas. Gerencie essas alocações usando um diagrama de arquitetura e deduza as trocas de componentes implementando as trocas funcionais.

Crie cenários de fluxos de dados para ilustrar as trocas funcionais entre os componentes.

Use a ferramenta de sincronização automatizada para inicializar o sistema lógico raiz de acordo com as interfaces definidas na Análise do Sistema.

Delegue cada interface do sistema a um ou mais componentes lógicos. Crie interfaces internas entre subcomponentes.

Especifique o comportamento dinâmico dos componentes lógicos completando as sequências de interação provenientes da Análise do Sistema. O enriquecimento das sequências de interação e a identificação das interfaces lógicas são duas atividades muito estreitas e iterativas.

O processo de refinamento do cenário é iterativo, cada atualização em uma origem pode ser propagada automaticamente para o destino.

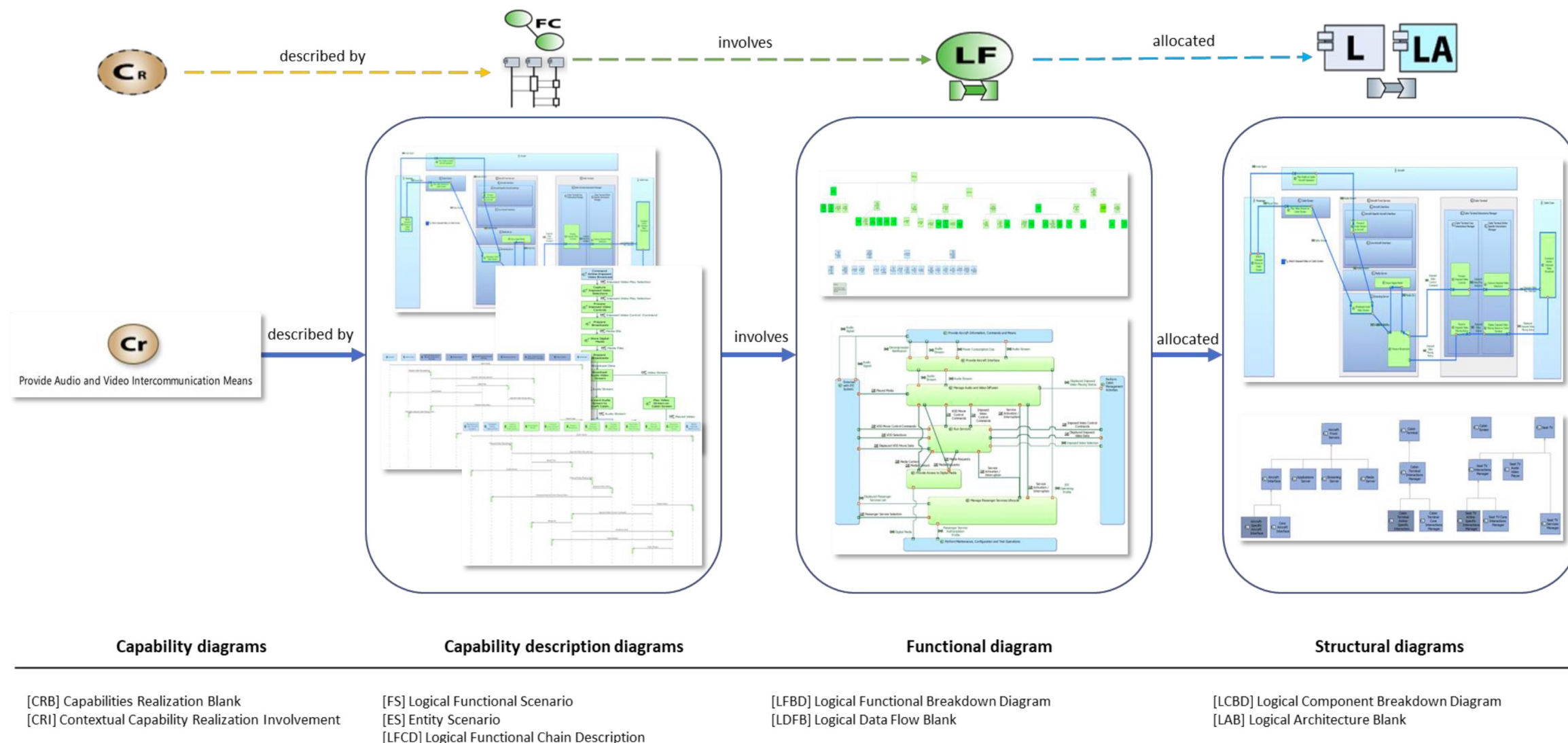


Figure 6.7: Logical Architecture model elements traceability and diagrams relationship



O PODER DO REUSO 😊



ACELERANDO AS COISAS

- **Uma grande vantagem** do “MB” é a capacidade de **reuso dos modelos**.
- Se todo projeto tiver que construir todo o modelo todas as vezes, não será diferente de fazer tudo baseado em documento, **na verdade será até mais demorado**.
 - Por isso temos que se beneficiar dos mecanismos de reuso das ferramentas.
- No Capella temos “dois mecanismos”:
 1. Coleções e Replicas
 2. Bibliotecas



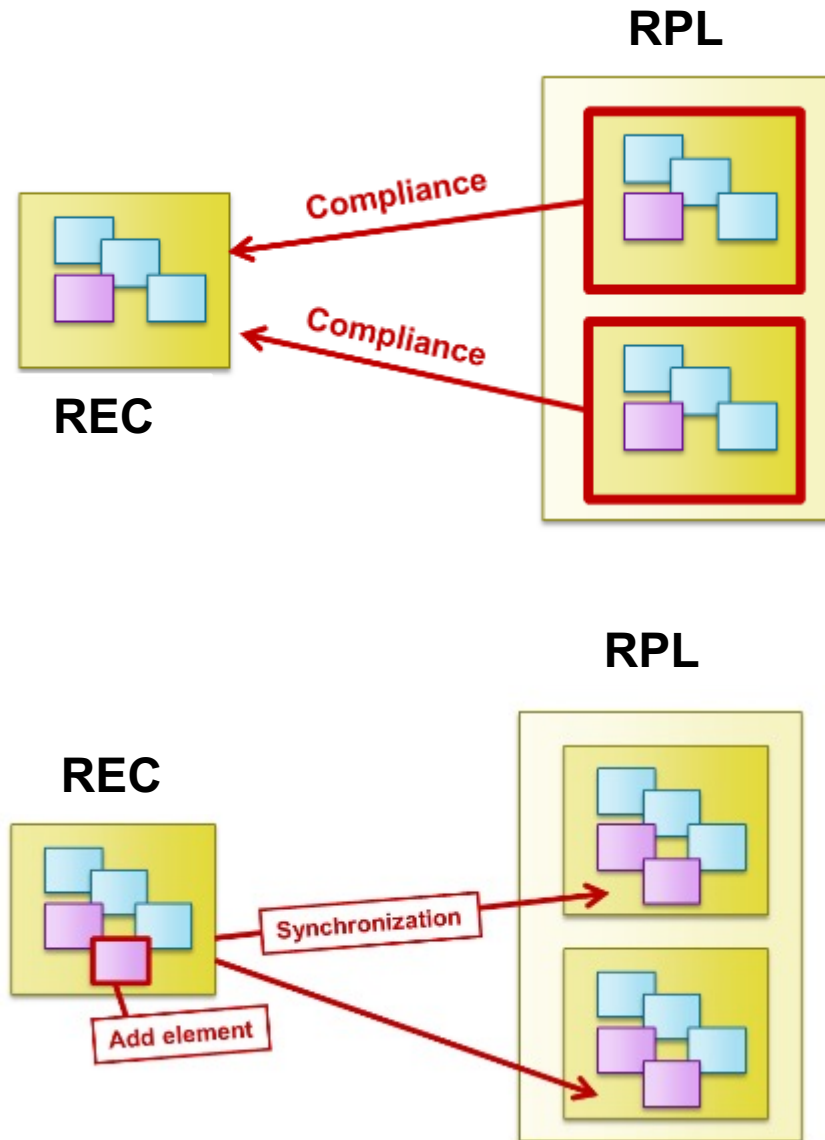
COLEÇÃO DE ELEMENTOS REPLICÁVEIS (REC) E REPLICAS (RPL)

REPLICABLE ELEMENTS COLLECTION (REC) E REPLICAS (RPL)

Escrito por Mateus S. Venturini



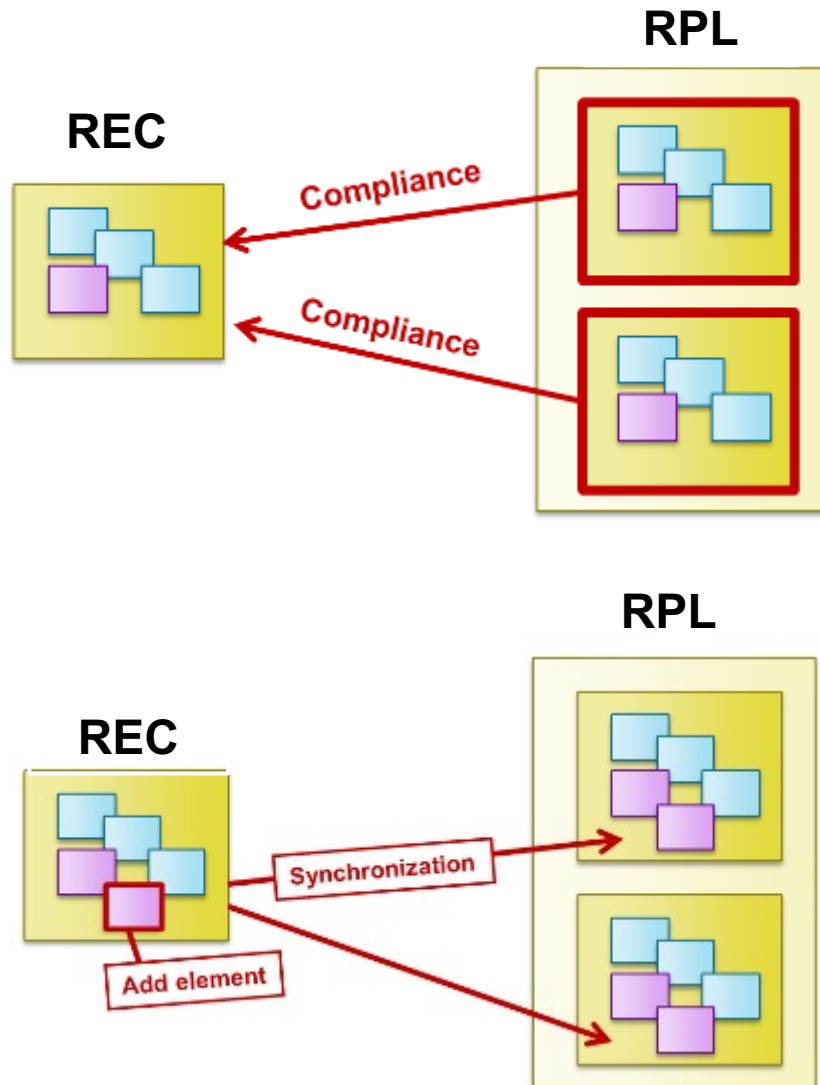
DEFINIÇÃO



- Uma **Coleção de Elementos Replicáveis (REC)** é um conjunto de elementos de modelo, identificados como sendo um **padrão (um modelo no sentido comum do termo)** para a **construção de Réplicas (RPLs)** que mantêm conformidade com ele.



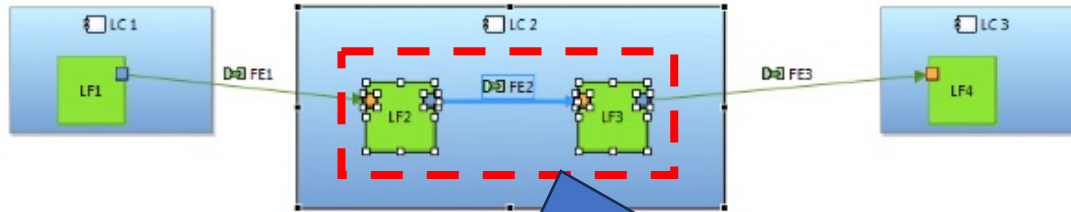
DEFINIÇÃO



- A REC can be viewed as a “contract” to which all its RPLs must comply. REC can embed RPLs of other RECs.
- REC and RPL are located in Catalogs. Technically, REC and RPL are technical objects pointing towards the list of the elements they embed.
- Capella provides tooling to manage the creation of REC and their instantiation, as well as update mechanisms (from REC to RPL and from RPL to REC) and validation rules.
- Different kinds of conformance are possible between a RPL and its REC. Capella defines three default kinds of conformance, but end-user can define their own ones.
 - Blackbox: No modification is allowed on the Replica.
 - Constrained Reuse: Internal elements can be added inside a RPL, but constraints and Interfaces (Function and Component Ports for example) defined in REC cannot be modified.
 - Inheritance: Any element can be added in the RPL, including new Interfaces.

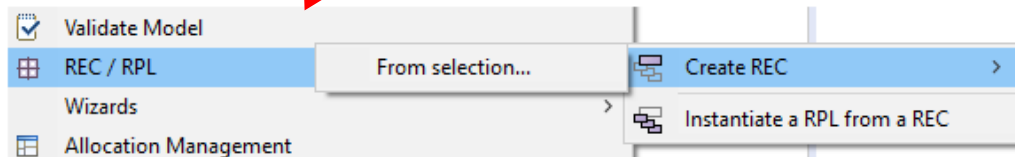


CRIANDO UMA REC



1. From the contextual menu, select "REC/RPL->Create REC->From selection..."

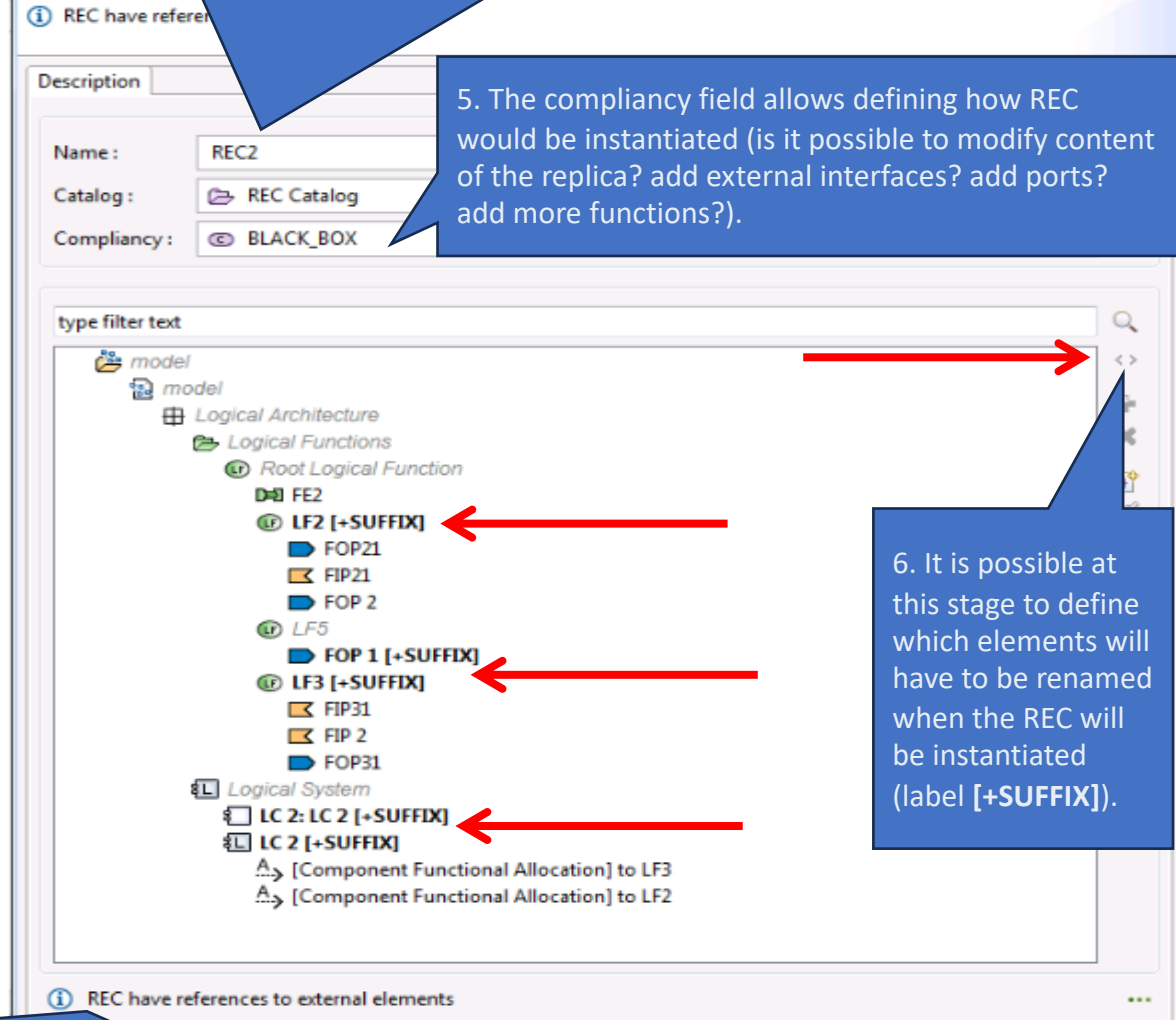
2. From a diagram, select a consistent set of elements (here, a Component and the Functions it is performing).



3. The scope (content) of the REC is displayed. This dialog helps modifying this scope (for example adding or removing elements). By default, the tool applies a set of business rules to include elements (for example, allocations between Component and Functions, children of an element, Etc.). Note here that despite Functional Exchange "fe23" is carrying Exchange Items, these Exchange Items are not included by default in the REC. In most of the cases, they should not be, as references are kept.

4. The REC creation dialog appears. A name shall be given to the REC. The Catalog field allows to select in which catalog this REC should be created. When working with Libraries, the Catalog is most likely located in a Library. In a library, an additional action "With whole library content..." is shown in the REC creation menu. If that action is chosen, the new REC will be initialized with the entire contents of the library.

5. The compliancy field allows defining how REC would be instantiated (is it possible to modify content of the replica? add external interfaces? add ports? add more functions?).



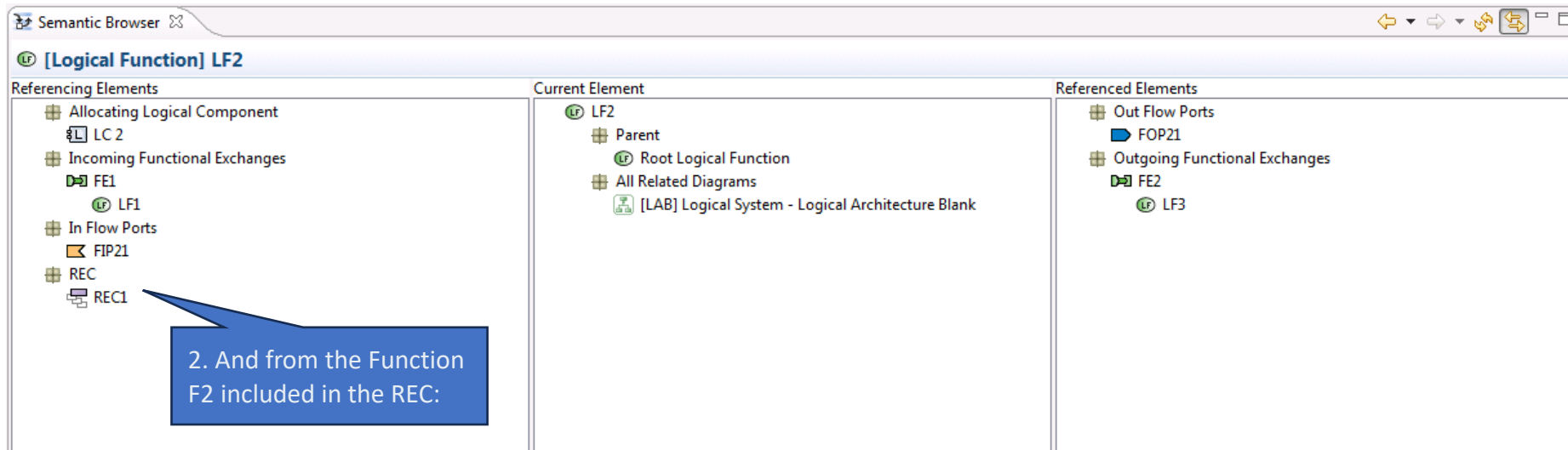
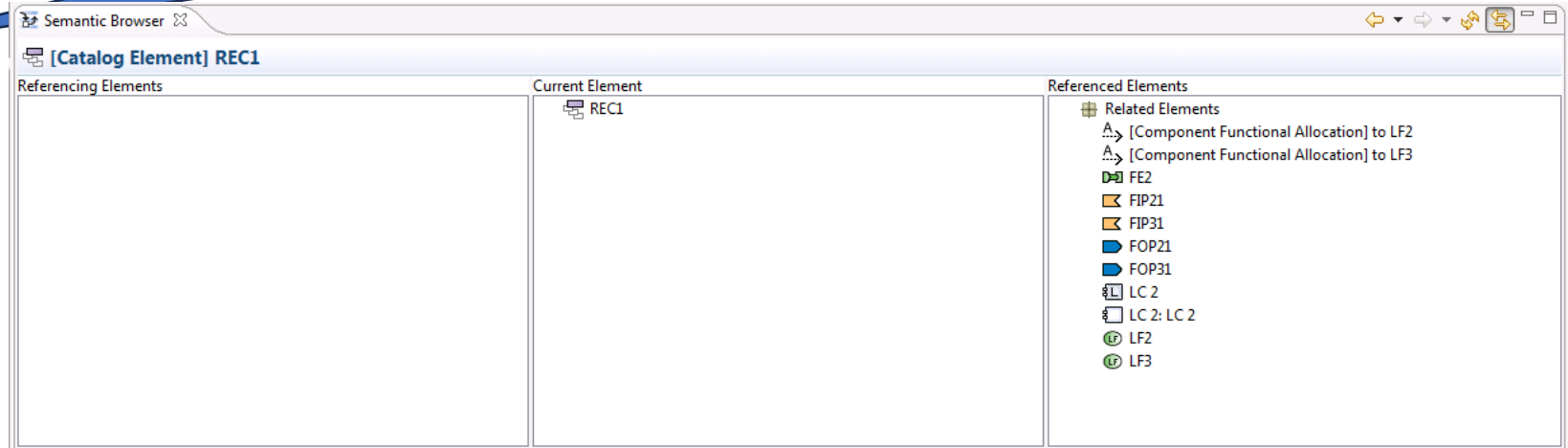
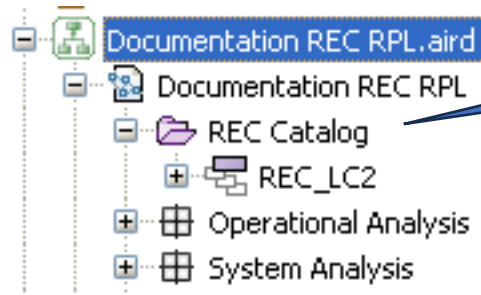
6. It is possible at this stage to define which elements will have to be renamed when the REC will be instantiated (label [+SUFFIX]).

7. Notice the message at the bottom of the dialog, selected elements are linked to some elements which are not included in the REC (many exchange items, visible by clicking on the browse button on the right). When the REC will be instantiated, elements of the newly RPL will be linked to these exchange items too.



CRIANDO...

1. Close the dialog and check the result.
In the Project Explorer, the newly created REC appears. The Semantic Browser also show REC-related information:

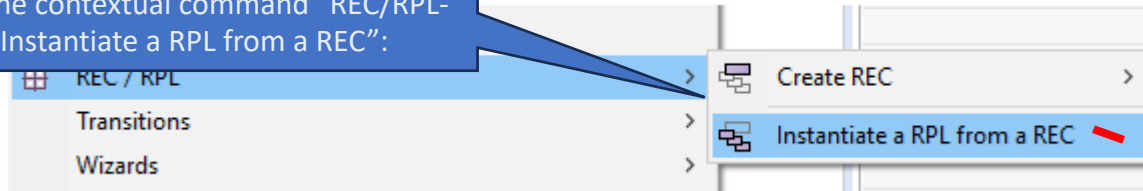


2. And from the Function F2 included in the REC:



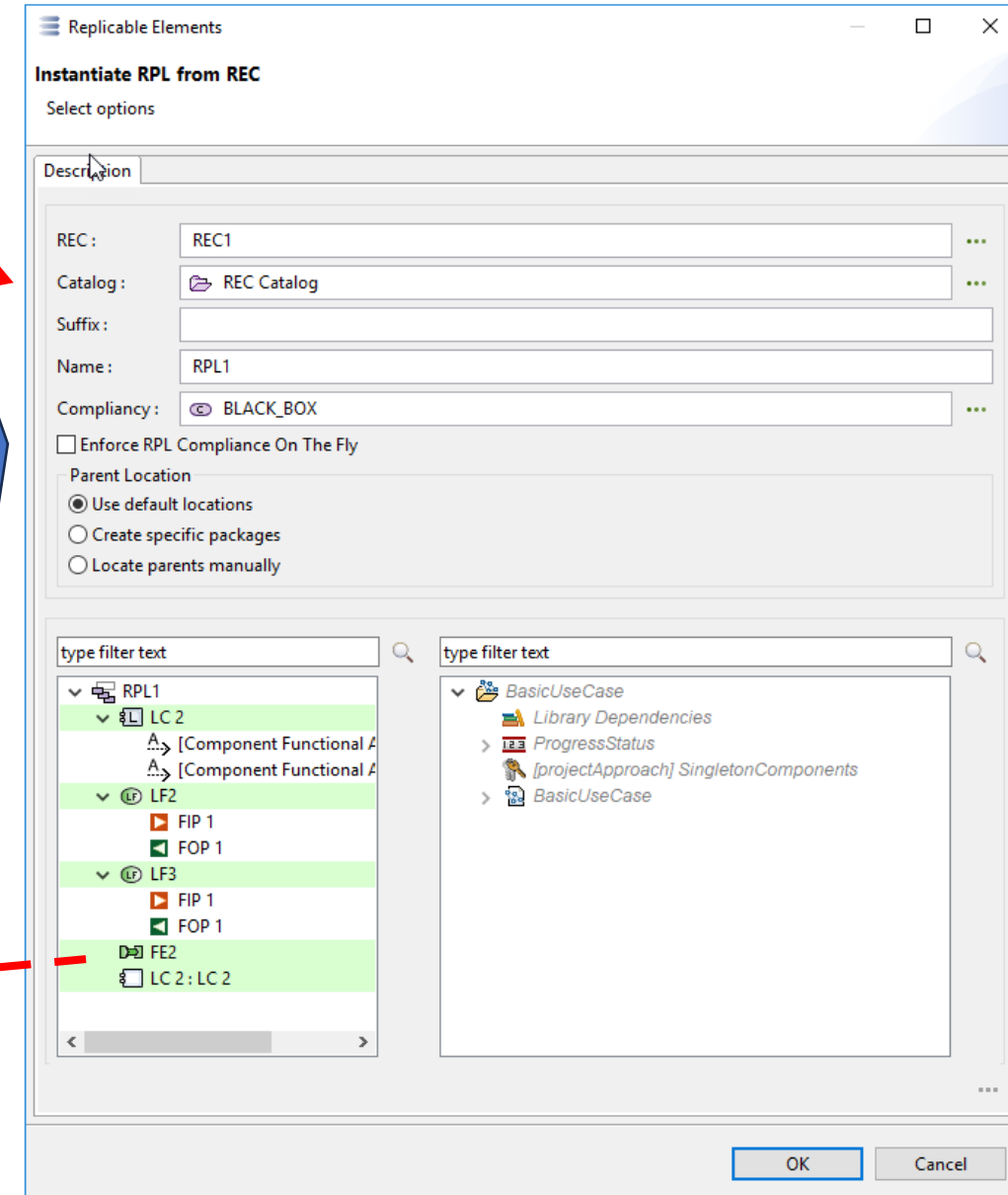
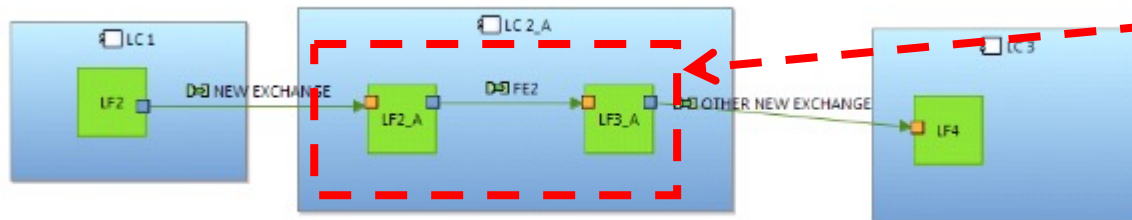
USANDO UMA REC EM UMA RPL

1. From anywhere in the model, use the contextual command “REC/RPL->Instantiate a RPL from a REC”:



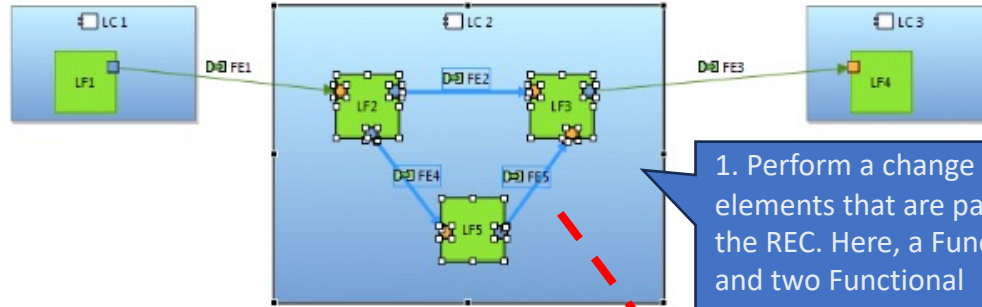
2. This dialog allows:

- The selection of the REC to instantiate (click on “Browse”)
 - The definition of a target container (Catalog) for the RPL going to be created.
 - The definition of a suffix for each element of the REC that was marked as having to be renamed.
 - The compliancy field allows defining how RPL can be modified according to its REC (is it possible to modify content of the replica? add external interfaces? add ports? add more functions?) See the RPL Validation part for further description of any kind of compliancy (**This feature is not fully available yet**)
 - To enable live compliancy validation for this RPL select “Enforce RPL Compliance on the fly”.
- All RPL elements corresponding to a REC element with the suffix tag [+SUFFIX] will have the RPL suffix.
- The parent locator options exist to specify where the RPL elements will be located:
- Use default locations: Elements will be located in standard containers in the model
 - Create specific packages: For each element type, a RPL specific package will be created. Elements of the corresponding type will be stored in that package. Some elements, e.g. Parts do not get a specific package and are located just as if the default locations option would be selected.
 - Locate parents manually: A location has to be found manually for the root elements of the RPL. The elements for which a location still has to be found are marked in Orange. The definition of a new location is performed using drag and drop between the two trees:





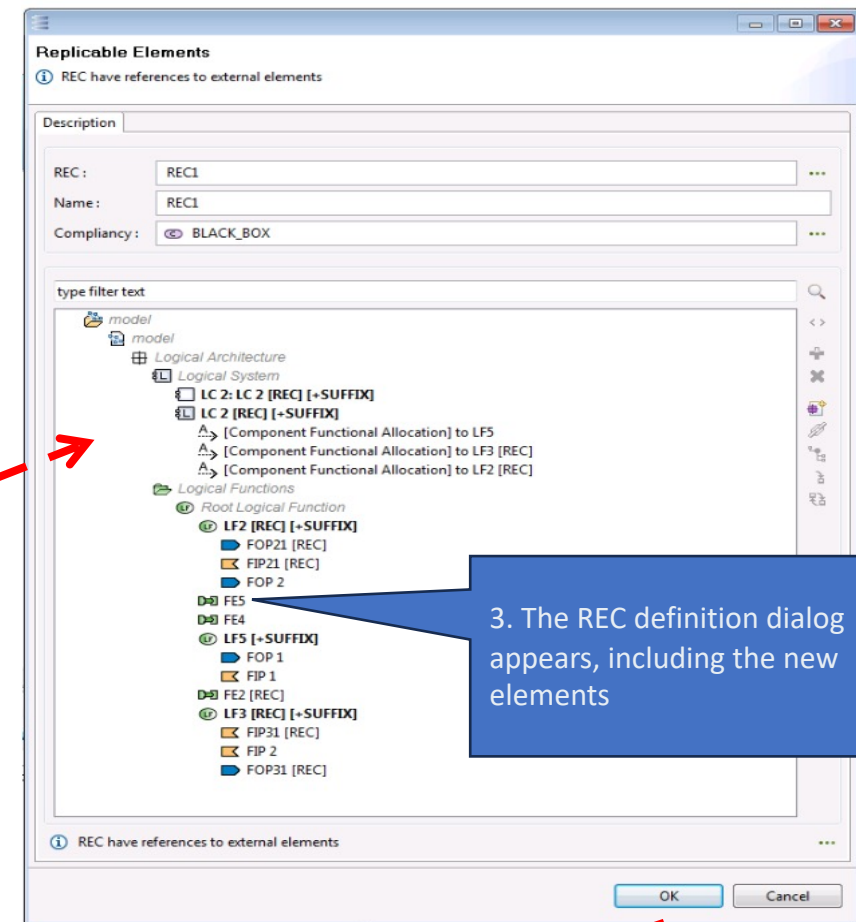
UPDATE REC FROM THE RPL



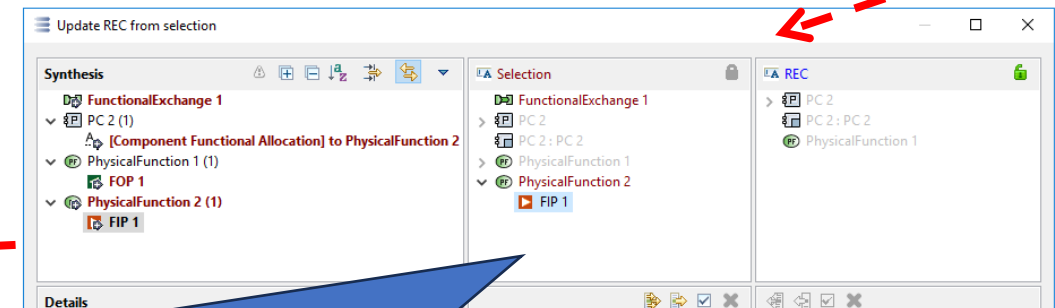
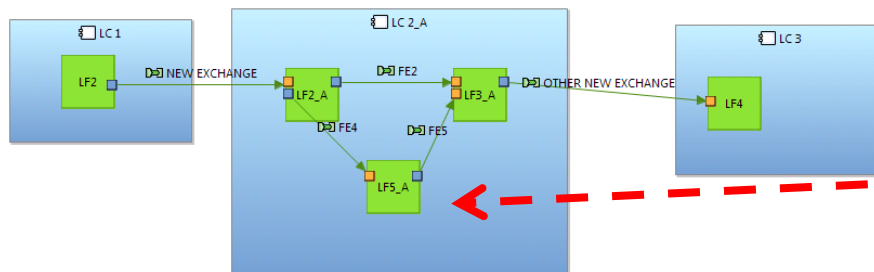
1. Perform a change on elements that are part of the REC. Here, a Function and two Functional Exchanges are added.



2. Select at least one original element of the REC and from the contextual menu, choose 'Update REC from selection'



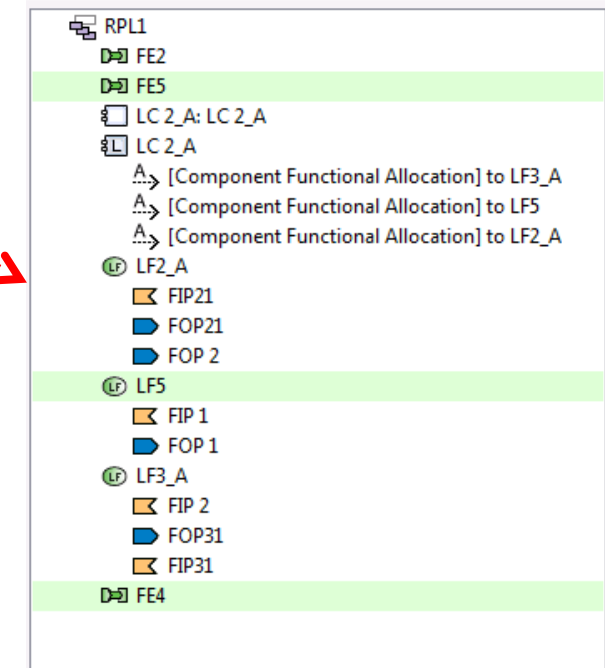
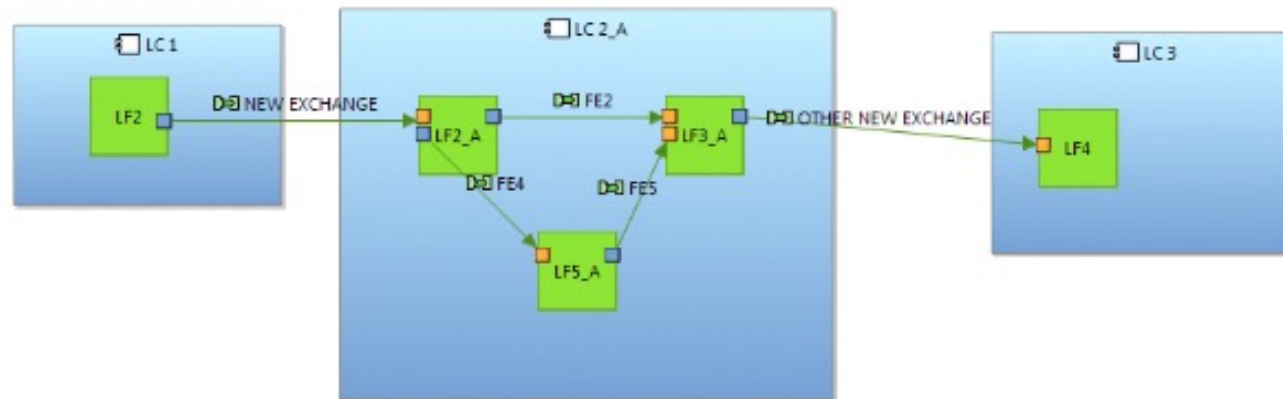
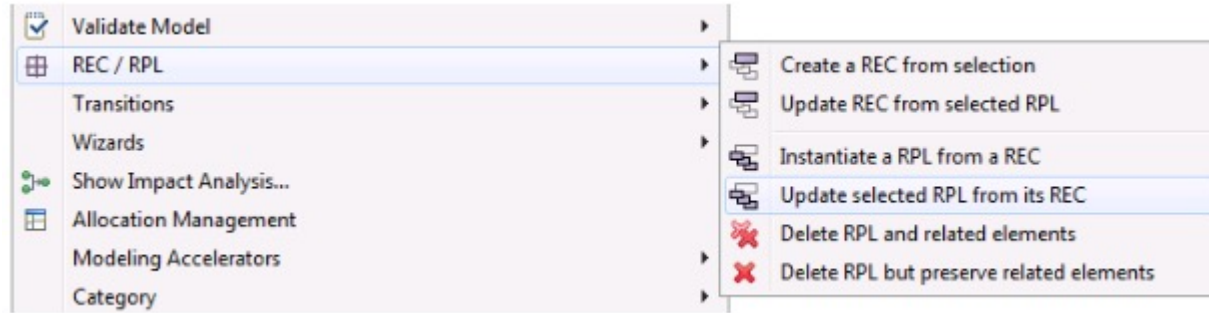
3. The REC definition dialog appears, including the new elements



4. Click on OK. This will open the DiffMerge view. Press 'Apply' to update the rec to include all additional changes to the REC (additional information about this dialog is available in the Model DiffMerge section) .



UPDATE RPL FROM THE REC



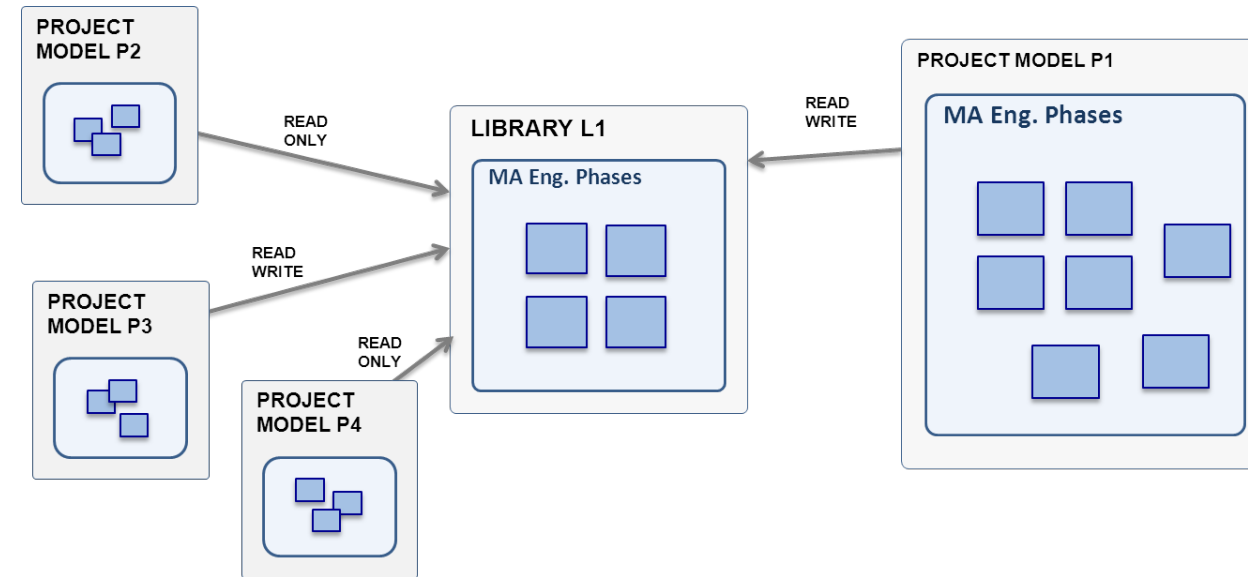


BIBLIOTECAS



DEFINIÇÃO

- Biblioteca é um modelo Capella destinado a ser compartilhado entre vários projetos.
- Um projeto pode fazer referência a uma biblioteca com READ ou READ/WRITE. Neste último caso, isso significa que o conteúdo da Biblioteca pode ser modificado a partir do próprio Projeto, sem ter que abrir especificamente a Biblioteca.
- Uma biblioteca pode ter referências a outras bibliotecas, mas uma biblioteca não pode ter uma referência a um projeto.





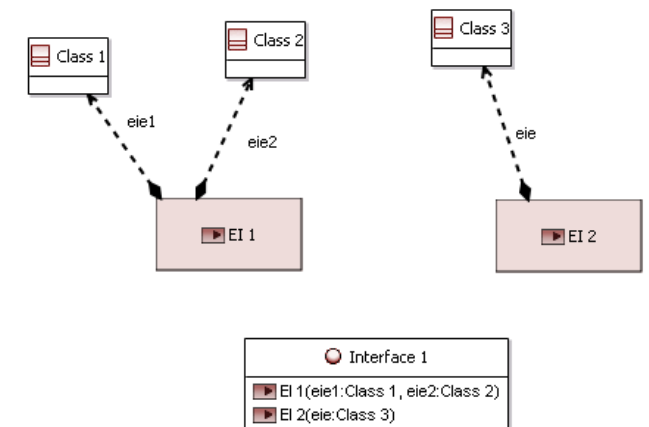
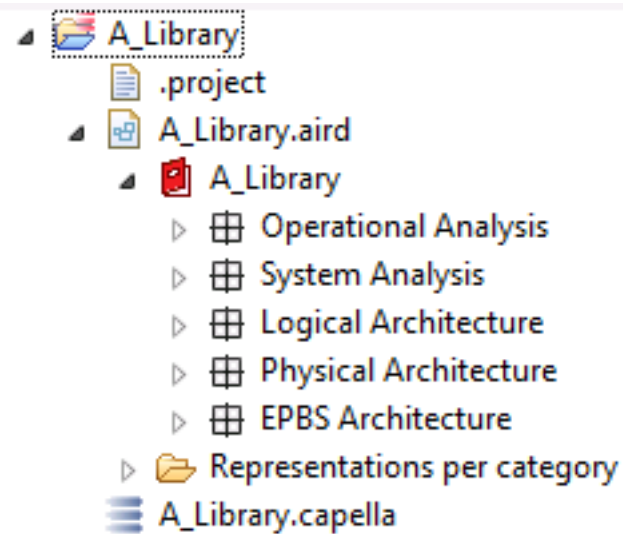
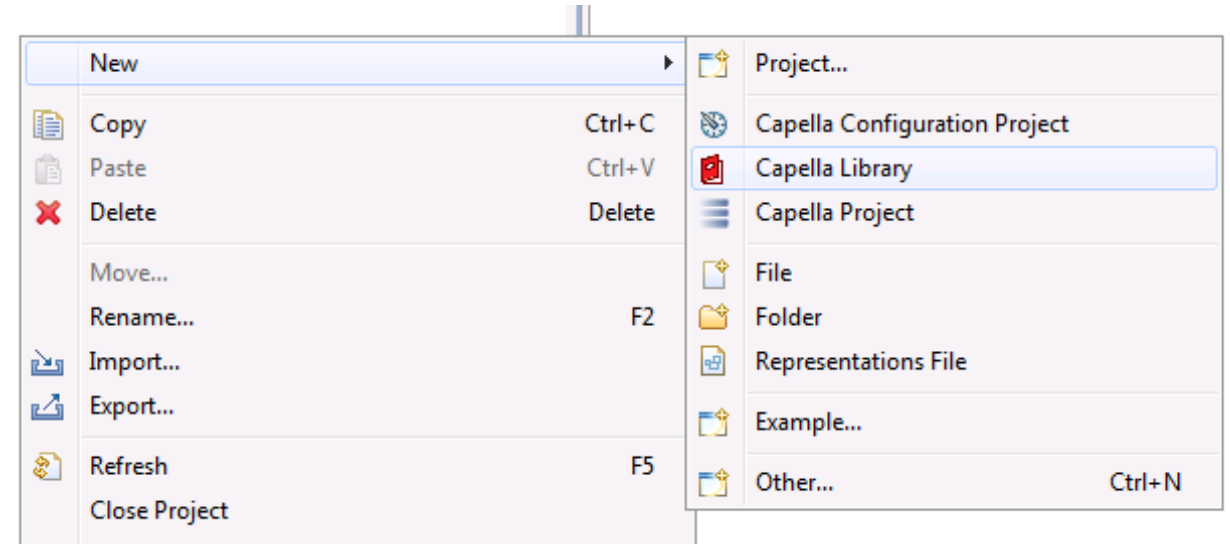
A QUE SE DESTINAM AS BIBLIOTECAS?

- Permitir a reutilização de elementos de modelo em modelos diferentes (por exemplo, vários projetos em um domínio geralmente precisam compartilhar o mesmo modelo de dados).
- Melhorar a organização (evitar duplicação e referências entre modelos)
- Catálogos de elementos replicáveis
- As bibliotecas se beneficiam das mesmas ferramentas que os modelos.
 - Edição do conteúdo da biblioteca através de diagramas e editores
 - Navegador semântico
 - Regras de validação e correções rápidas



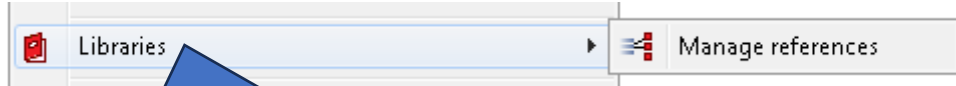
CRIAÇÃO DE BIBLIOTECAS

- As bibliotecas são criadas da mesma forma que os projetos Capella padrão.
- Do Project Explorer, criar uma nova biblioteca usando o menu contextual

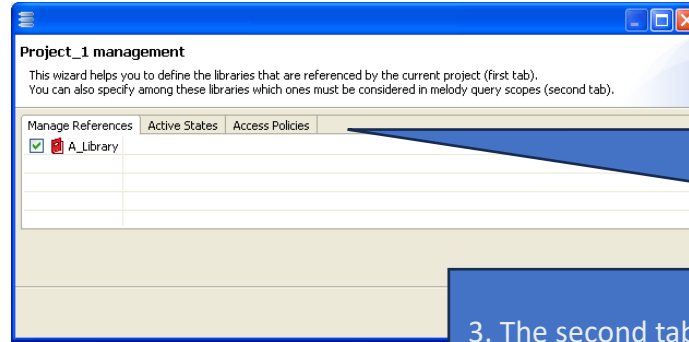




REFERENCIANDO UMA BIBLIOTECA

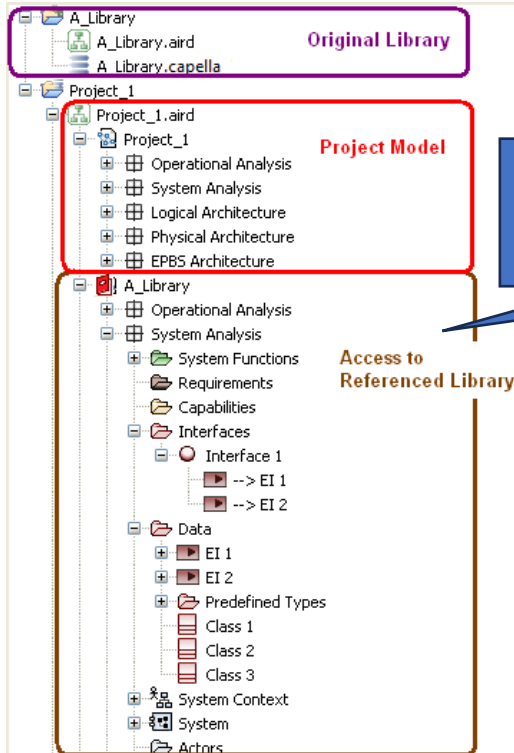
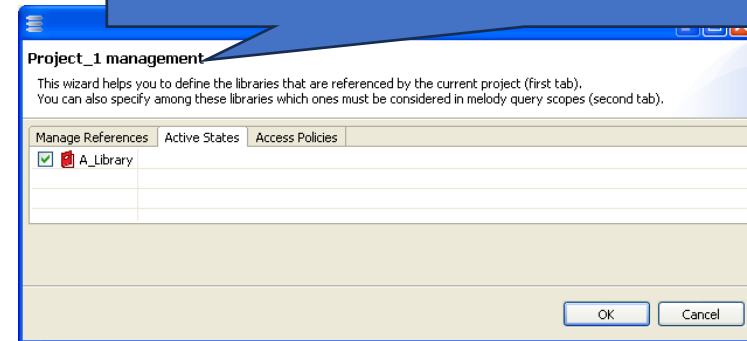


1. Select the “Libraries | Manage References” item in the contextual menu on the “aird” file of a standard Project Model.

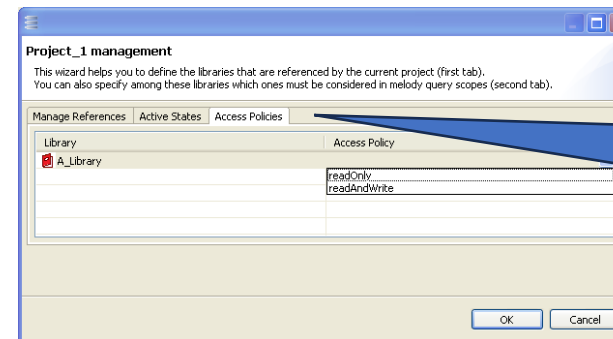


2. The first tab of this dialog displays all the accessible Libraries in the current workspace (A Library in a closed Eclipse Project will not be proposed).

3. The second tab of the dialog displays which Library is currently active. When a Library is not active, queries in Editors for example will not display the content located in the Library.



5. Once the Project Model is opened, the referenced Library can be seen directly from the Project itself.

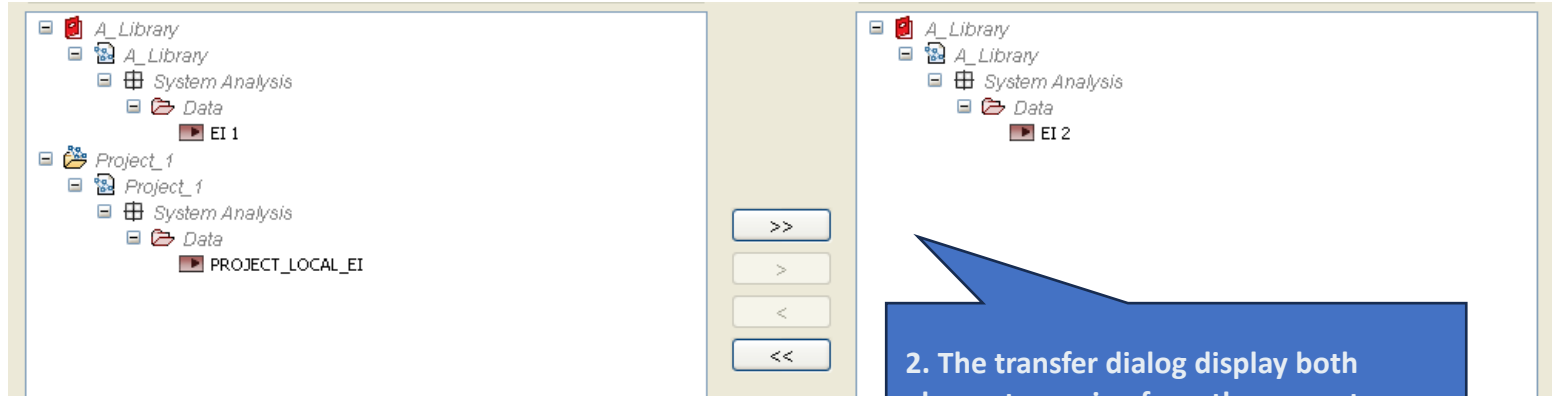


4. The third tab allows specifying whether the content of a referenced Library can be directly modified from the Project itself. The default is “Read only”.



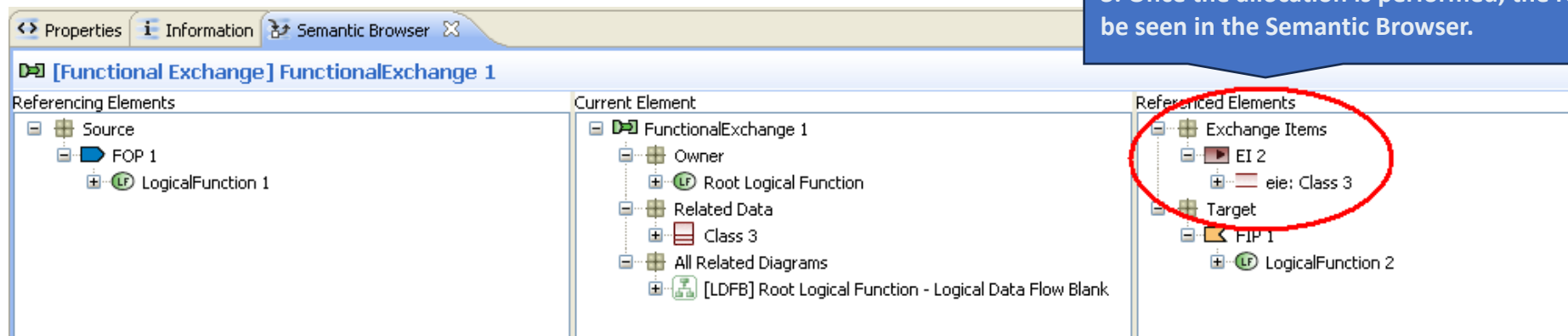
USANDO: ALLOCATION OF EXCHANGE ITEMS TO A FUNCTIONAL EXCHANGE

1. Example with the allocation of Exchange Items to a Functional Exchange



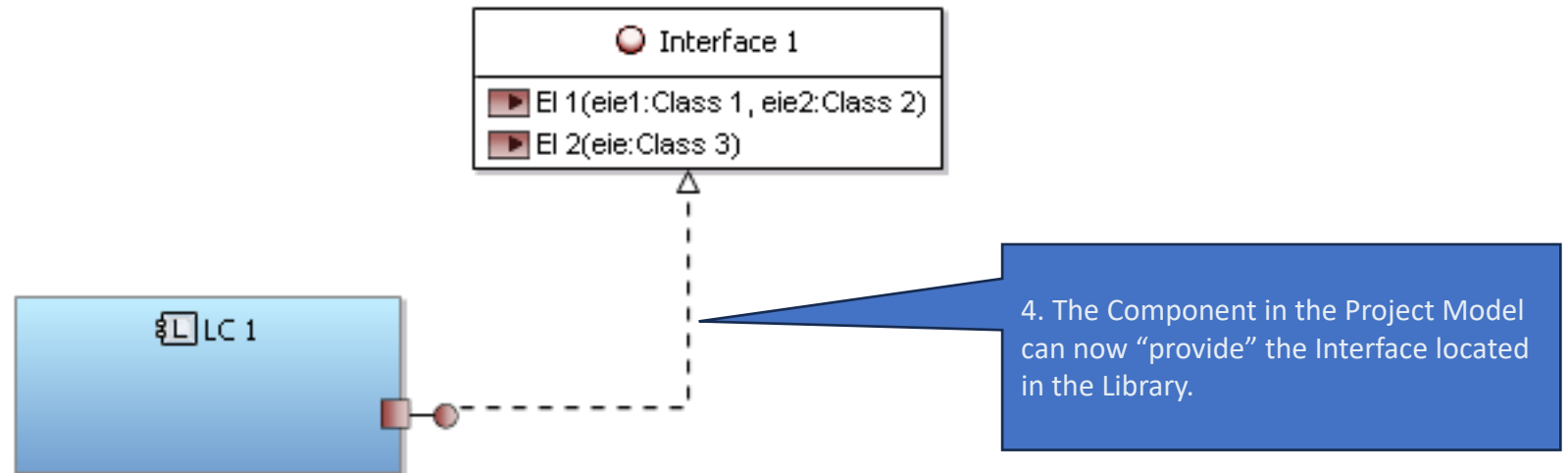
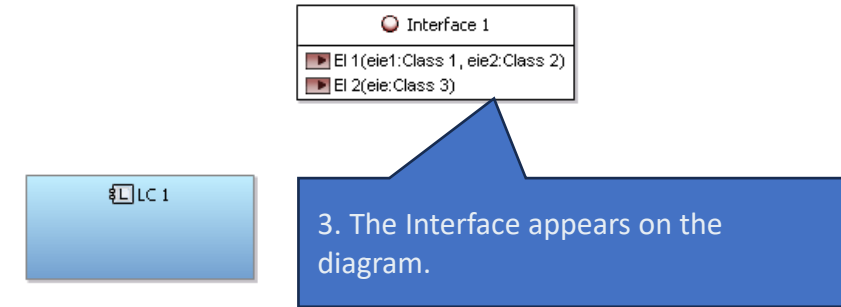
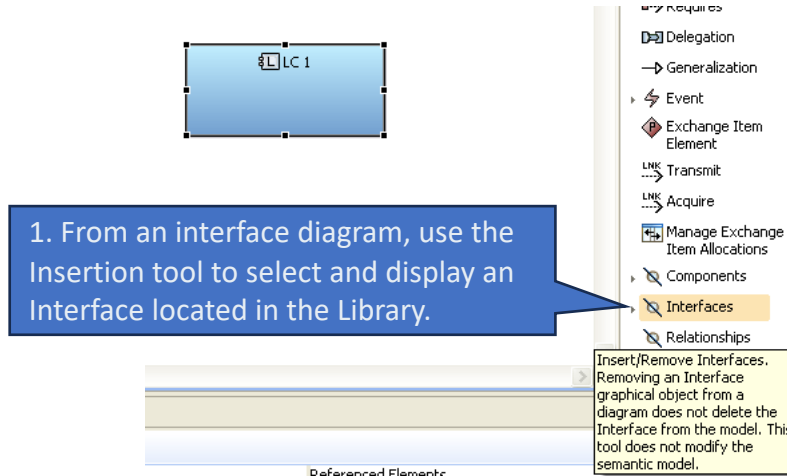
2. The transfer dialog display both elements coming from the current Project and elements from the Library.

3. Once the allocation is performed, the result can be seen in the Semantic Browser.



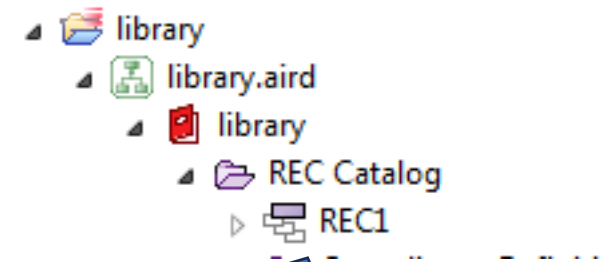
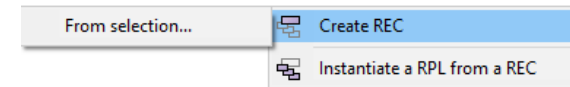
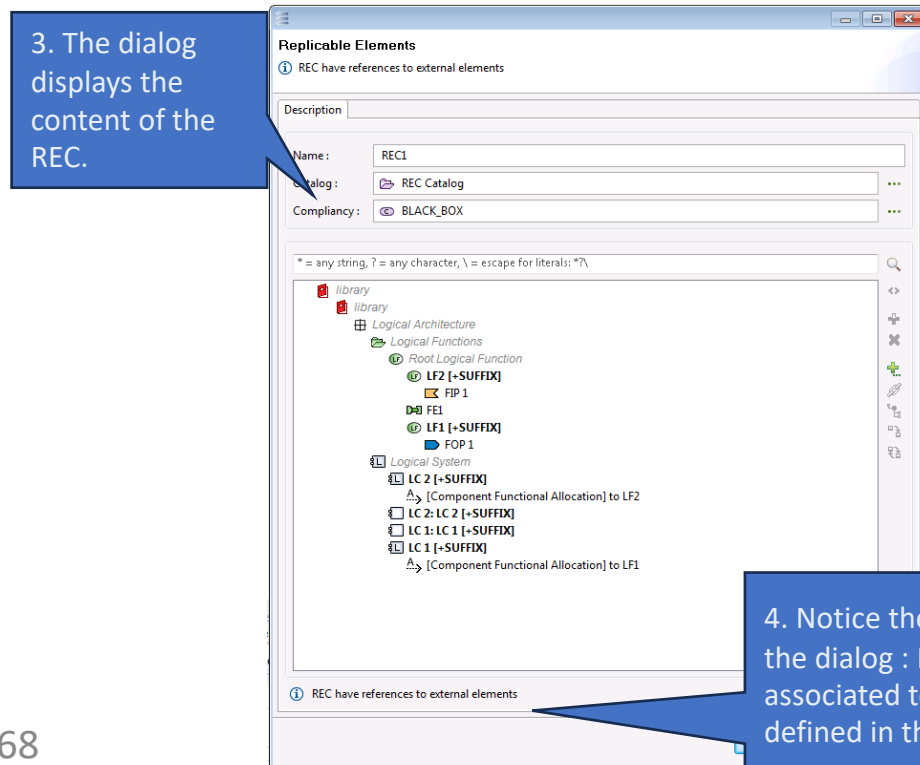
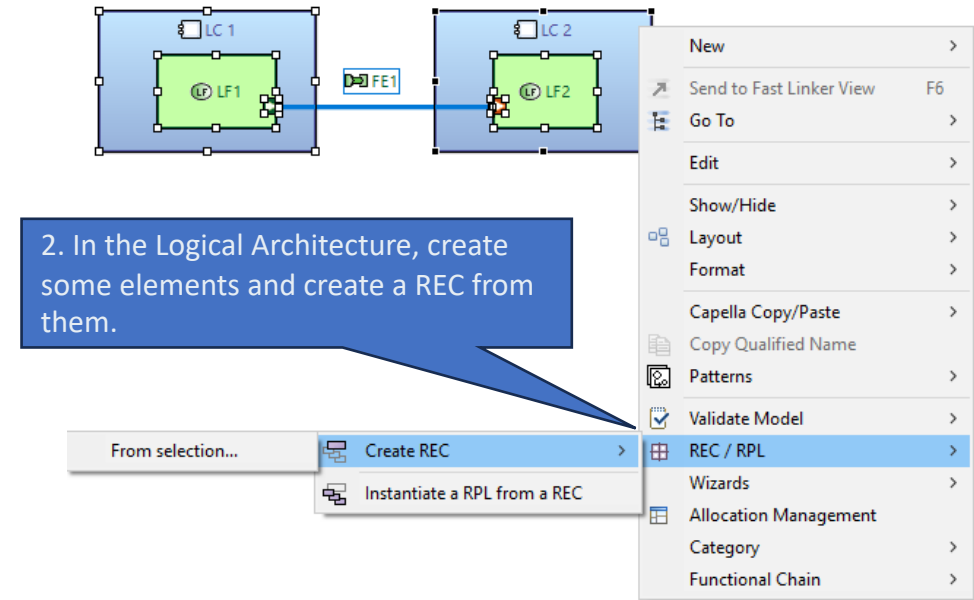
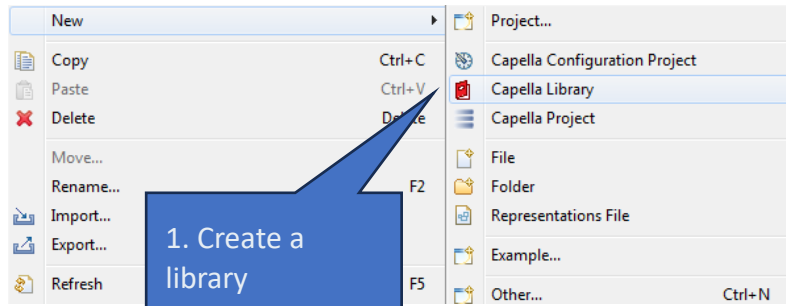


USANDO: COMPONENTS AND INTERFACES





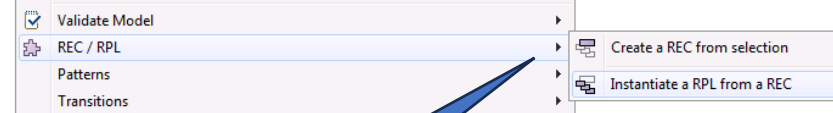
USANDO COM RPL/REC



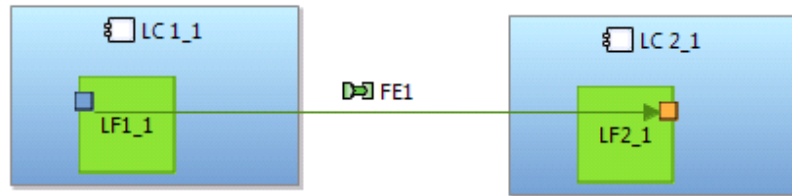
4. Notice the message at the end of the dialog : Functional Exchange is associated to some Exchange Items defined in the library:



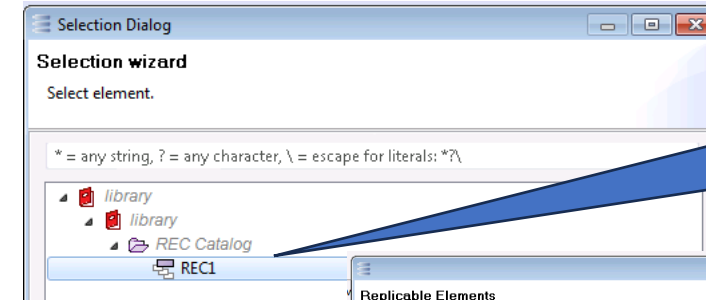
ADICIONANDO A RPL DA LIB NO PROJETO



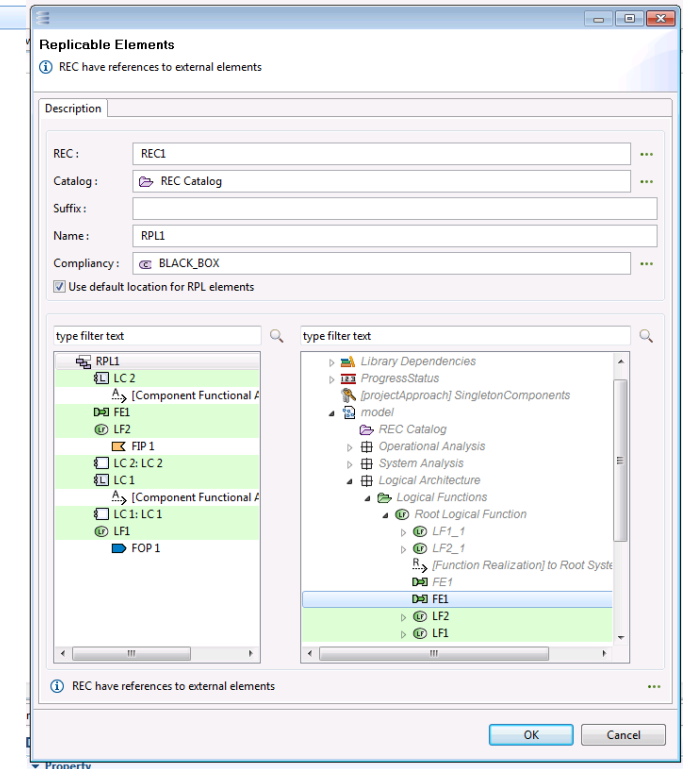
1. In a Logical Architecture Blank diagram, instantiate a new RPL



2. Display all elements of the RPL in the diagram



3. In the dialog, select the REC located in the referenced library:





REFERÊNCIAS SOBRE REC->RPL / LIBS

- **[HOW TO] Replicate model elements in Capella (4'25'')**
- <https://www.youtube.com/watch?v=h-ax61eVlxM>
- **Webinar - Strategies and tools for model reuse with Capella (58'23'')**
- <https://www.youtube.com/watch?v=l28EhAXe-i8>
- **In-Flight Entertainment System (IFE) – Example**
- <https://download.eclipse.org/capella/samples/1.3.1/InFlightEntertainmentSystem.zip>
- **Capella Help – Replicable Elements**



CONSIDERAÇÕES FINAIS



CONSIDERAÇÕES

- Respostas da Arquitetura Funcional:
 - ***"Como o sistema funcionará para atender às expectativas"***
- Na arquitetura funcional desdobramos o sistema em agregadores funcionais = componentes
- Temos que decidir como organizar as funções e os componentes
- Mapeia as funções internas do sistema.



ATIVIDADES PARA A PRÓXIMA AULA

- Voltamos para o nosso exercício:



PROPOSTA DE MISSÃO

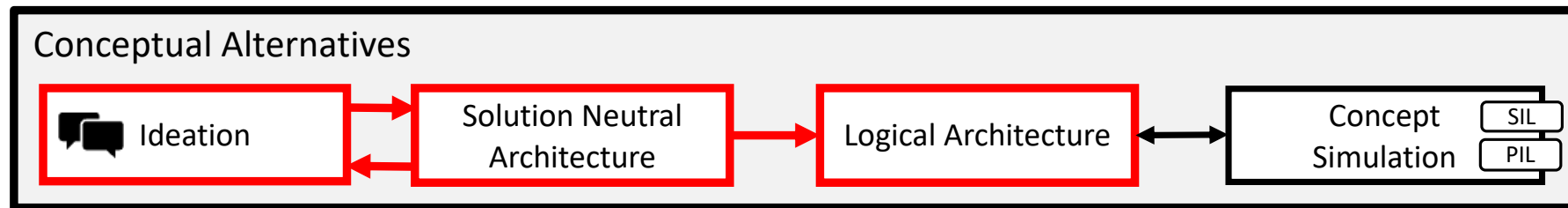
1. *Os viajantes do tempo Doc Emmett Brown e Marty McFly não podem revelar a existência de máquinas do tempo pois isso geraria um conflito pela posse da tecnologia.*
2. *Ações específicas na linha do tempo podem ocasionar infinitas linhas temporais acarretando no fim da estrutura do universo.*
3. *Desta forma, eles precisam de meios para estruturar um **sistema logístico seguro** de peças para uma montagem/manutenção ágil das máquinas do tempo, em cada uma das épocas, apoiado por ações de **inteligência, vigilância e reconhecimento**.*
4. *É preciso que se **monitorem os acontecimentos do entorno** do local de desenvolvimento, movimentação de pessoas, eventos, e outros fluxos para criar uma previsão de acontecimentos.*
5. *Deve ser **feito com os recursos disponíveis** em cada uma das épocas, de forma a não levantar suspeitas e colaborar com a ocultação do transito de peças.*





ATIVIDADES PARA A PRÓXIMA AULA

- Fazer a etapa da formalização do modelo funcional
- Apresentar como o sistema logístico vai atender à demanda.
- Apresentar o modelo da arquitetura funcional:
 - Características mínimas: desdobrar em 3 subcomponentes, das funções de fronteira quebrar/juntar em no mínimo 10 subfunções, mostrar análise de coesão-acoplamento dos subcomponentes, fazer a máquina de estado de cada subcomponente, fazer o diagrama de interfaces internas, escrever 10 requisitos (8 funcionais e 2 não funcionais) desdobrados dos requisitos da intervenção sistêmica.



→ ROP

→ RTLI