



REVISÃO

- Pensamento Sistêmico: “uma disciplina para ver o todo... um framework para ver interrelações em vez de partes ... um processo de descoberta e diagnóstico ... e uma sensibilidade para a sutil interconexão que dá aos sistemas seu caráter único”
- Um sistema é $\langle \rangle$ de $\langle \rangle$, que $\langle \rangle$ de uma forma $\langle \rangle$ que não $\langle \rangle$ por suas $\langle \rangle \langle \rangle$.
- Princípio da Emergencia: Quando as entidades do sistemas são conectadas, suas interações vão causar funções, comportamentos, performances e outras propriedades que irão emergir.
- Engenharia de Sistemas são meios e abordagens transdisciplinares, baseados em conceitos e principios sistemicos, para permitir uma realização bem sucedida, uso e descarte de sistemas engenheirados



IEA-P – DEPARTAMENTO DE PROJETOS
(PROJECT DEPARTMENT)

Representações Clássicas

[2024]

Prof. Dr. Christopher S. Cerqueira



	SEMANA	TEORIA	INDIVIDUAL	PESO	GRUPO	PESO
	1	Estrutura do Curso e Apresentações Pessoais				
	04-Mar	O que é Engenharia de Sistemas? INCOSE	AI-01 - Leitura/Resumo Cap 1 - HB INCOSE	5%	AG-01 - Entrega dos grupos.	0%
	08-Mar	Contexto: Aeronave x Sistemas do Dom. Aéreo				
	2	Gramática: Representações clássicas				
	11-Mar		AI-02 - Leitura/Resumo paper sobre representações clássicas.	5%	AG-02 -	0%
	15-Mar	Lógica: Diagrama de Blocos				
	3	Gramática: Arquitetura e Visões				
	18-Mar	Gramática: Funções	AI-03 - Exercícios sobre arquitetura funcional e diagrama de blocos.	5%	AG-03 - Resumo sobre arquitetura funcional.	10%
	22-Mar	Lógica: Diagrama de Classes				
	4*	Gramática: Stakeholders				
	25-Mar	Gramática: Ciclo de Vida e CONOPS	AI-04 - Exercícios sobre diagrama de casos de uso	5%	AG-04 - Resumo sobre stk, ciclo de vida e CONOPS	10%
	29-Mar	Lógica: Diagrama de Casos de Uso				
	5	Gramática: Requisitos				
	01-Apr	Gramática: Verificação e Validação	AI-05 - Exercício de escrita de requisitos.	10%	AG-05 - Resumo sobre requisitos vindos da parte de segurança (safety) do RBAC.	10%
	05-Apr	Contexto: ANAC (RBACs)				
	6	Lógica: Diagrama de Estados				
	08-Apr		AI-06 - Exercício sobre diagrama de estados.	10%	AG-06 - Revisita do CONOPs do DECEA usando storyboards	15%
	12-Apr	Contexto: DECEA (ICAs/DCAs)				
	7	Lógica: Diagrama de Sequência				
	15-Apr		AI-07 - Exercício sobre diagrama de sequencia.	0%	AG-07 - Representação do CONOPs usando Diagrama de Sequencia.	20%
	19-Apr	Temas do projeto do segundo semestre e P1				
	8	P1				
	22-Apr		AI-08(P1) - Teoria e linguagem (Pres/Consulta - sem chatGPT)	60%	AG-01(P1) - Apresentação Gravada: 20min	40%
	26-Apr					
				100%		105%



OBJETIVO DA AULA DE HOJE

- Introduzir os diagramas clássicos do linguajar da Engenharia de Sistemas.
 - *Vamos aprender a formar as letras e sílabas antes de formar as frases.*
 - *EFFBD*
 - *DFD*
 - *N2*
 - *IDEFO*
 - *Diagrama de Blocos*

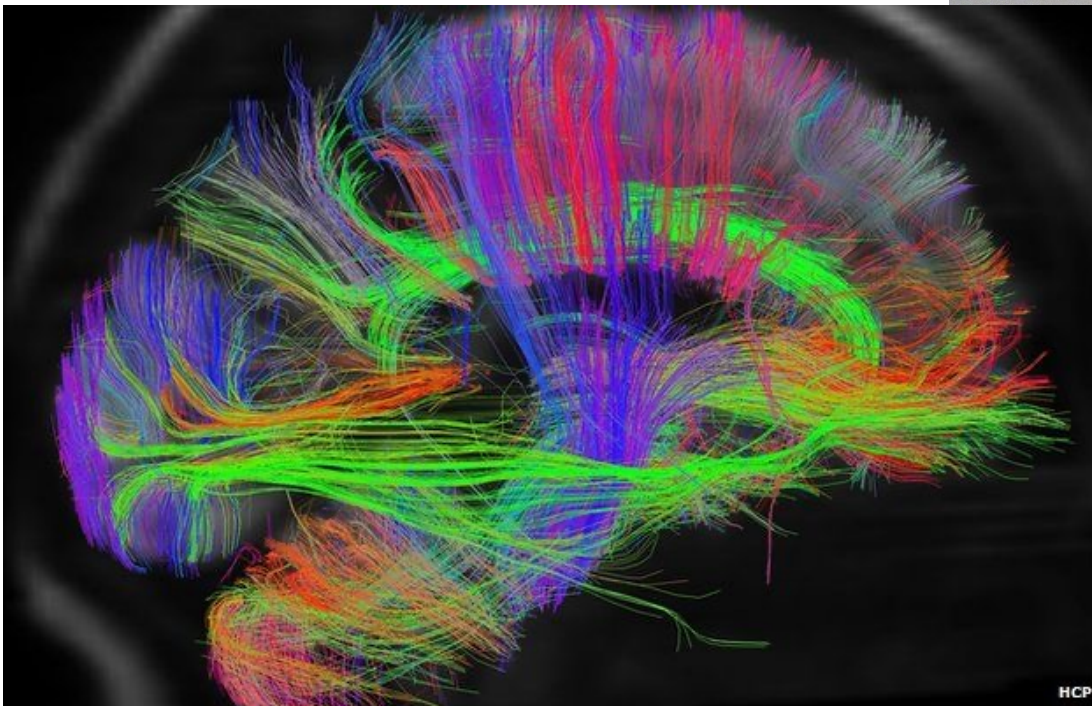


LINGUAGEM DA ENGENHARIA DE SISTEMAS



REDUZINDO A FUNÇÃO: NÓS SOMOS RECONHECEDORES DE PADRÕES

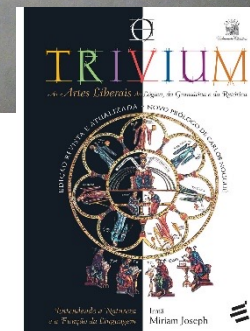
<https://www.psychologytoday.com/blog/the-athletes-way/201311/what-is-the-human-connectome-project-why-should-you-care>



FUNÇÃO DA GRAMÁTICA

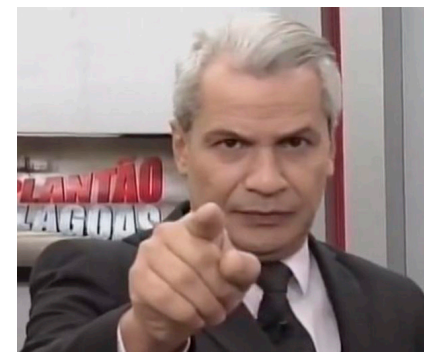
A função fundamental da gramática é estabelecer leis para relacionar símbolos de modo a expressar pensamento. Uma frase expressa um pensamento – uma relação de ideias – numa declaração, numa pergunta, numa ordem, num desejo, numa prece ou numa exclamação. Símbolos categoremáticos são aqueles que são relacionados; símbolos sincategoremáticos são os meios de relacioná-los; a oração é a relação mesma.

As regras para relacionar símbolos regem três operações gramaticais: substituir símbolos equivalentes, combinar símbolos e separar símbolos.





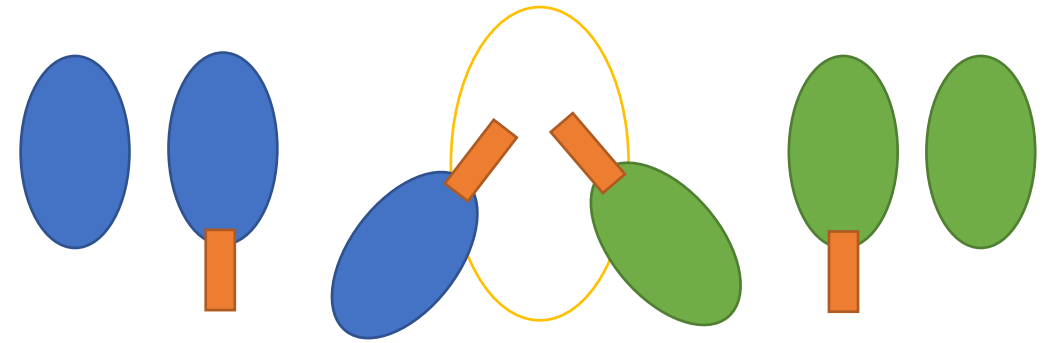
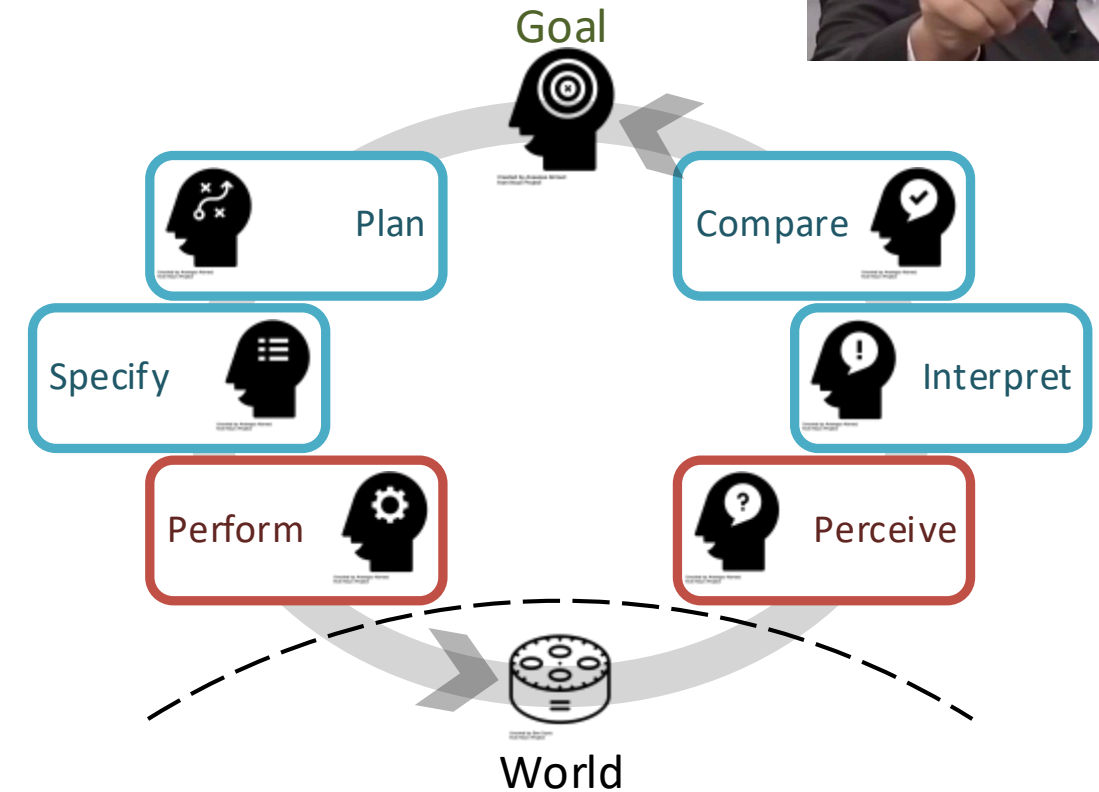
COGNIÇÃO



Cognição é “o processo mental de adquirir conhecimento e entendimento através do pensamento, inteligência e sentidos”.

Engloba processos como a atenção, a formação do conhecimento, a memória e a memória de trabalho, o julgamento e a avaliação, o raciocínio e a "computação", a resolução de problemas e a tomada de decisões, a compreensão e a produção da linguagem.

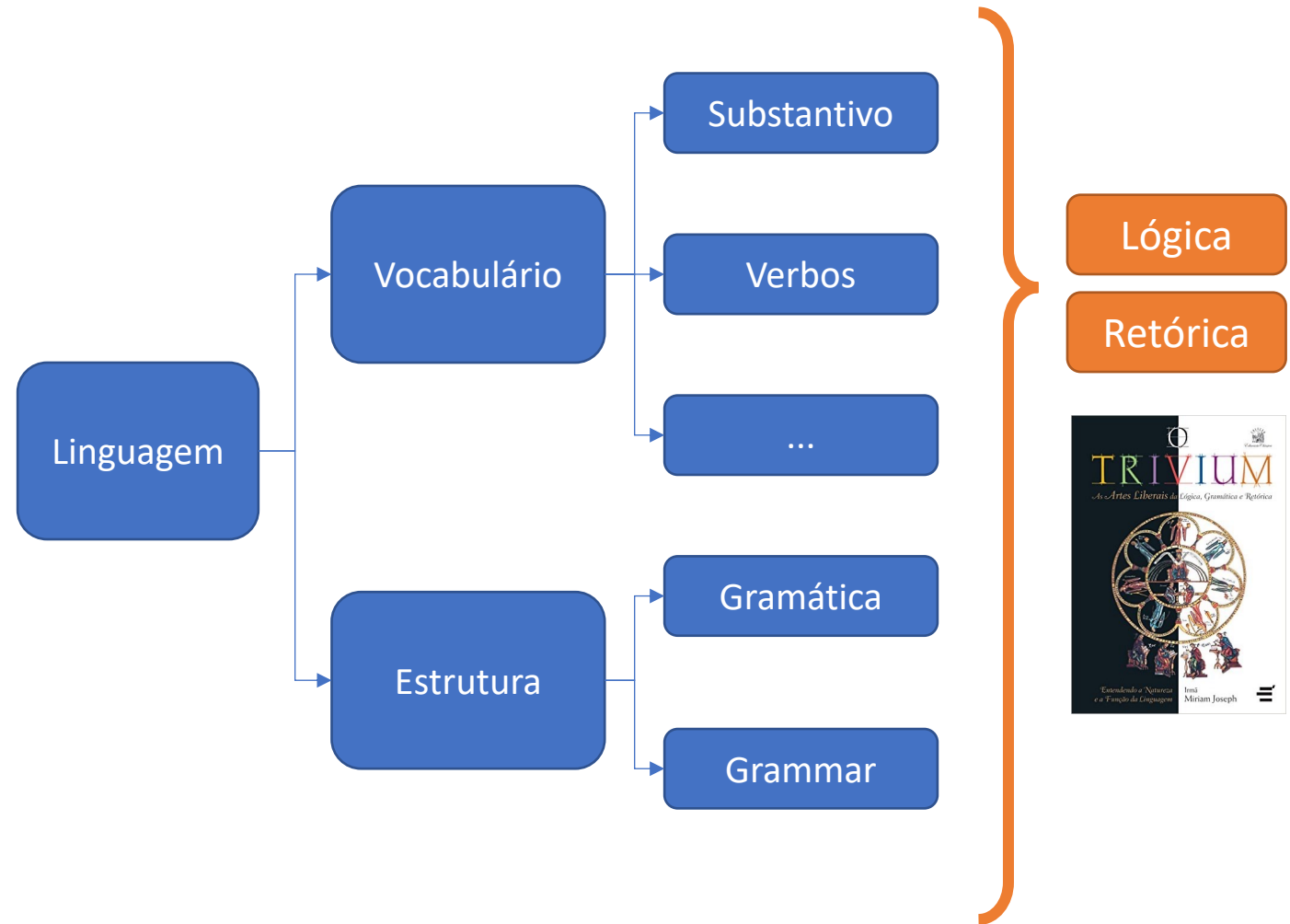
Processos cognitivos usam conhecimento pregresso para gerar novo conhecimento.

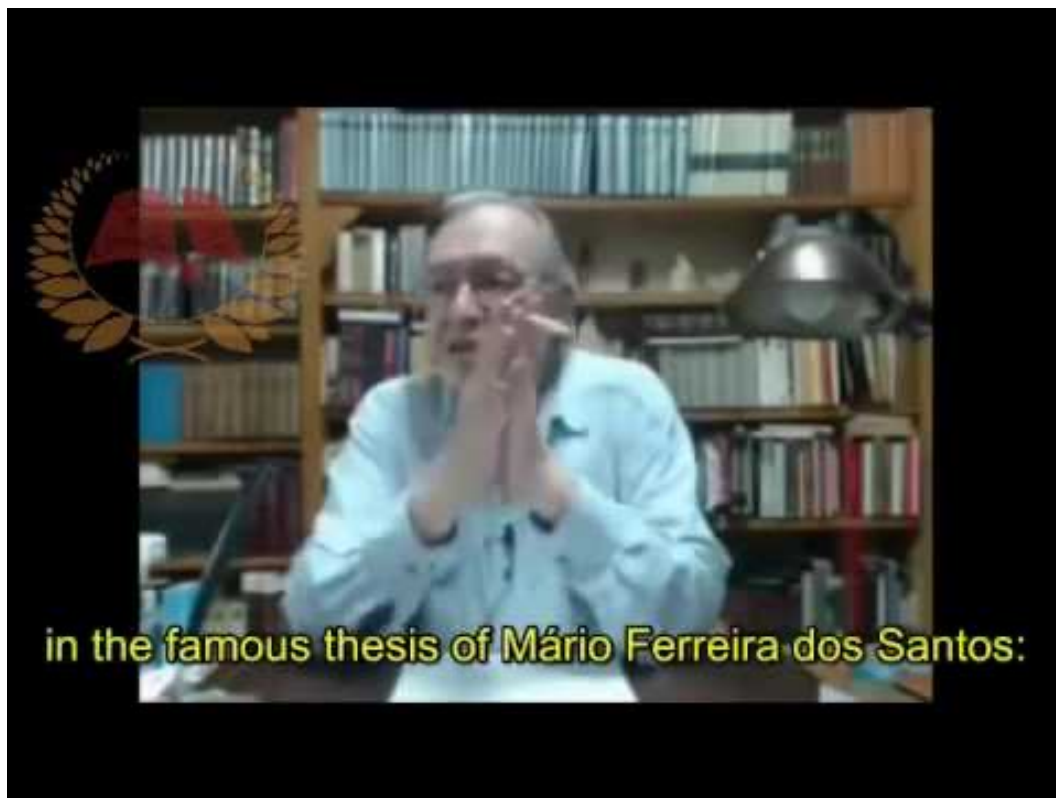




LINGUAGEM

é um sistema que consiste no desenvolvimento, aquisição, manutenção e utilização de sistemas complexos para comunicação.





<http://www.olavodecarvalho.org/apostilas/presenca.htm>

6:15



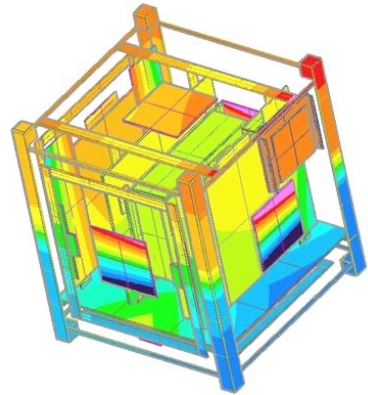
https://www.ted.com/talks/david_mccandless_the_beauty_of_data_visualization#

Linguagem/representações são ferramentas mentais que mapeiam o conhecimento em modelos com a linguagem conhecida. “Usuários” manuseiam “signos” que já são imediatamente reconhecidos (já existindo no seu modelo mental).

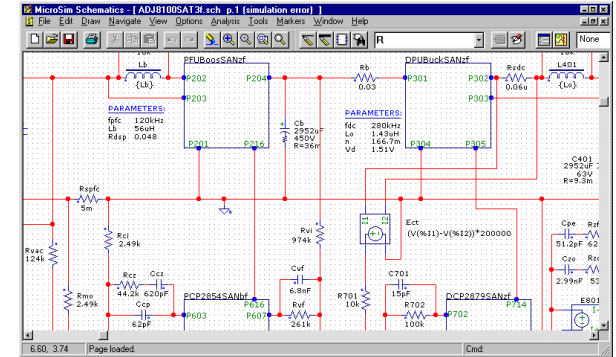


CADA ENGENHARIA POSSUI SUA LINGUAGEM

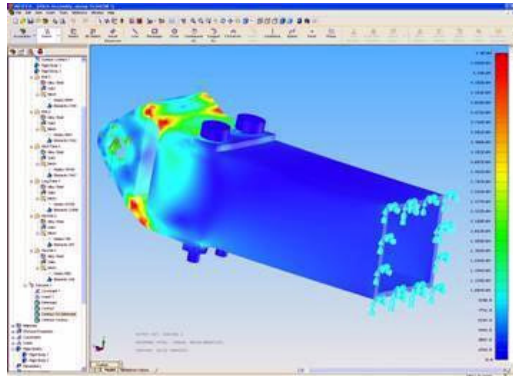
Thermal Eng.



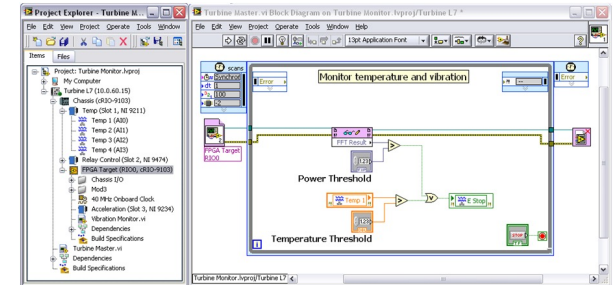
Electrical Eng.



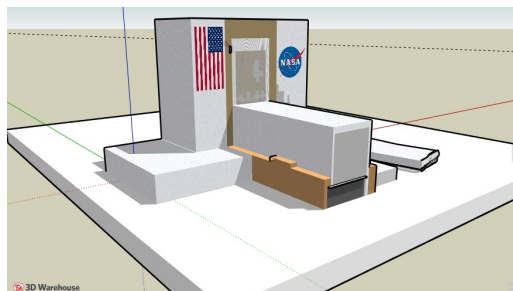
Mechanical Eng.



Control Eng.



Infrastructure Eng.

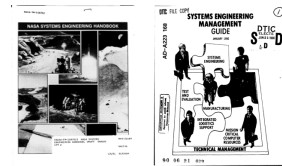


Systems Eng ???

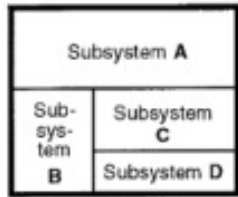




QUAL A LINGUAGEM DA ENG. SIS.?

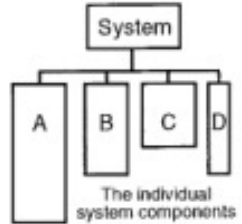


The whole **does more** than the sum of the parts.

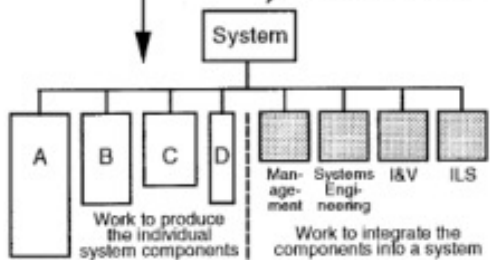


System
Components (subsystems) held together by "glue" (integration)

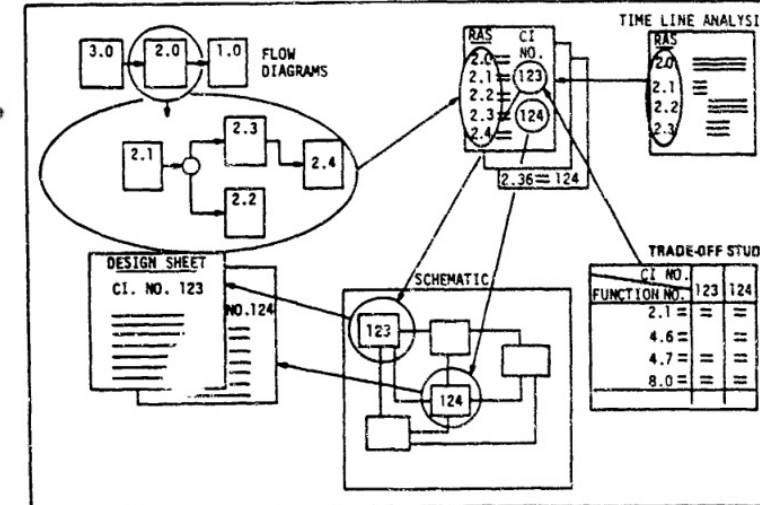
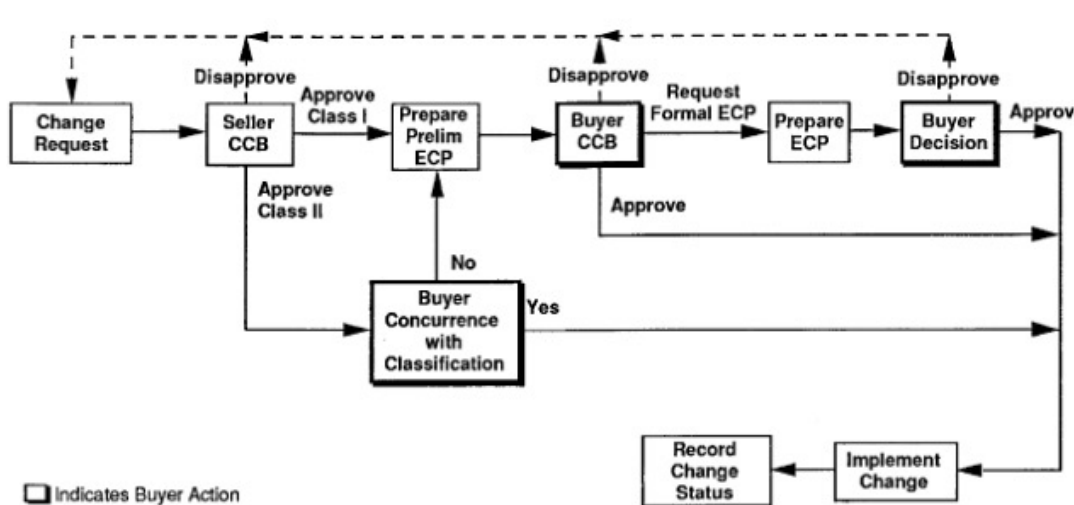
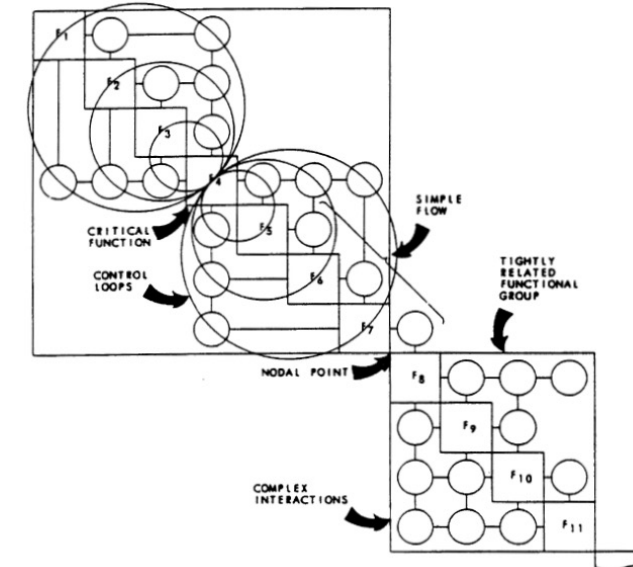
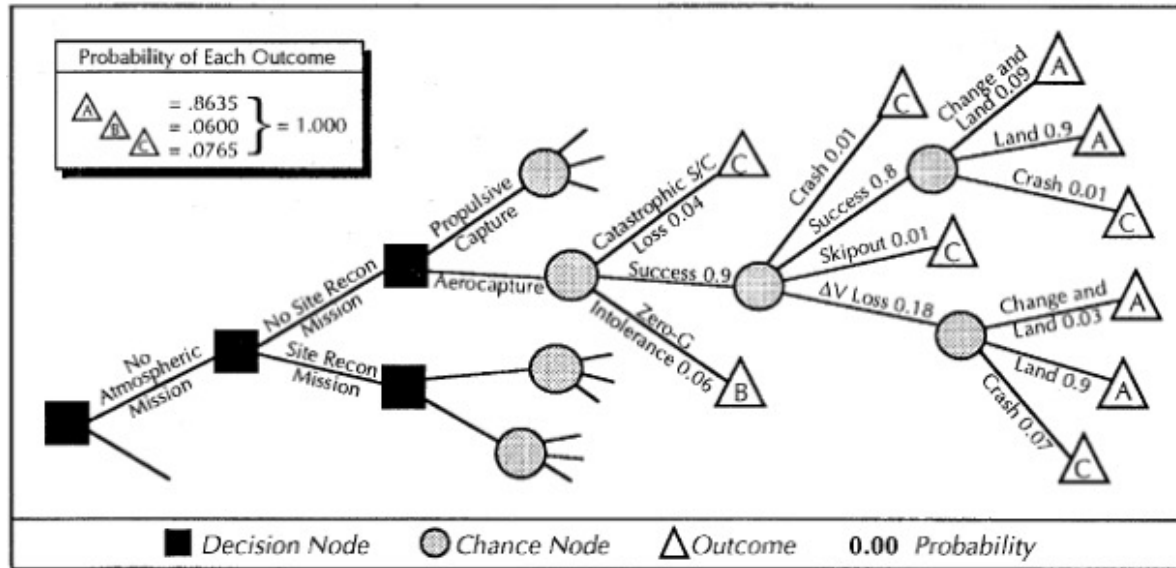
Product Breakdown Structure (PBS)
Shows the components which form the system.



Work Breakdown Structure (WBS)
All work components necessary to produce a complete system



The whole **takes more work** than the sum of the parts.





COM O TEMPO... FICARAM COLORIDOS

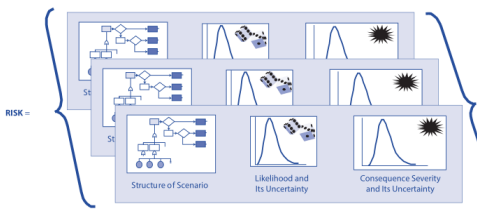
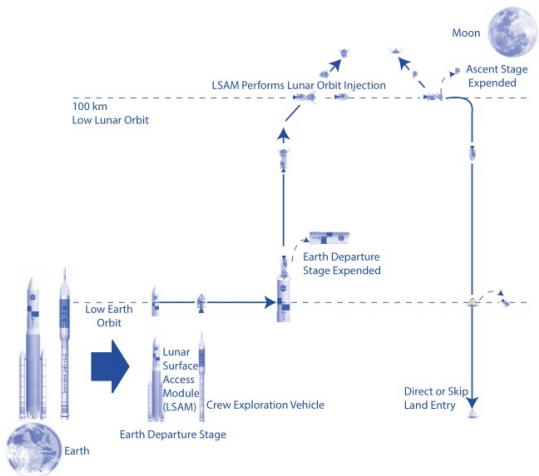
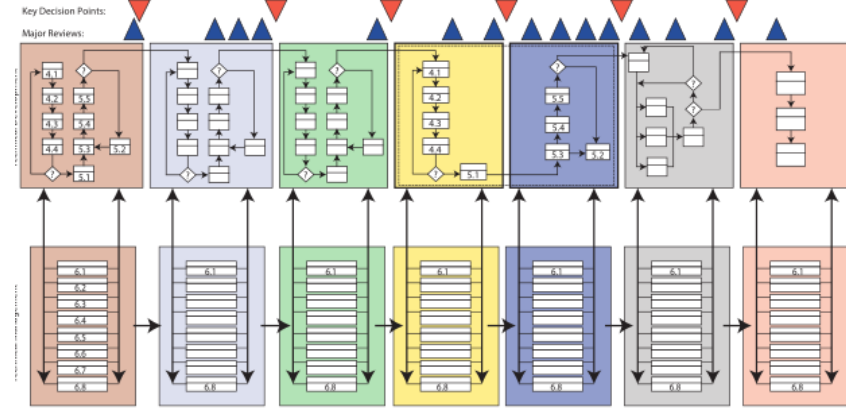
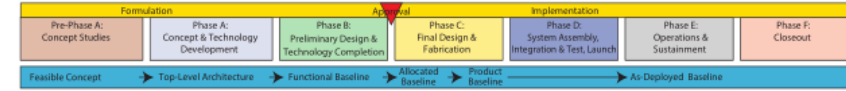
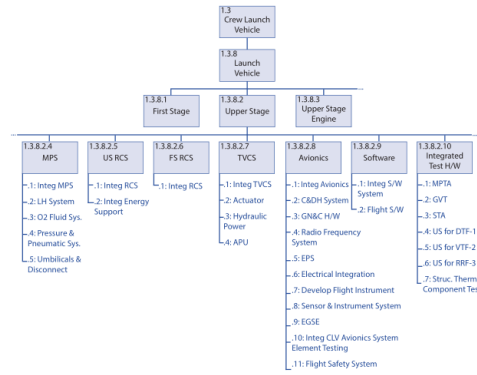
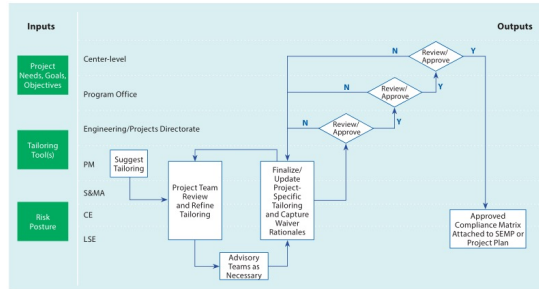


TABLE D-1 Requirements Verification Matrix

Requirement No.	Document	Paragraph	Shall Statement	Verification Success Criteria	Verification Method	Facility or Lab	Phase*	Acceptance Requirement?	Preflight Acceptance?	Performing Organization	Results
	Unique identifier or each requirement	Document number the requirement is contained within	Text within reason of the requirement, i.e., the "shall"	Success criteria for the requirement	Verification method for the requirement (analysis, inspection, demonstration, test)	Facility or laboratory used to perform the verification and validation	Phase in which the verification and validation will be performed	Indicate whether this requirement is also verified during any pre-flight or recurring acceptance testing of each unit.	Indicate whether this requirement is also verified during initial acceptance testing of each unit	Organization responsible for performing the verification	Indicate documents that contain the objective evidence that requirement was satisfied
P-1	xxx	3.2.1.1 Capability Support Uplinked Data (LDR)	System X shall provide a max. ground-to-station uplink of...	1. System X shall provide a max. ground-to-station uplink of... 2. System X locks to the forward link at the min and max operating frequency tolerances	Test	xxx	5	Yes	No	xxx	TPS xxxx
P-1	xxx	Other paragraphs	Other "shall" in PTRS	Other criteria	xxx	xxx	xxx	Yes/No	Yes/No	xxx	Memo xxx
S-1 or other unique designator	xxxxx (other specs, ICDs, etc.)	Other paragraphs	Other "shall" in specs, ICDs, etc.	Other criteria	xxx	xxx	xxx	Yes/No	Yes/No	xxx	Report xxx

Concept	Demonstration Units			Environment		Unit Description			Overall TRL			
	Breadboard	Brassboard	Developmental Model	Laboratory Environment	Relevant Environment	Space Environment	Space Launch Operation	Form		Fit	Function	Appropriate Scale
1.0 System												
1.1 Subsystem X												
1.1.1 Mechanical Components												
1.1.2 Mechanical Systems												
1.1.3 Electrical Components			X			X	X	X	X			
1.1.4 Electrical Systems												
1.1.5 Control Systems					X							
1.1.6 Thermal Systems							X	X				
1.1.7 Fluid Systems												
1.1.8 Optical Systems		X										
1.1.9 Electro-Optical Systems												
1.1.10 Software Systems												
1.1.11 Mechanisms		X										
1.1.12 Integration												
1.2 Subsystem Y												
1.2.1 Mechanical Components												



REPRESENTAÇÕES



REPRESENTAÇÕES

- Ao longo dos últimos anos, os engenheiros de sistemas evoluíram para algumas representações gráficas para apresentar as características funcionais e de fluxo de dados de seu projeto de sistema.
- Os mais comuns são: enhanced Function Flow Block Diagram (eFFBD), Data Flow Diagram (DFD), N2 (N-Squared) Chart e o IDEF0 Diagram.



FUNÇÕES

- Uma função é uma ação, uma operação ou um serviço, executada pelo sistema ou um de seus componentes, ou também por um ator interagindo com o sistema.
 - Executar uma função geralmente produz **itens de troca (exchanges)** esperados por outras funções e, para fazer isso, requer outros itens fornecidos por outras funções.
 - Várias funções podem ser agrupadas em uma função mãe (elas são então chamadas de subfunções, ou funções filhas, desta função). Simetricamente, uma função pode ser refinada em várias funções.
 - Este agrupamento não é uma forte relação de decomposição estrutural; o agrupamento de funções forma apenas uma representação sintética destes, essencialmente para fins documentais.
- Geralmente, em um modelo finalizado, apenas as funções folha (sem subfunções) referem-se e carregam a descrição funcional esperada.
- **Por convenção, uma função é nomeada com um verbo.**



ENHANCED FUNCTION FLOW BLOCK DIAGRAM (EFFBD)

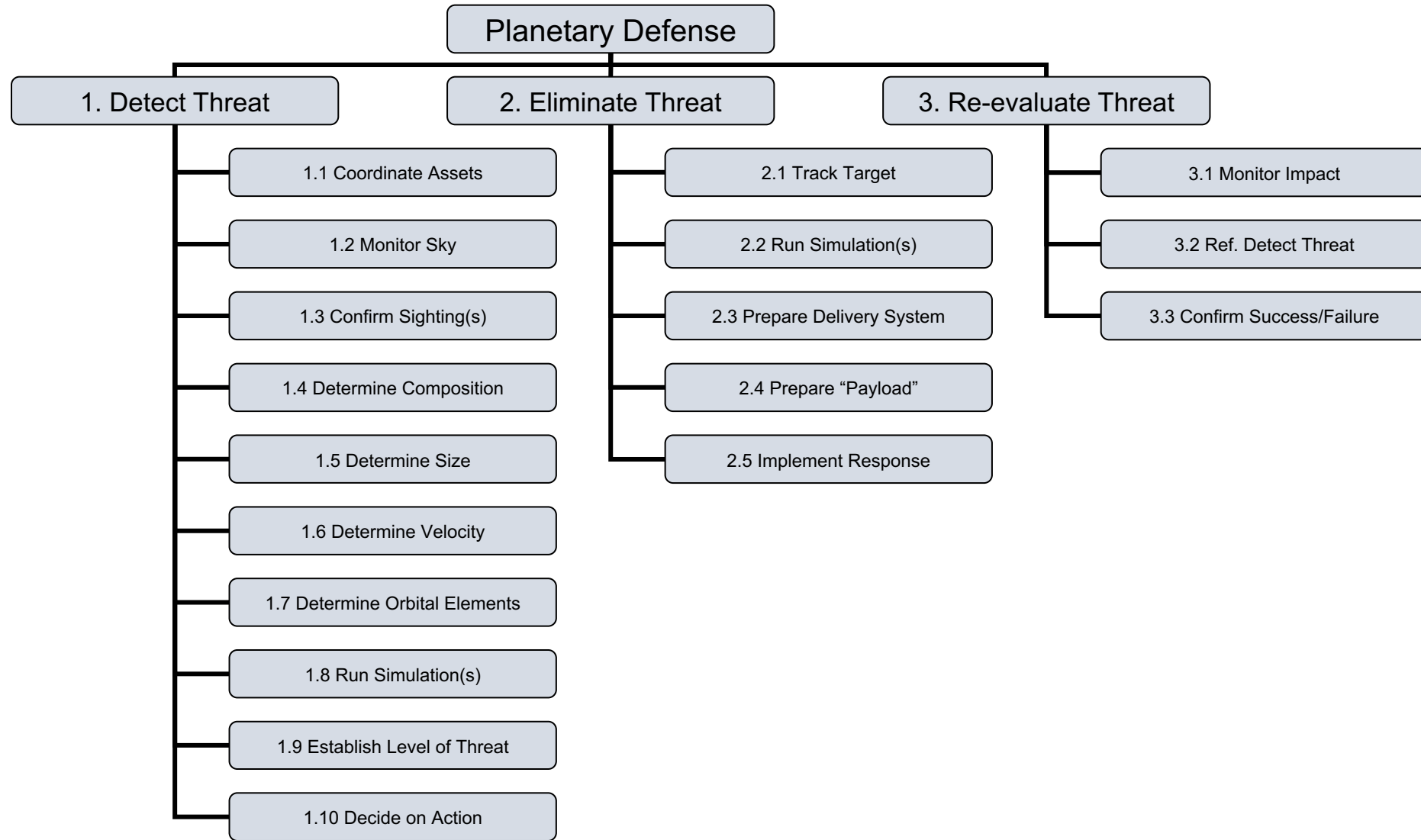
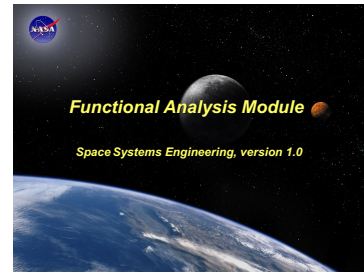


INTRODUÇÃO AO DIAGRAMA FFBD

- O FFBD é um diagrama de **várias camadas**, sequenciado, do **fluxo funcional** de um sistema.
- Um FFBD geralmente define as sequências e de suporte ao **delhamento passo a passo dos sistemas**, mas também pode ser usado de forma eficaz para definir processos ao desenvolver e produzir sistemas.
- No método FFBD, são organizadas e representadas por sua **ordem lógica de execução**.

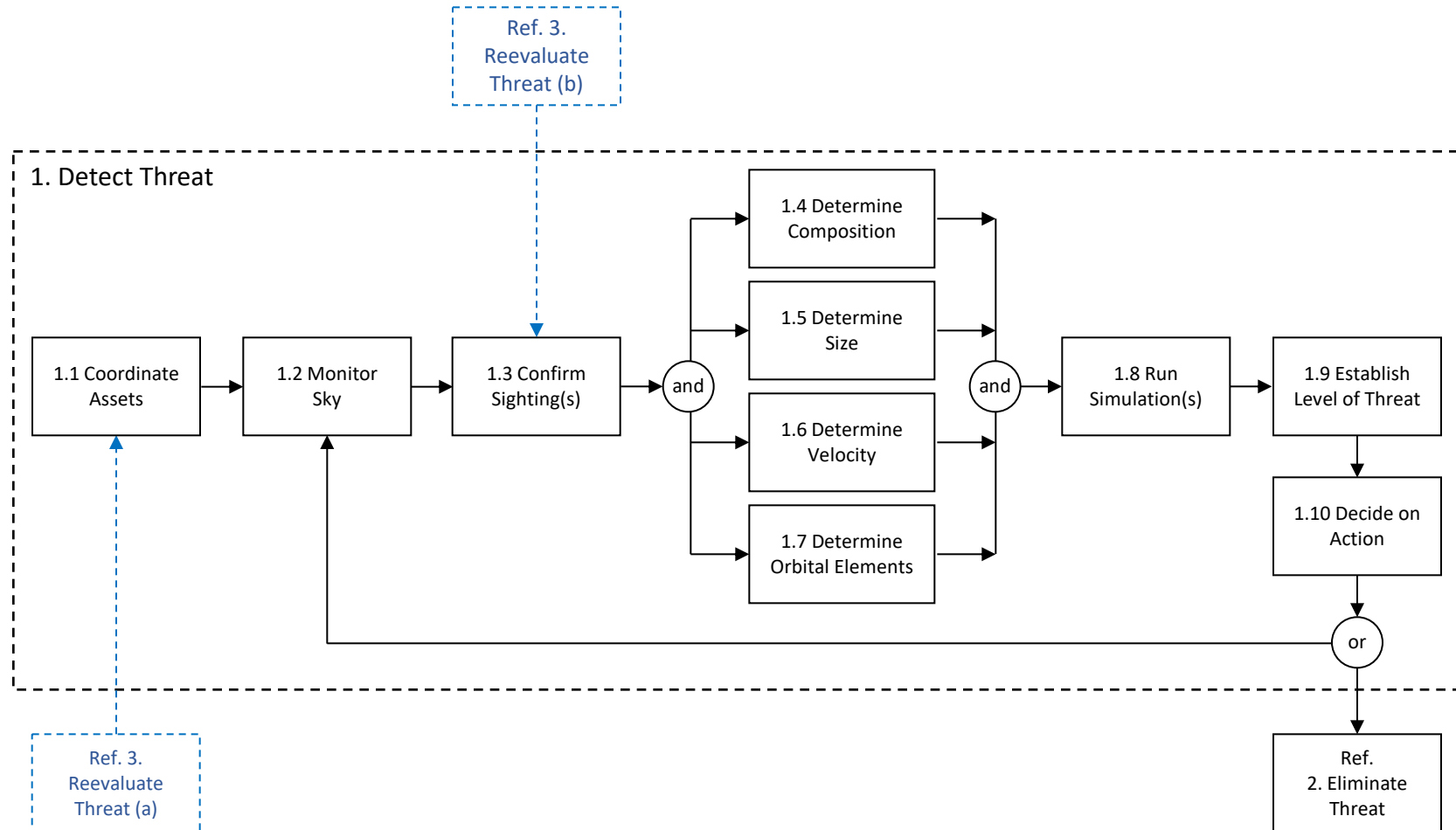


PLANETARY DEFENSE PROGRAM



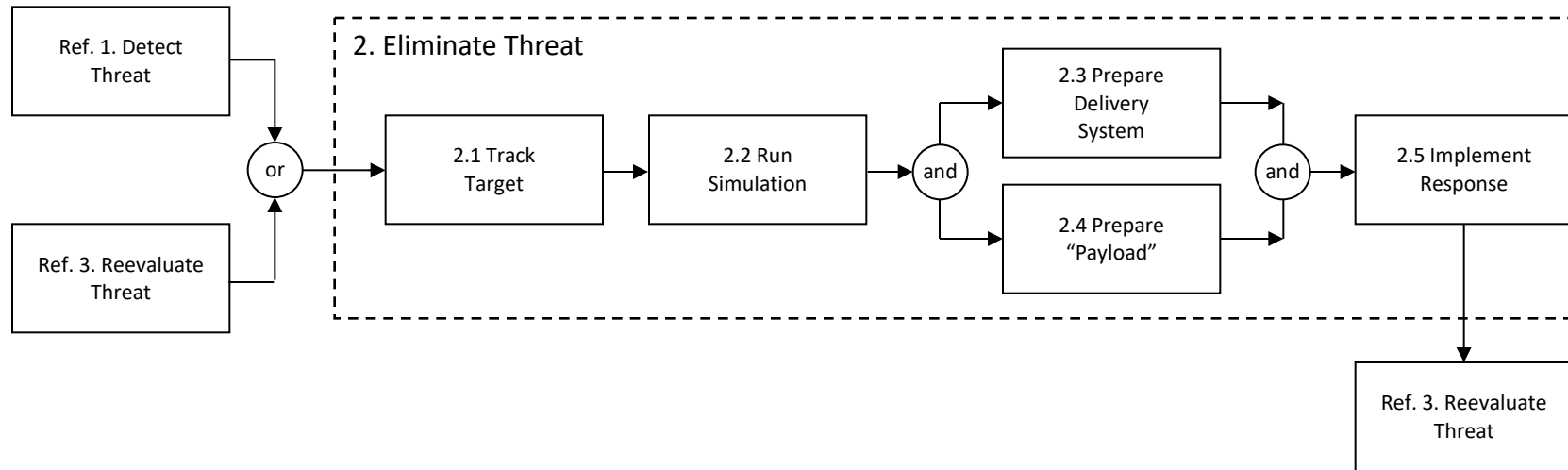


PLANETARY DEFENSE LEVEL 1 FUNCTIONAL FLOW BLOCK DIAGRAM FOR THREAT DETECTION



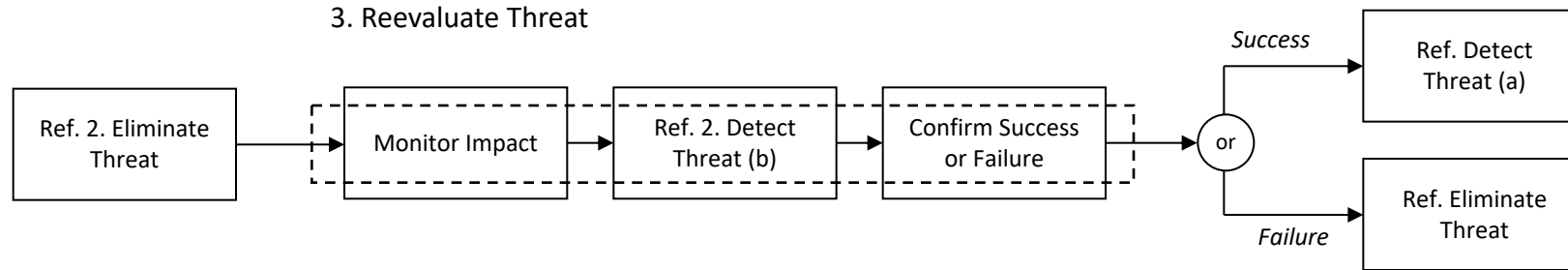


PLANETARY DEFENSE LEVEL 1 FUNCTIONAL FLOW BLOCK DIAGRAM FOR THREAT ELIMINATION





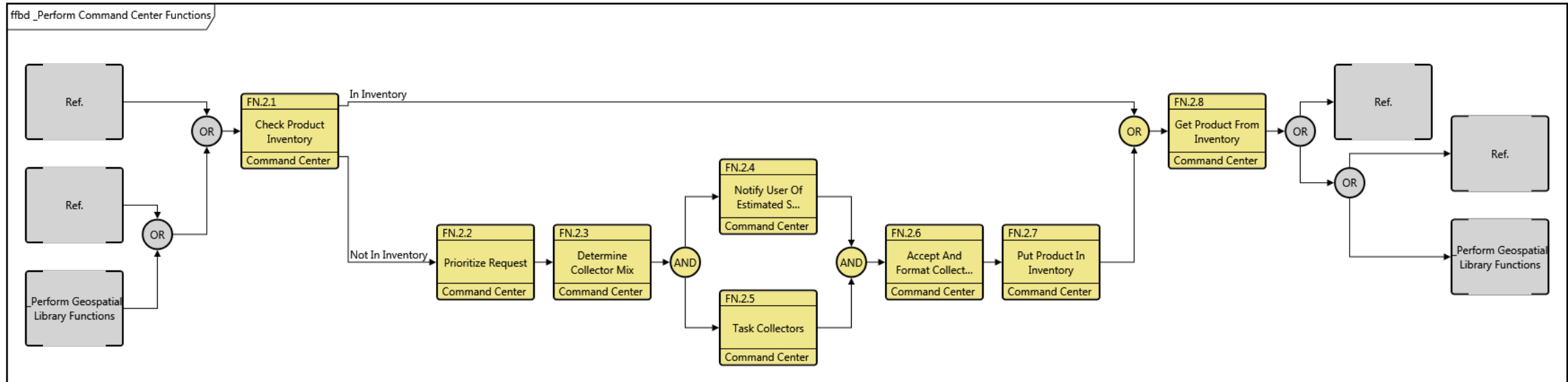
PLANETARY DEFENSE LEVEL 1 FUNCTIONAL FLOW BLOCK DIAGRAM FOR THREAT REEVALUATION





INTRODUÇÃO AO DIAGRAMA FFBD

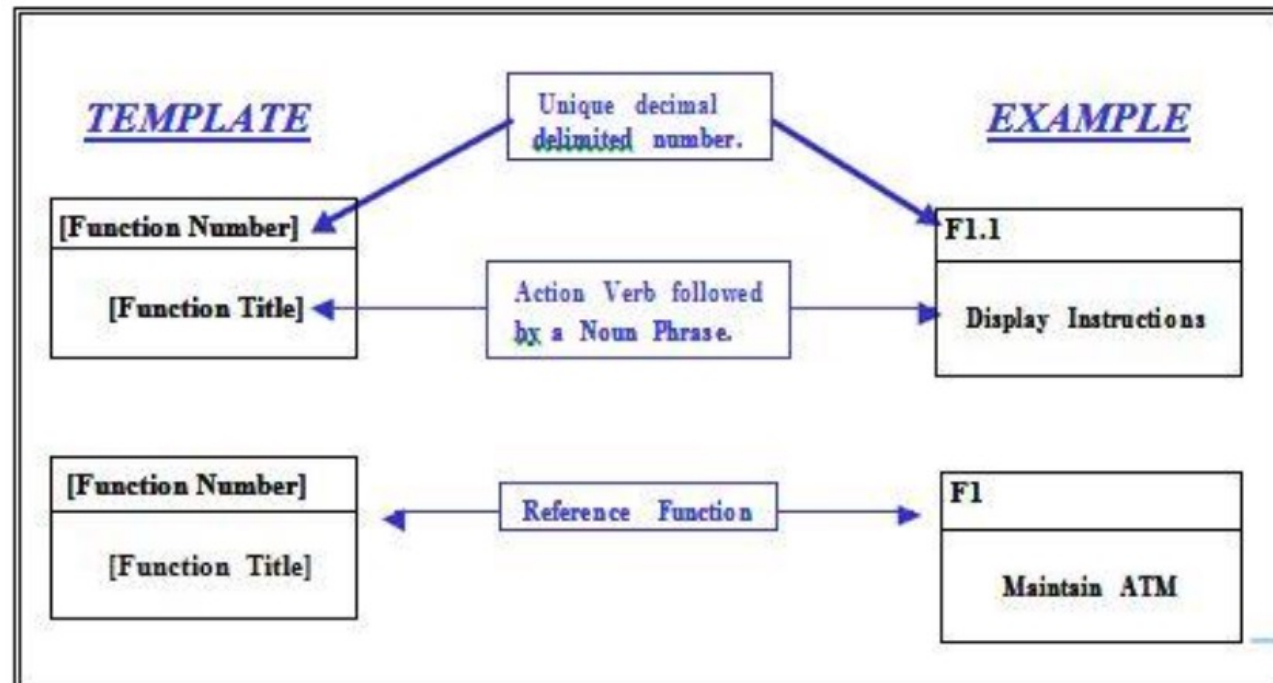
- Um conceito-chave na modelagem é que, para uma função começar, a função ou funções anteriores dentro do fluxo de "controle" devem ter terminado. Por exemplo, uma função "exibir destinos" logicamente não começaria até que uma função "detectar destinos" fosse concluída.
- A sequência lógica descreve o ambiente de "controle" do modelo funcional.
 - *UML/SysML incorporou essa lógica no diagrama de atividades*





SIMBOLOGIA

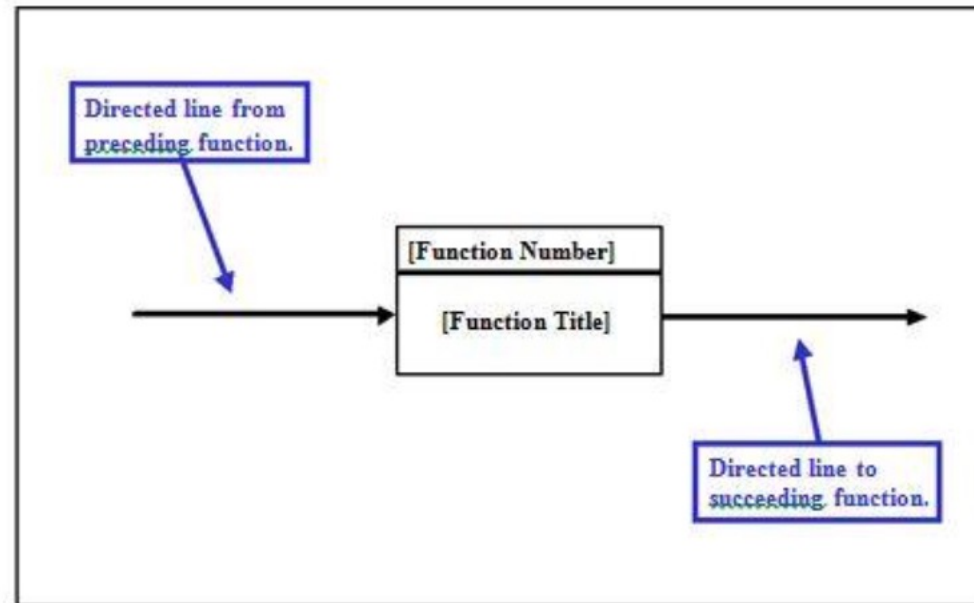
- Uma função deve ser representada por um retângulo contendo o título da função (um verbo de ação seguido de uma substantivo) e seu número exclusivo.





FLUXO FUNCIONAL

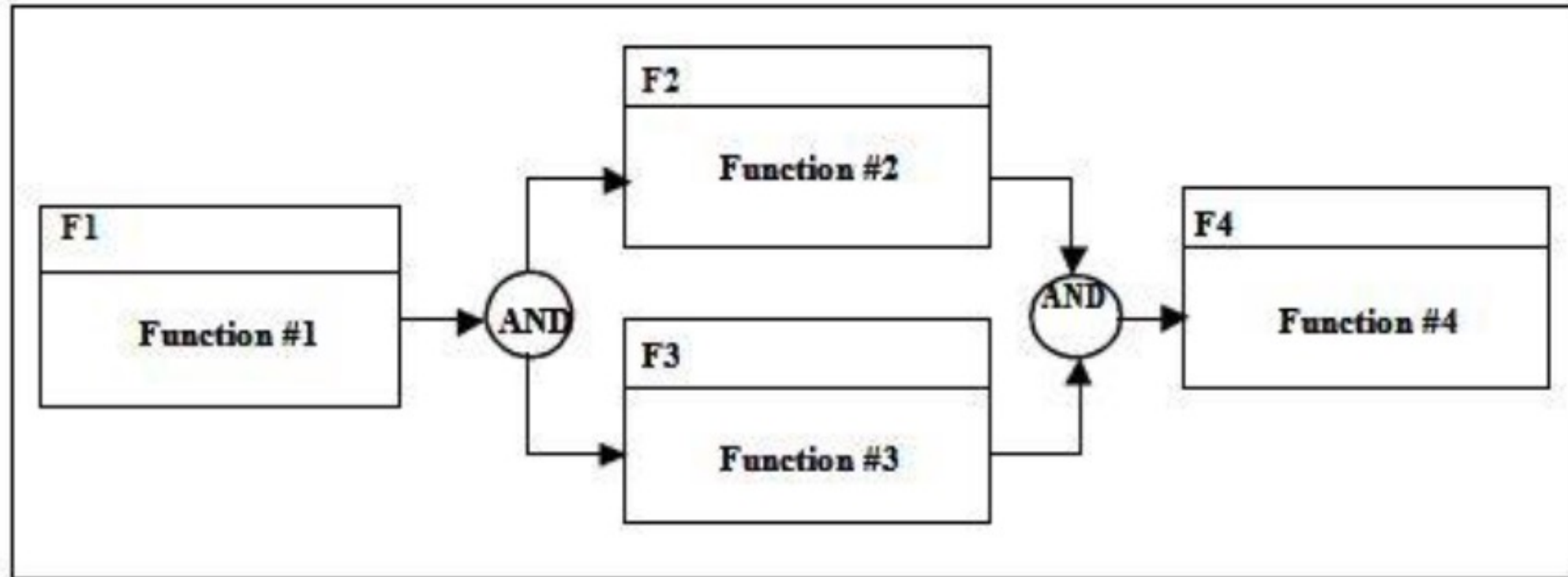
- Uma linha com uma única ponta de seta deve representar o fluxo funcional da esquerda para a direita.





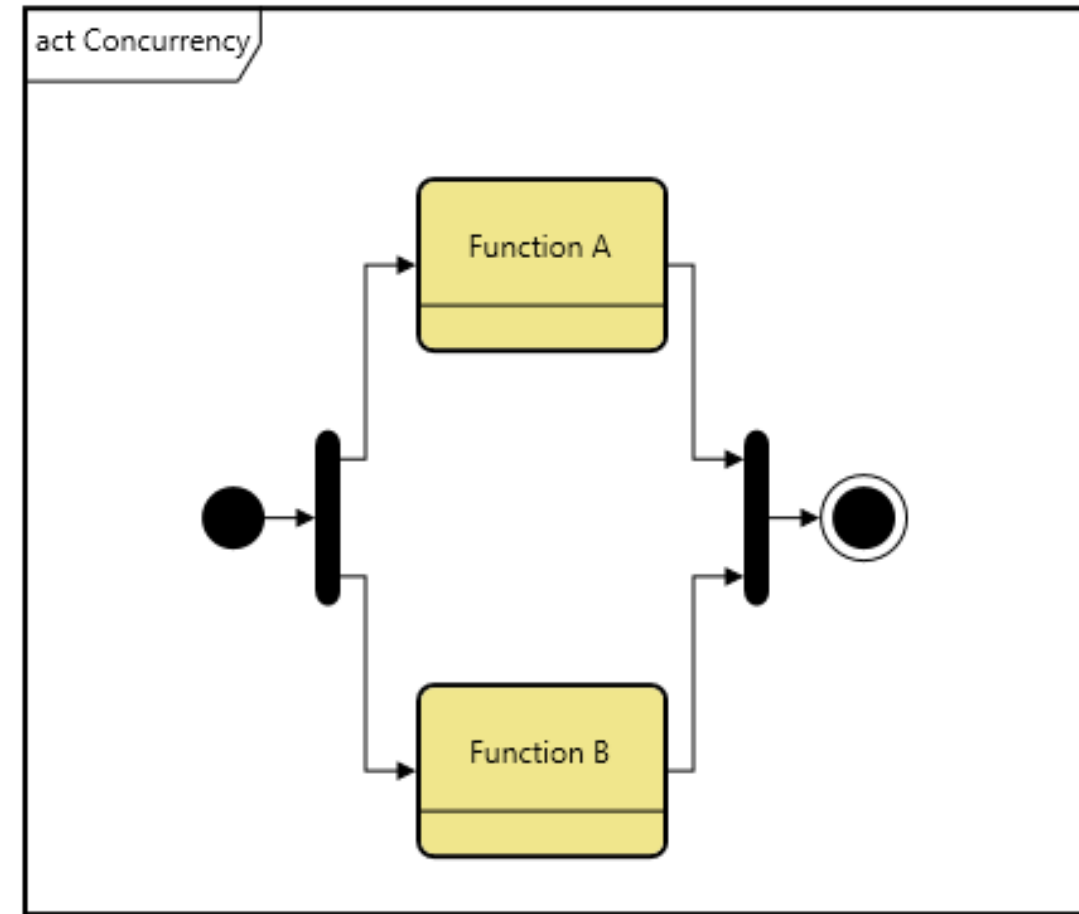
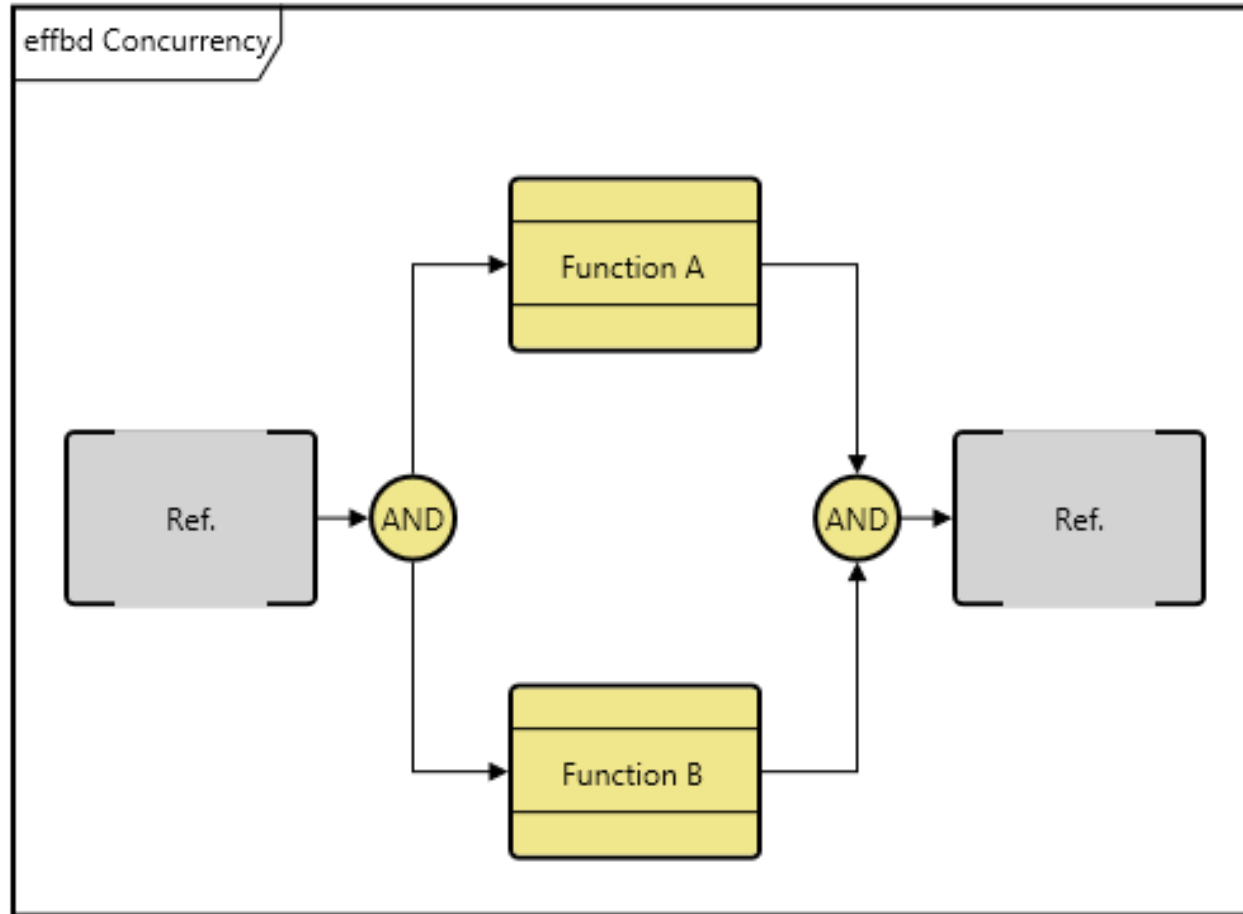
CONDIÇÕES LÓGICAS: E (AND)

- AND: Uma condição na qual todos os caminhos anteriores ou sucessivos são necessários.





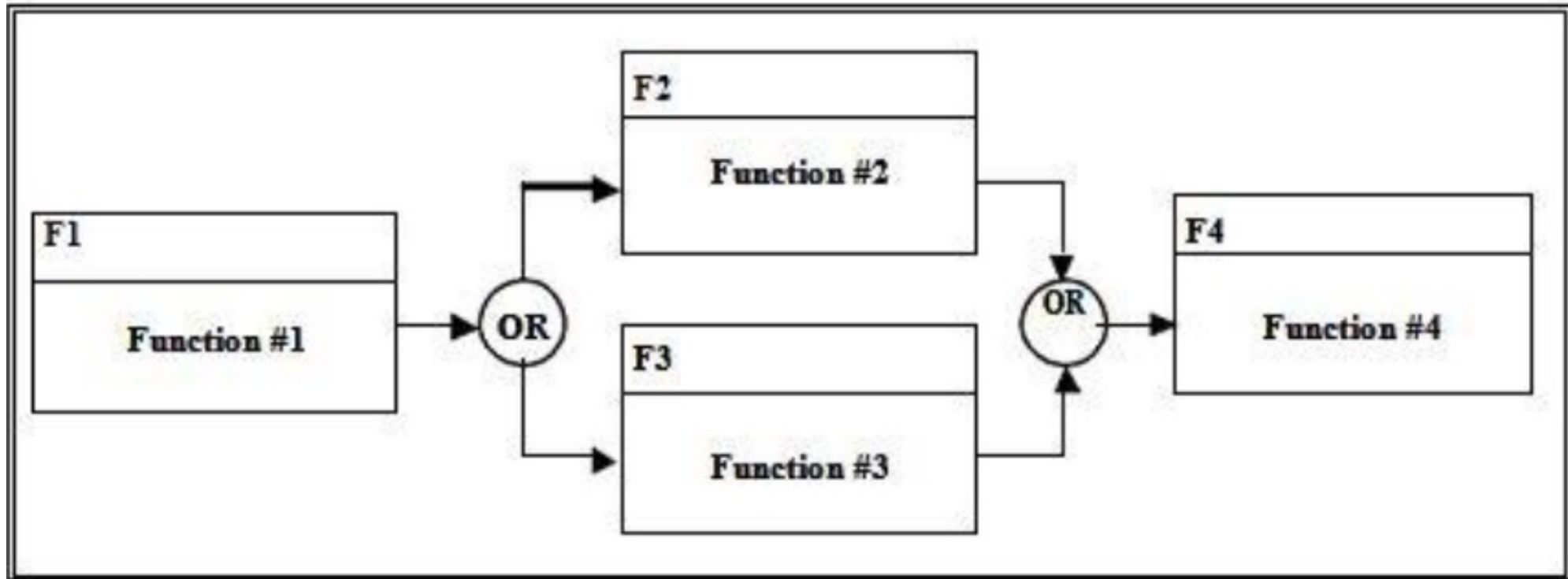
TAMBÉM PODE SER CHAMADO CONCORRÊNCIA / SIMULTÂNEO OU PARALELO





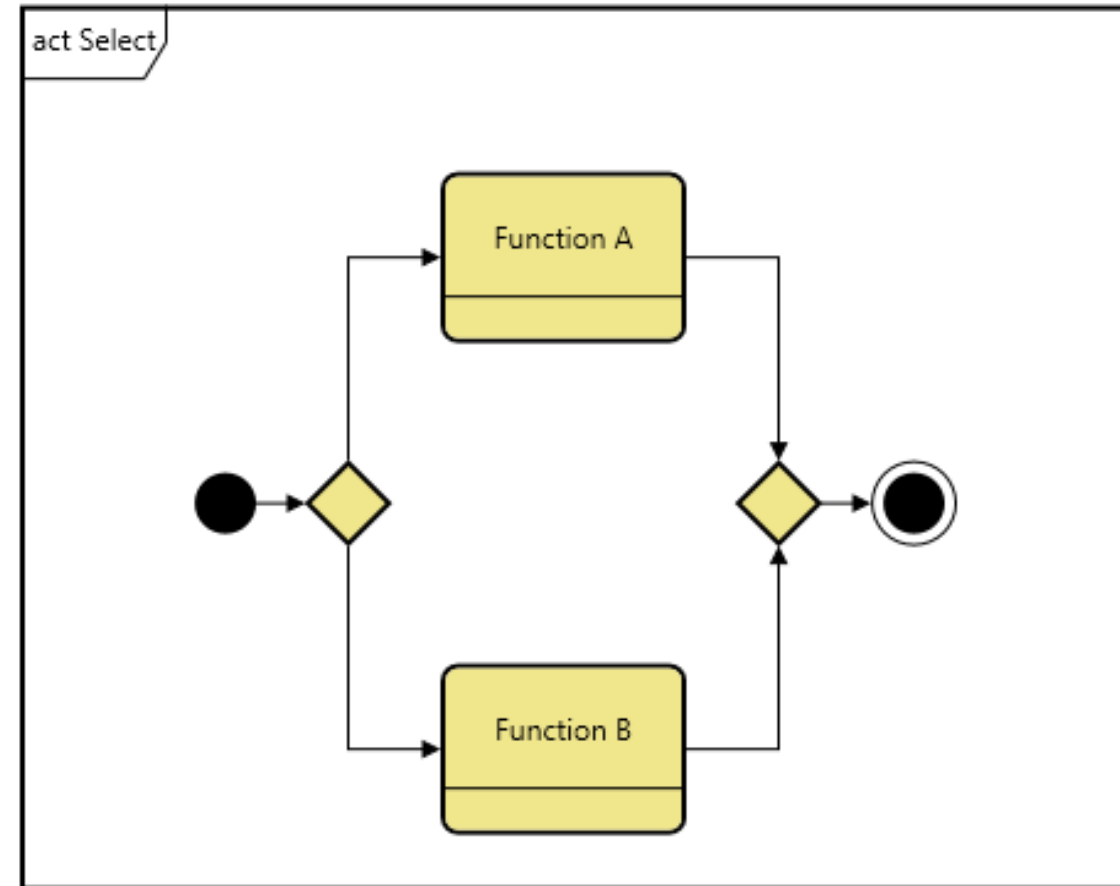
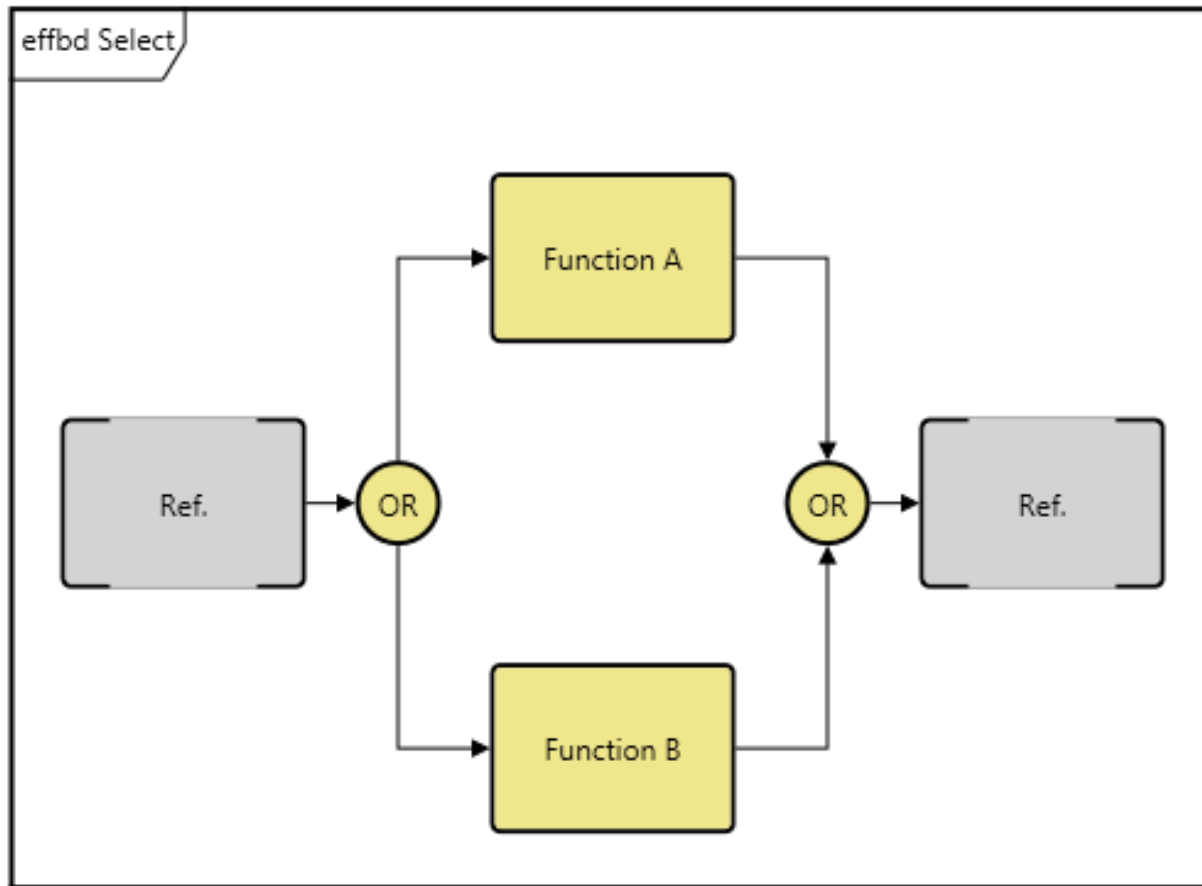
CONDIÇÕES LÓGICAS: OU EXCLUSIVO

- Exclusive OR: Uma condição na qual um dos vários caminhos anteriores ou sucessivos é necessário, mas não todos.





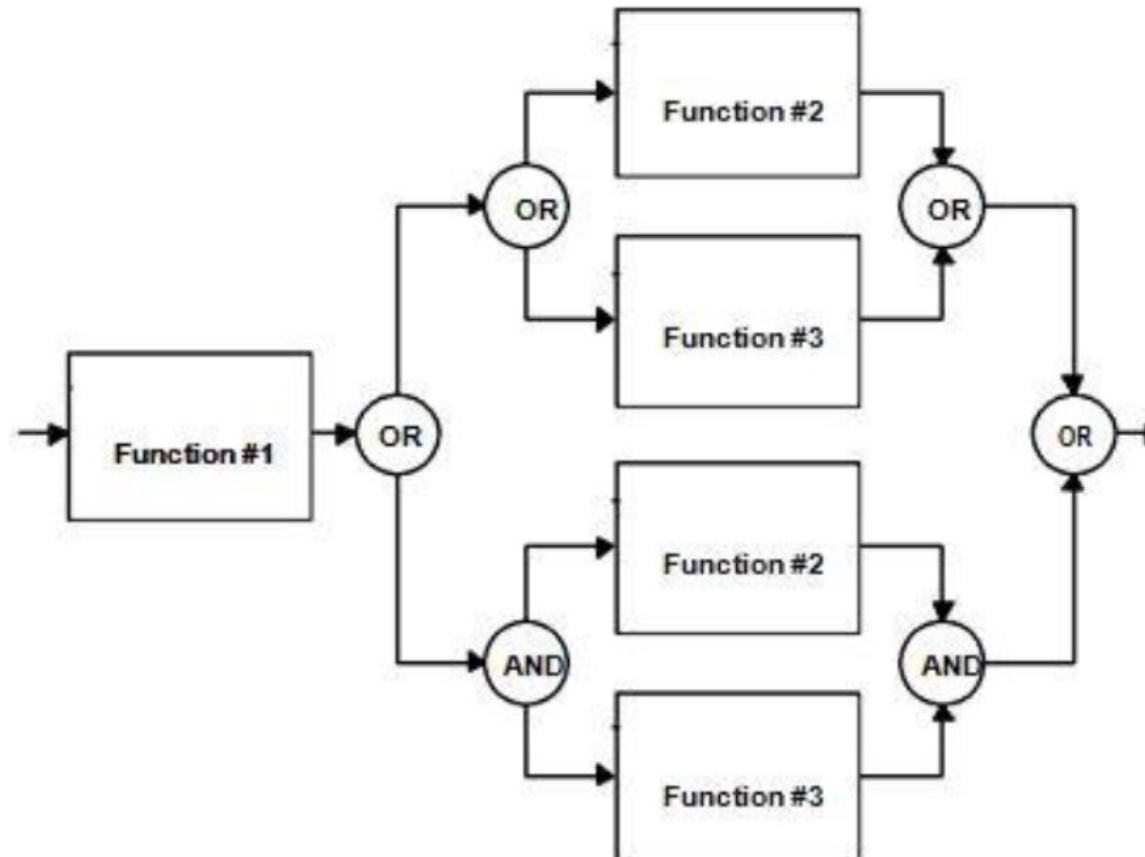
TAMBÉM PODE SER CHAMADO SELECT





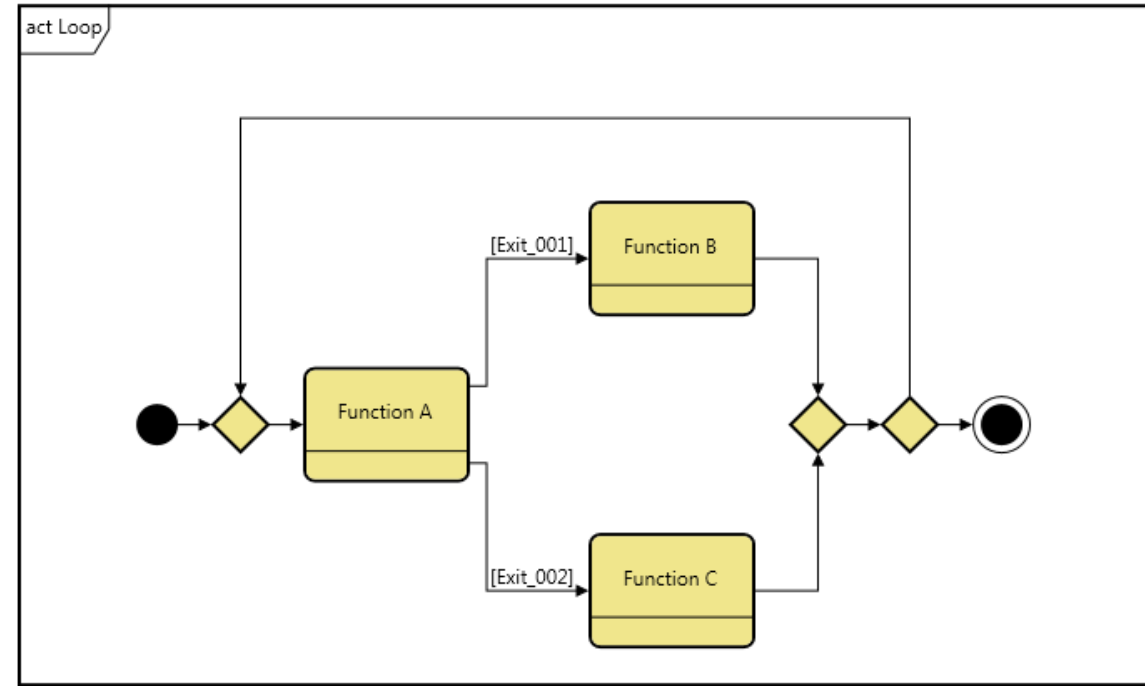
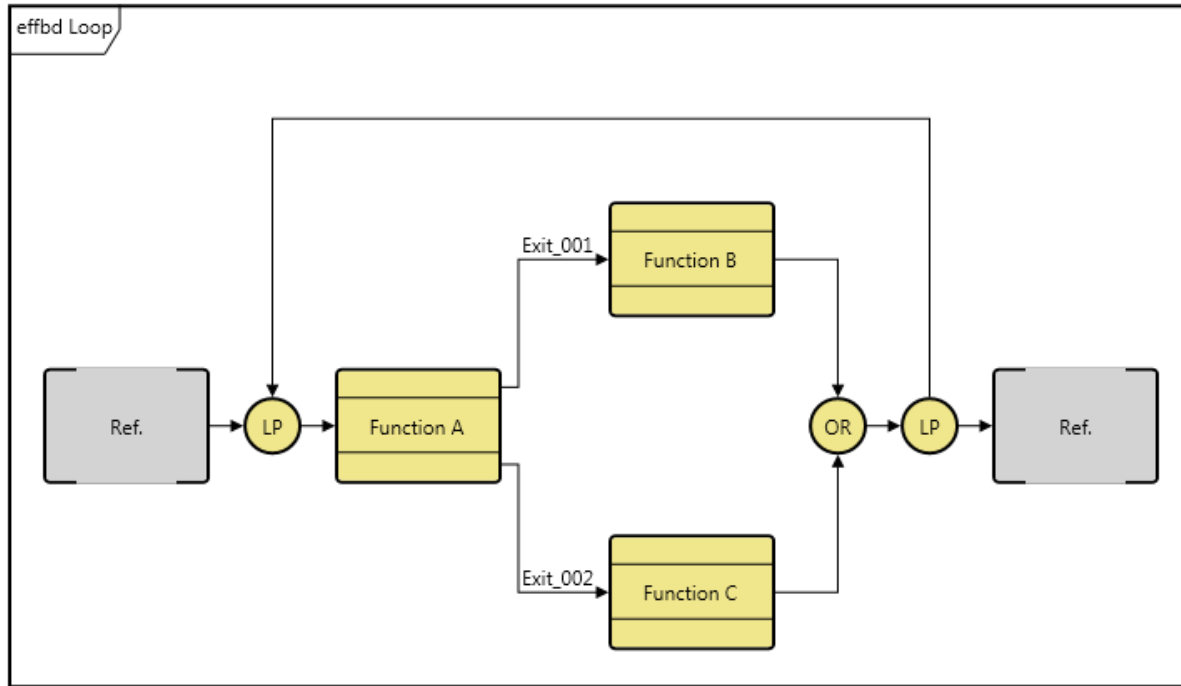
CONDIÇÕES LÓGICAS: OU INCLUSIVO

- Inclusive OR: Uma condição na qual um, alguns ou todos os vários caminhos anteriores ou sucessivos são necessários.





LOOP (POSSUI CONDIÇÃO DE SAÍDA)



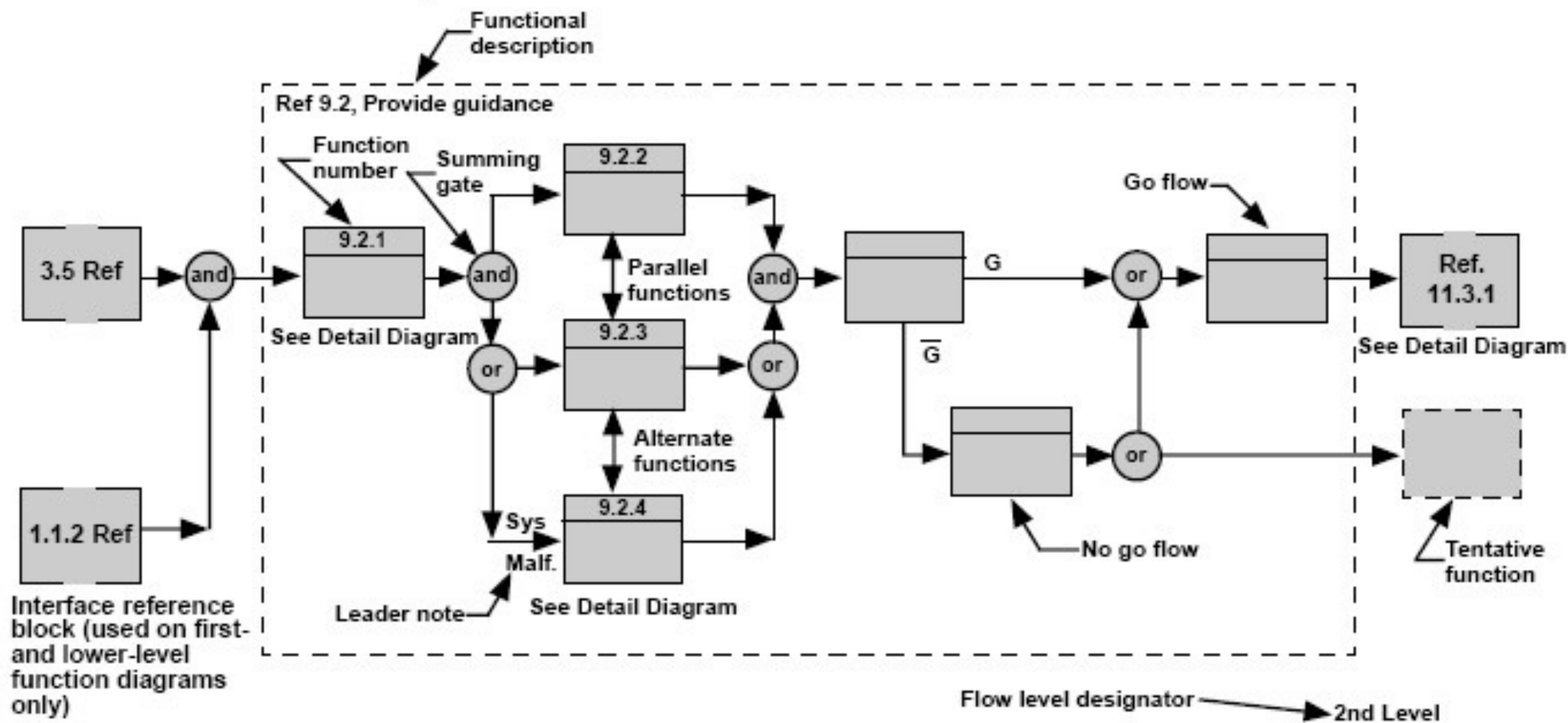


DESENVOLVIMENTO DE DIAGRAMAS FUNCIONAIS DE BLOCOS DE FLUXO

- FFBDs podem ser desenvolvidos **em uma série de níveis**.
- Os FFBDs mostram as mesmas tarefas identificadas através da **decomposição funcional** e as exibem em sua relação lógica e sequencial.
-
- O FFBD também incorpora **operações alternativas** e de contingência, que melhoram a probabilidade de sucesso da missão. O diagrama de fluxo fornece uma compreensão da operação total do sistema, serve como base para o desenvolvimento de procedimentos operacionais e de contingência.

Abbreviations/Notes:

“And” Gate: Parallel Function
 “Or” Gate: Alternate Function

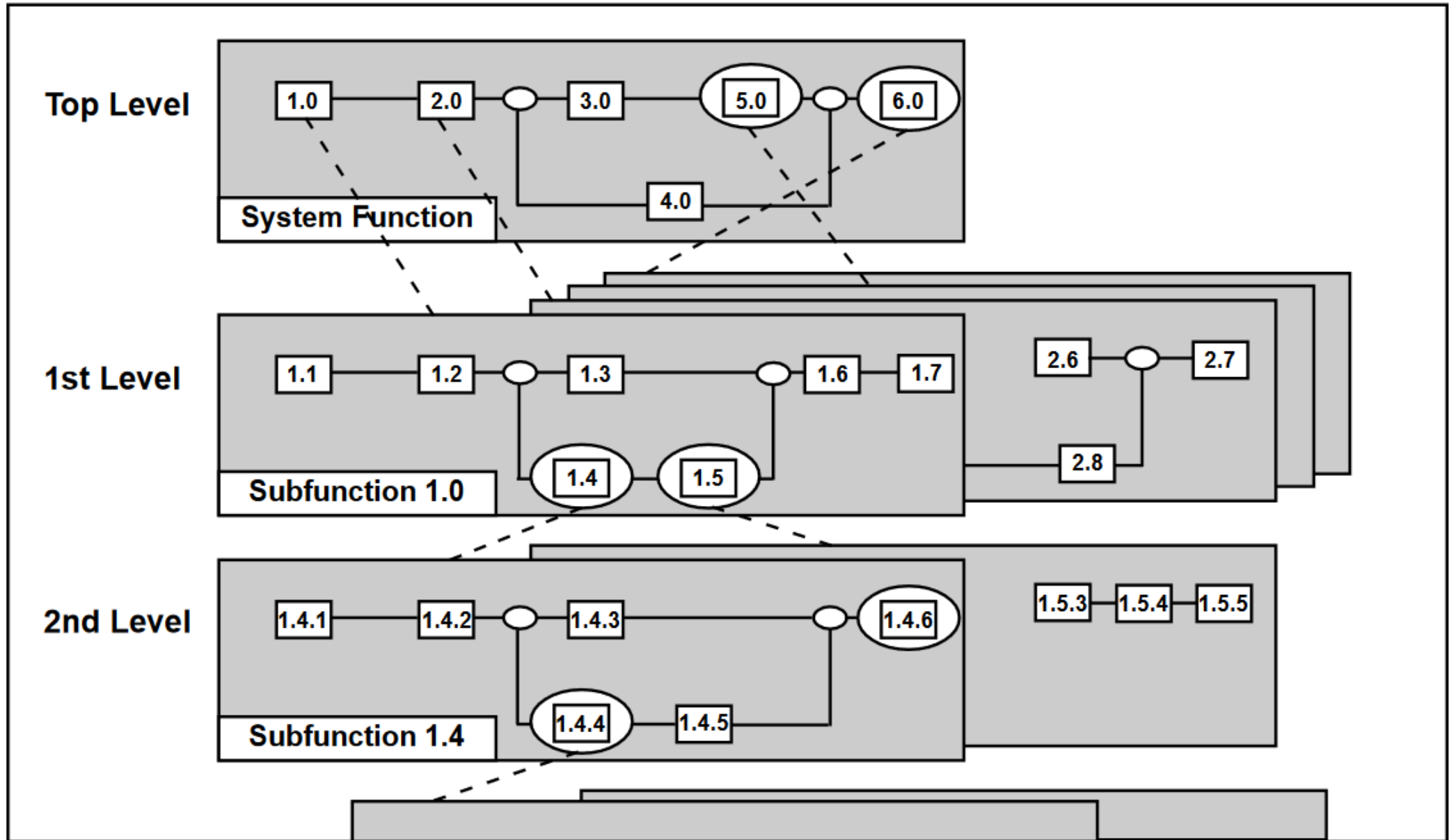


Interface reference block (used on first- and lower-level function diagrams only)

Flow level designator → 2nd Level

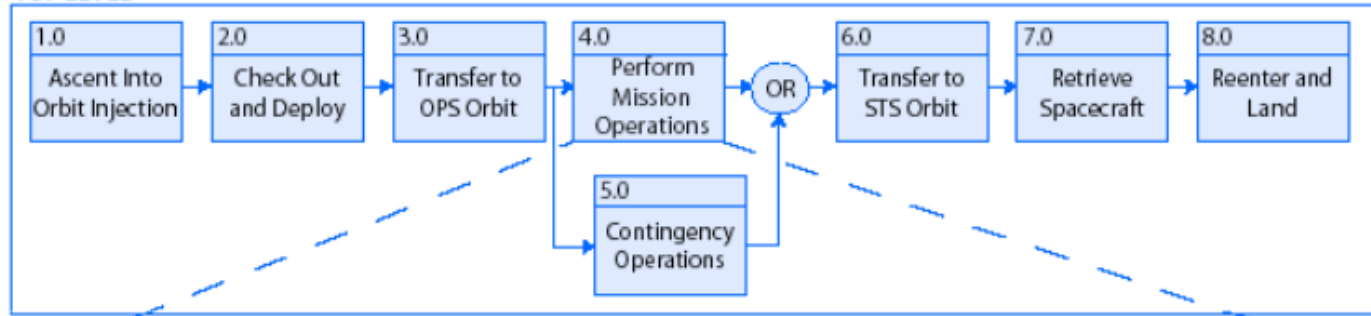
Scope Note: _____

Title block and standard drawing number → Functional Flow Block Diagram Format

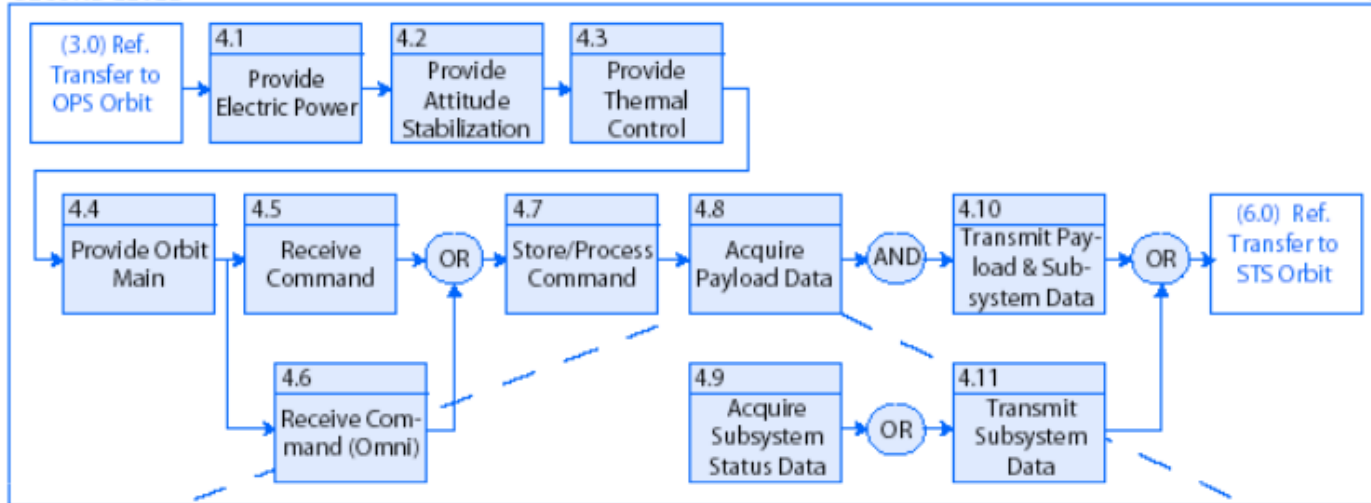




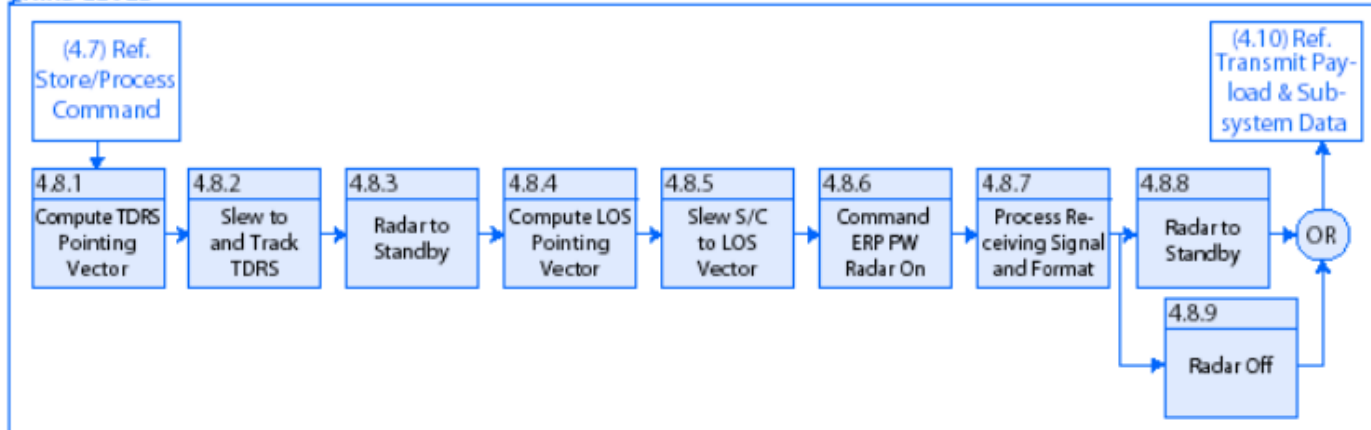
TOP LEVEL



SECOND LEVEL



THIRD LEVEL





ENHANCED FUNCTIONAL FLOW BLOCK DIAGRAM

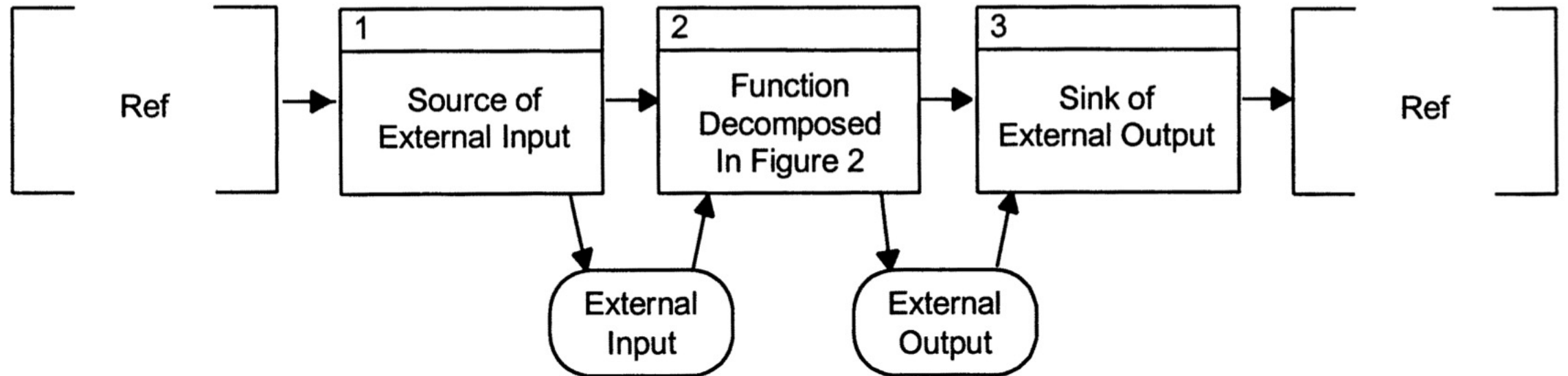


Figure 1. Enhanced functional flow block diagram context for Figure 2.

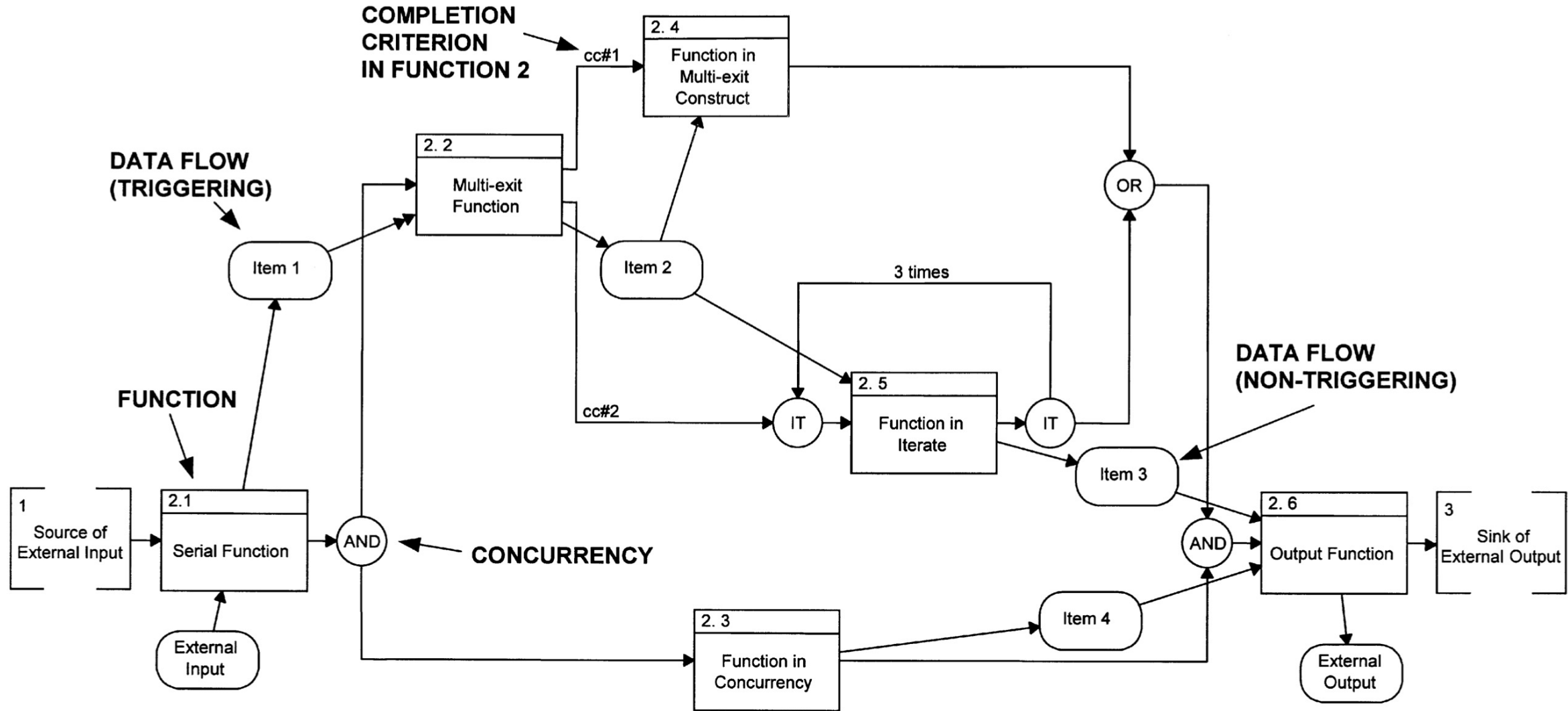


Figure 2. Enhanced functional flow block diagram for function 2 in Figure 1.

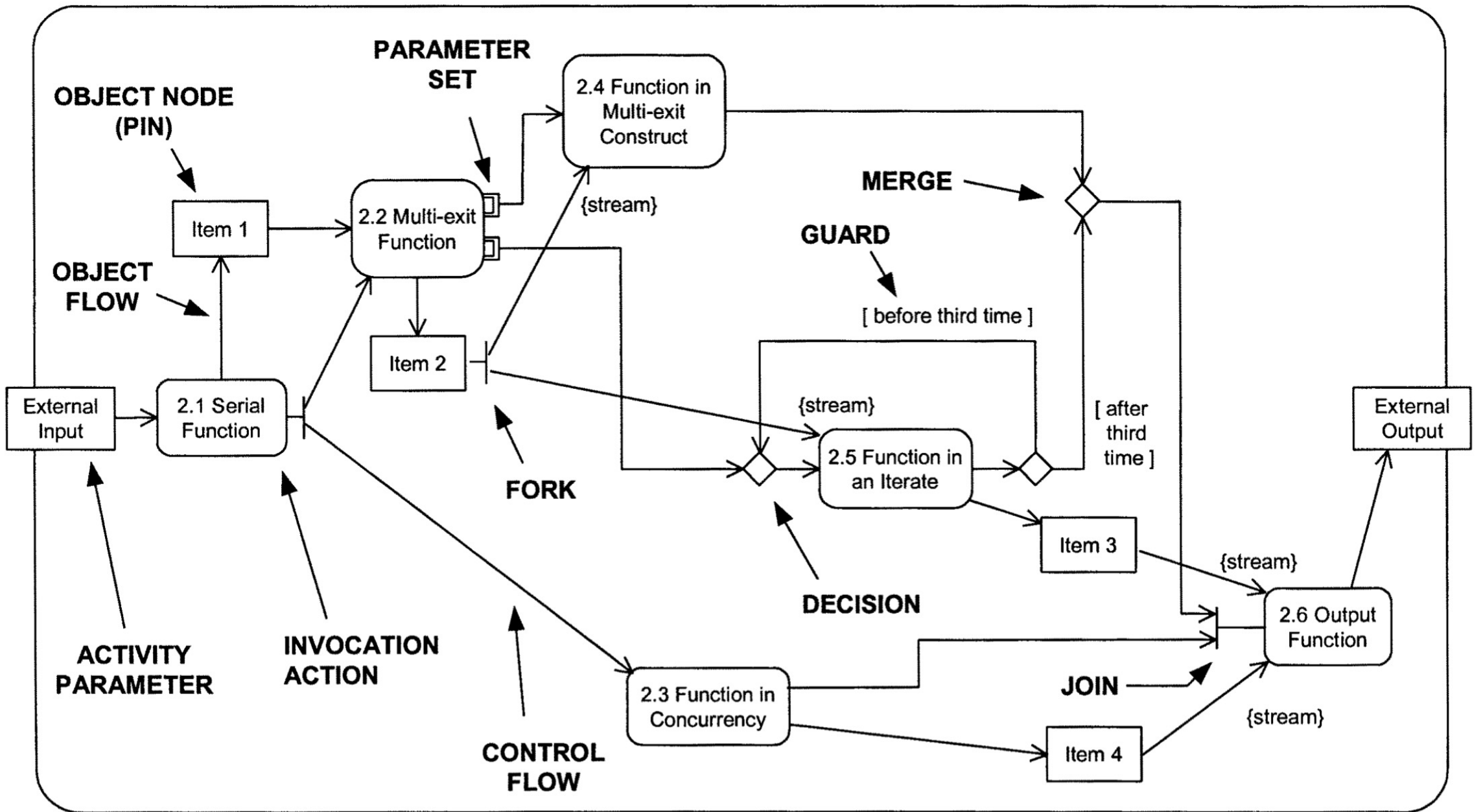


Figure 3. UML 2 Activity diagram corresponding to Figure 2.



DATA FLOW DIAGRAM (DFD)



DIAGRAMA DE FLUXO DE DADOS

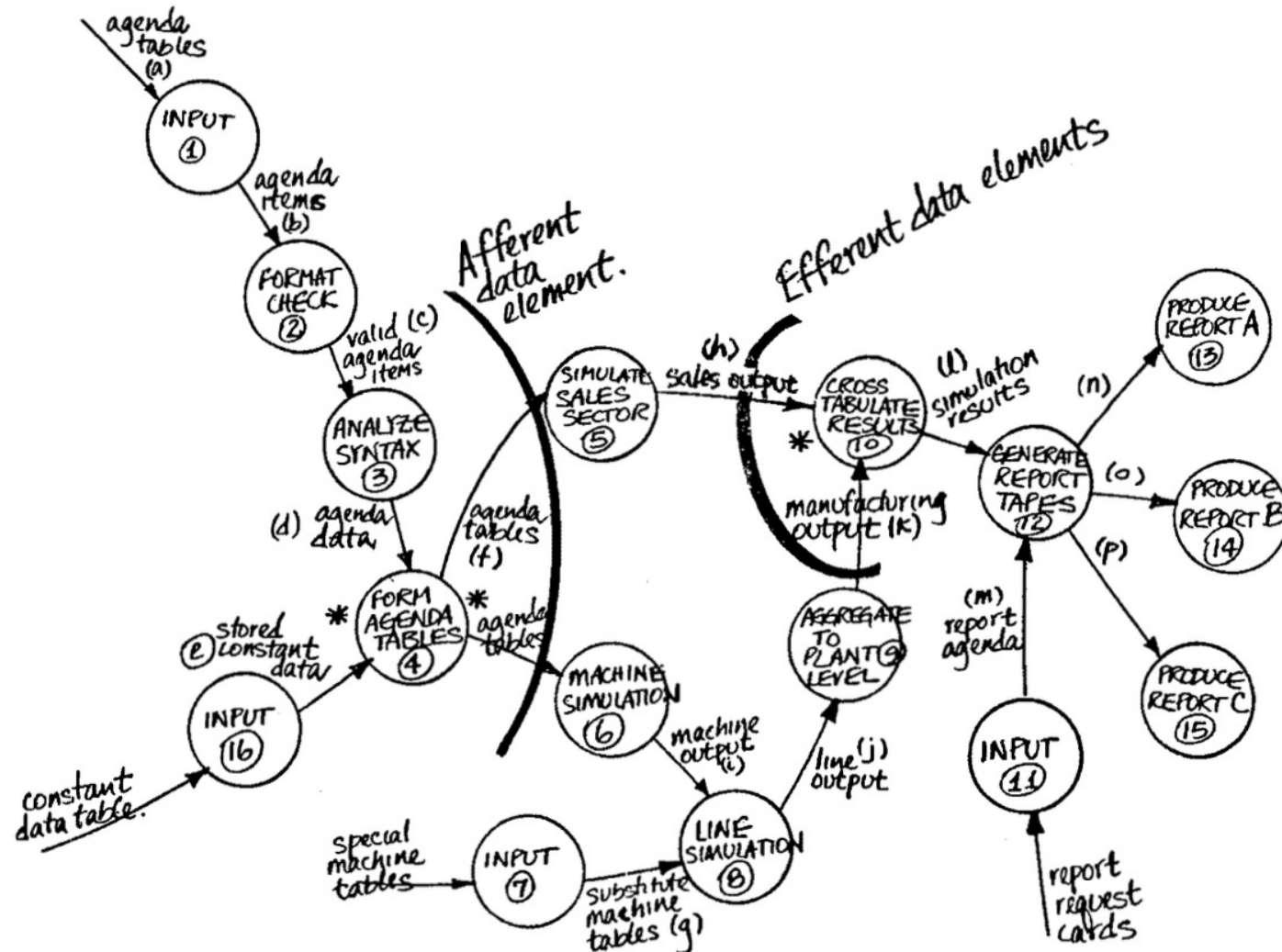


Figure 10.5. Data flow diagram for a simulation system.



ORIGEM

- O diagrama de fluxo de dados (DFD) é uma representação gráfica ou visual que usa um conjunto padronizado de símbolos e notações para descrever as operações por meio da movimentação de dados.
- Eles são muitas vezes elementos de uma metodologia formal, como o Método de Análise e Projeto de Sistemas Estruturados.



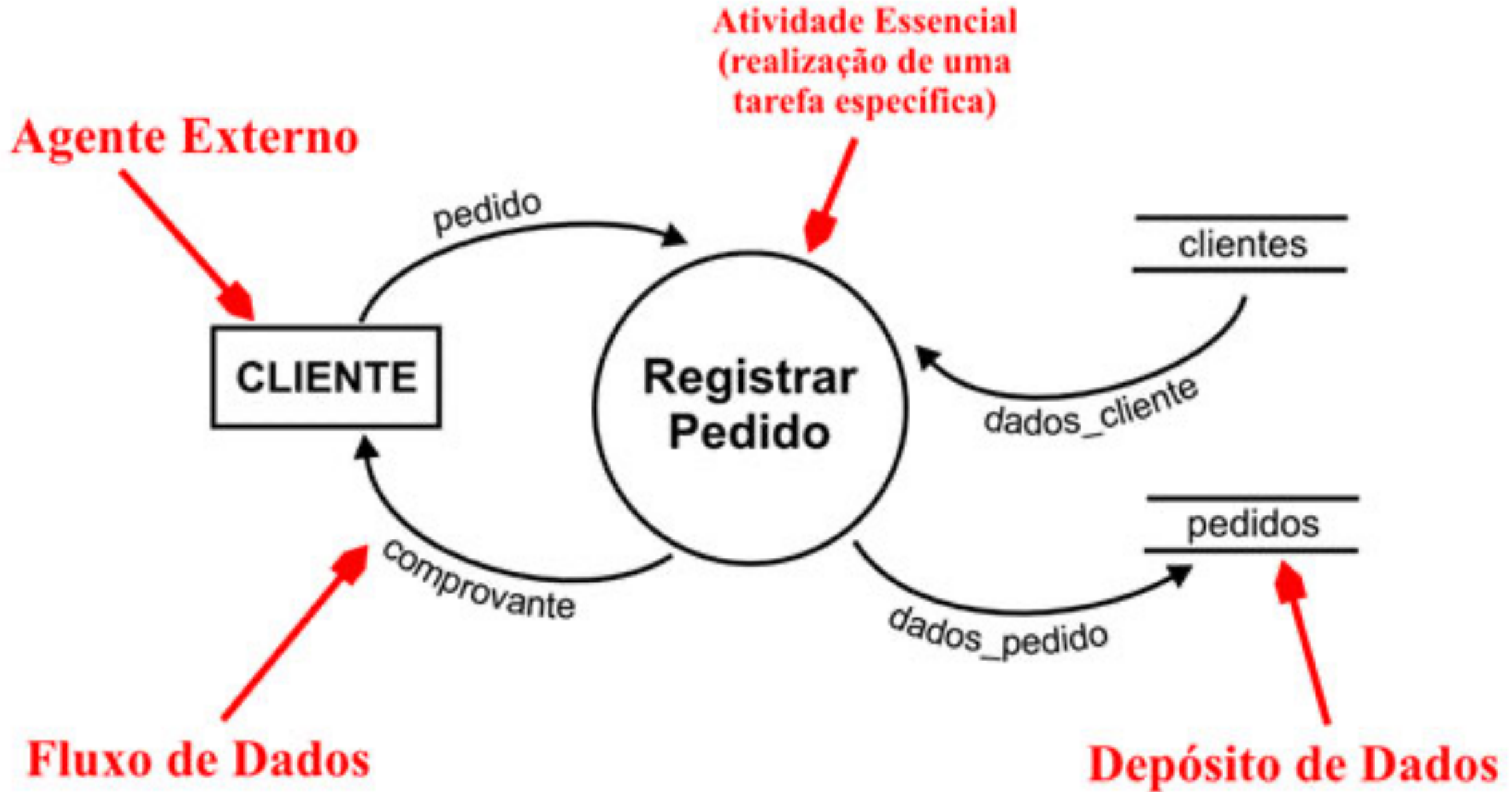
TODAS AS NOÇÕES DE DFD REPRESENTARÃO O SEGUINTE:

- External entities: informações que entram ou saem do Sistema
- Flows: definir o movimento de informações para, de e dentro do sistema
- Stores: locais onde as informações são mantidas, na maioria das vezes bancos de dados ou tabelas de banco de dados
- Processes: transformar informações



NÍVEIS

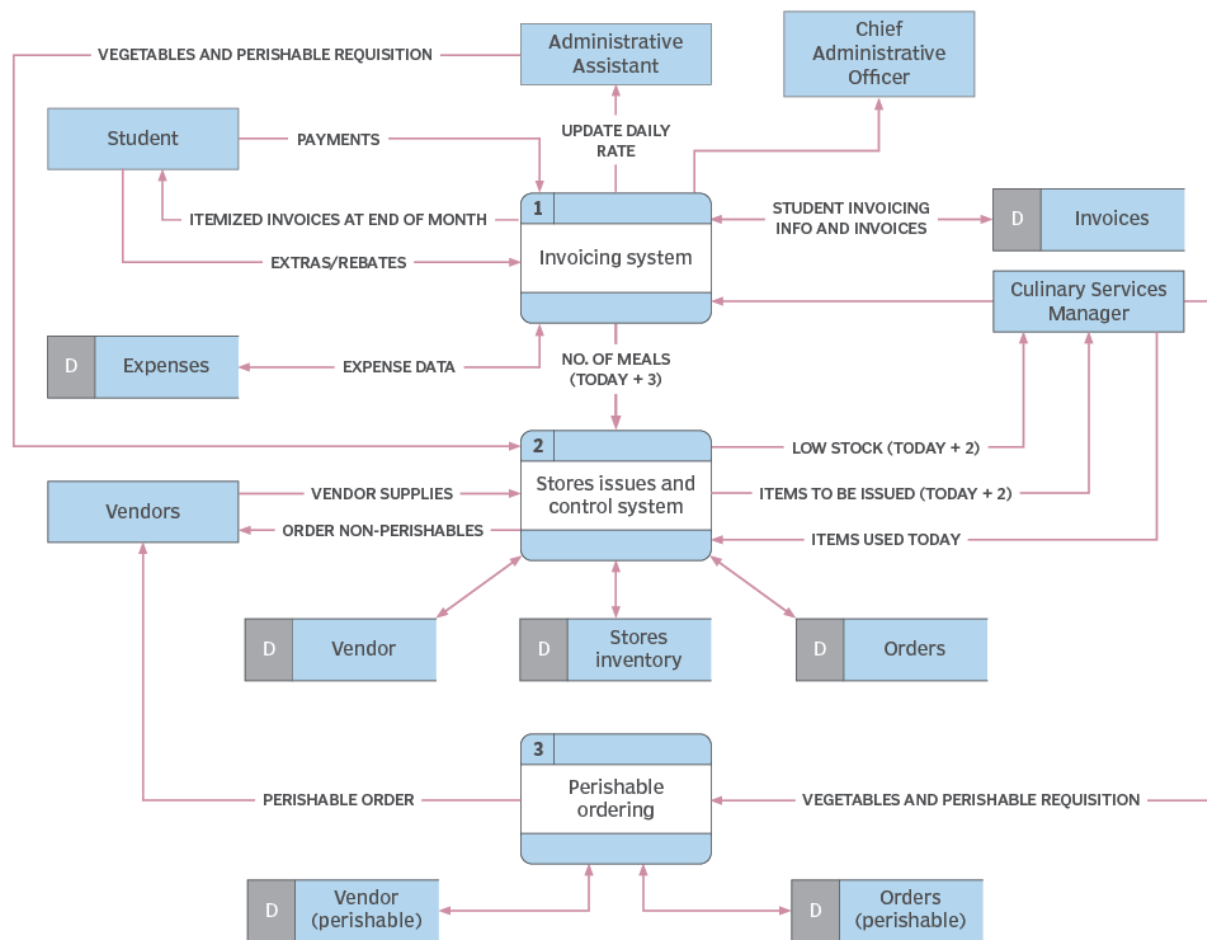
- Níveis ou camadas são usados em DFDs para representar graus progressivos de detalhe sobre o sistema ou processo.
 - Level 0: Também conhecido como "diagrama de contexto", este é o nível mais alto e representa uma visão muito simples e de nível superior do sistema que está sendo representado.
 - Level 1: Ainda uma visão relativamente ampla do sistema, mas incorpora subprocessos e mais detalhes.
 - Level 2: Fornece ainda mais detalhes e continua a detalhar os subprocessos conforme necessário.
 - Level 3: Embora essa quantidade de detalhes seja incomum, sistemas complexos podem se beneficiar da representação nesse nível.





Example of a data flow diagram

Using the Gane & Sarson method, this data flow diagram shows the information system of a school's culinary program.





N2 (N-SQUARED)



N2 DIAGRAMS

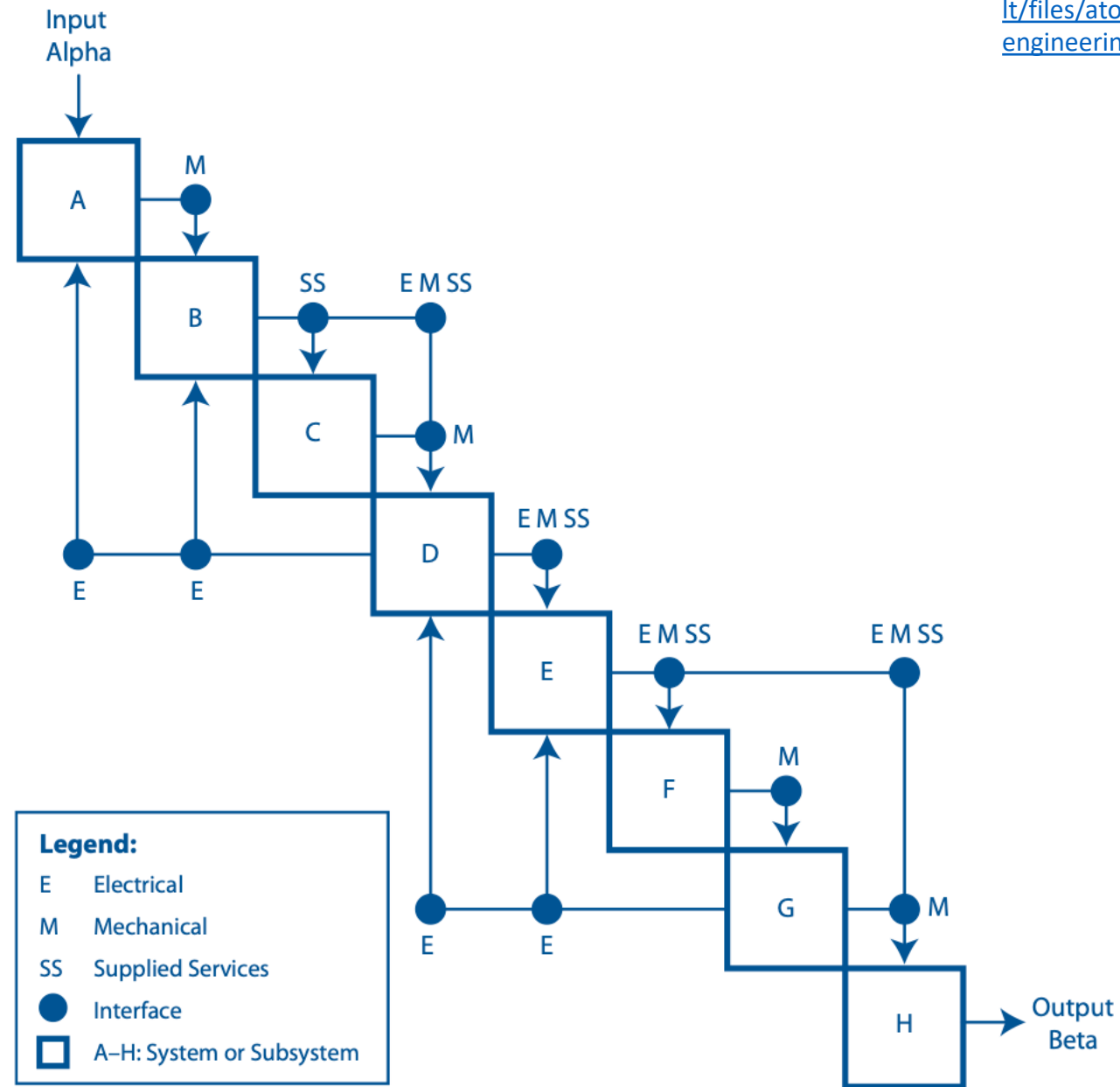


Figure 4.3-4 Example of an N2 diagram



N2 ANALYSIS

- N2 Analysis é uma ferramenta que usa uma matriz $n \times n$ para registrar as interconexões entre elementos de um sistema. Tem uma série de usos potenciais:
 - No projeto do sistema para avaliar o grau de ligação e acoplamento em um sistema e, assim, determinar arquiteturas candidatas com base na estrutura natural do sistema.
 - No projeto de sistemas para registrar, e daí ajudar a gestão de, as interfaces em um sistema.
 - Na análise de sistemas para identificar e documentar a interconectividade em um sistema para ajudar a entender o comportamento observado e fornecer orientação para melhoria.



N2 DIAGRAMS

- O diagrama N2 tem sido amplamente utilizado para desenvolver interfaces de dados, principalmente nas áreas de software. No entanto, ele também pode ser usado para desenvolver interfaces de hardware.
- As funções do sistema são colocadas na diagonal; o restante dos quadrados na matriz $N \times N$ representa as entradas e saídas da interface. Quando um espaço em branco aparece, não há interface entre as respectivas funções.
- Os dados fluem no sentido horário entre as funções (por exemplo, o símbolo $F_1 \rightarrow F_2$ indica dados que fluem da função F_1 para a função F_2).
- Os dados que estão sendo transmitidos podem ser definidos nos quadrados apropriados.

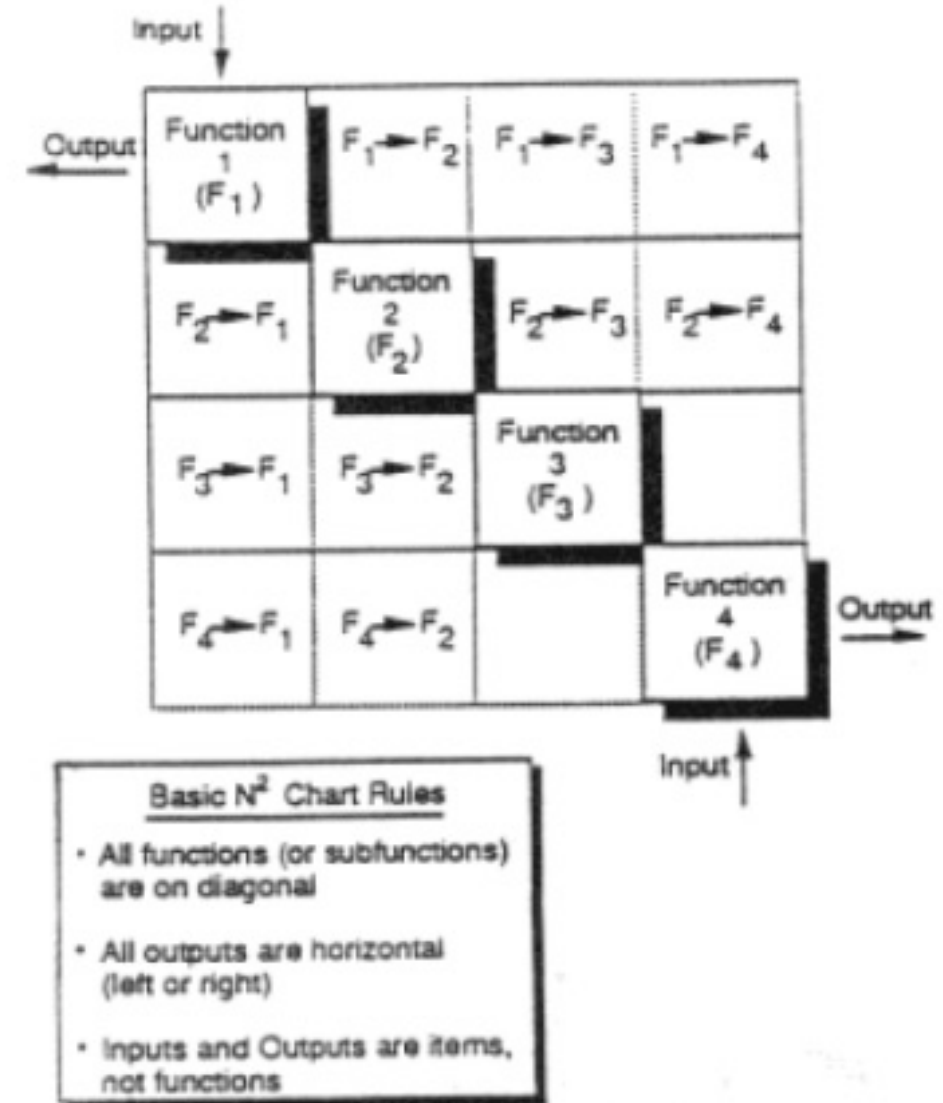
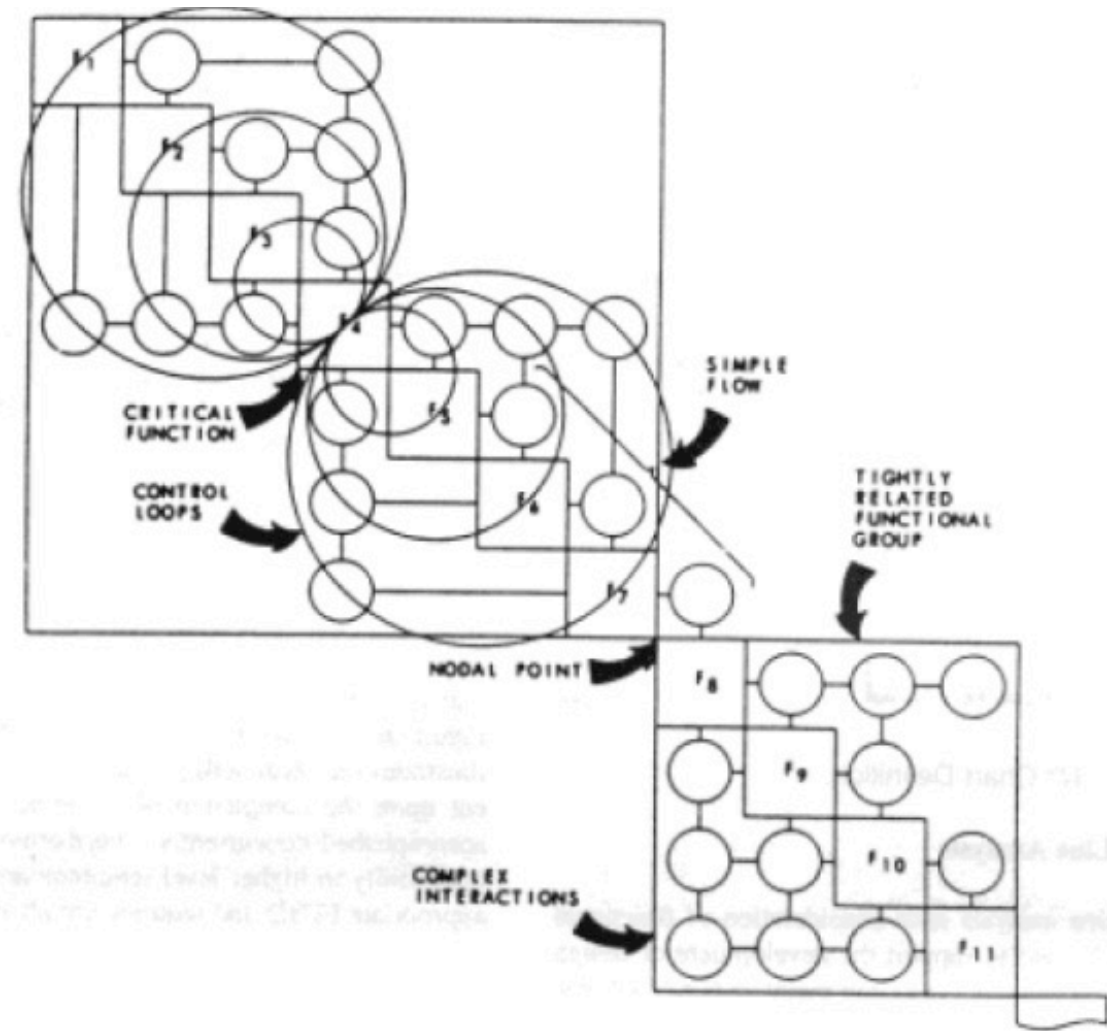


Figure B-7 — N² Chart Definition.



- Alternativamente, o uso de círculos e números permite uma listagem separada das interfaces de dados.
- O fluxo de dados no sentido horário entre funções que têm um loop de feedback pode ser ilustrado por um círculo maior chamado loop de controle.
- A identificação de uma função crítica, onde a função F4 tem um número de entradas e saídas para todas as outras funções no módulo superior.
- Existe um fluxo simples de dados de interface entre os módulos superior e inferior nas funções F7 e F8.
- O módulo inferior tem interação complexa entre suas funções.
- O gráfico N2 pode ser feito para níveis sucessivamente mais baixos para os níveis funcionais dos componentes de hardware e software. Além de definir os dados que devem ser fornecidos em toda a interface, o gráfico N2 pode identificar áreas onde podem surgir conflitos.





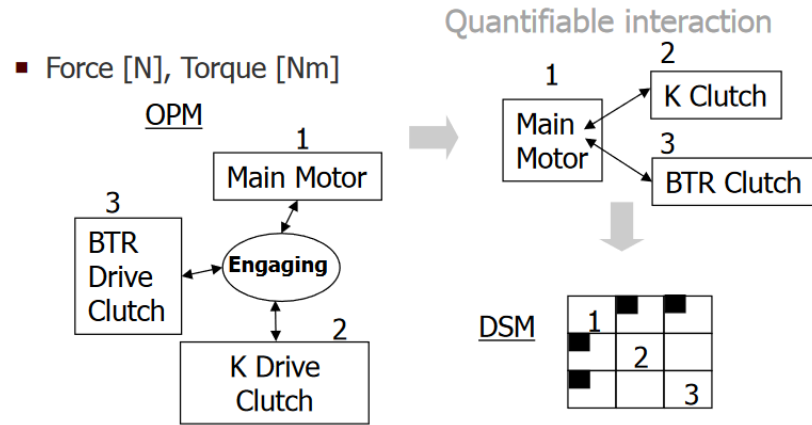
N2 É MUITO USADO PARA MAPEAR INTERFACES

Data types	Representation	Applications
Component-based (ProdBuct)	Component relationships	System architecting, engineering and design
People-based (Organization)	Organizational unit relationships	Organizational design, interface management, team integration
Activity-based (Process)	Activity input/output relationships	Process improvement, project scheduling, iteration management, information flow management
Parameter-based (low-level Process)	Design parameter relationships	Low level activity sequencing and process construction, sequencing design decisions

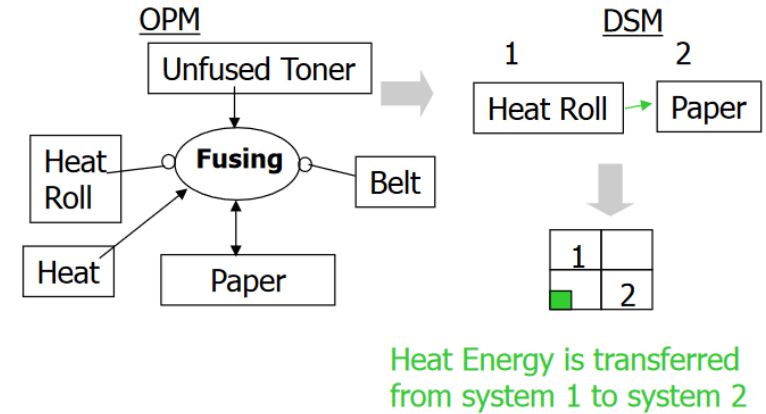


A ENGENHARIA DE SISTEMAS GERALMENTE LIDA COM 4 TIPOS DE INTERFACES

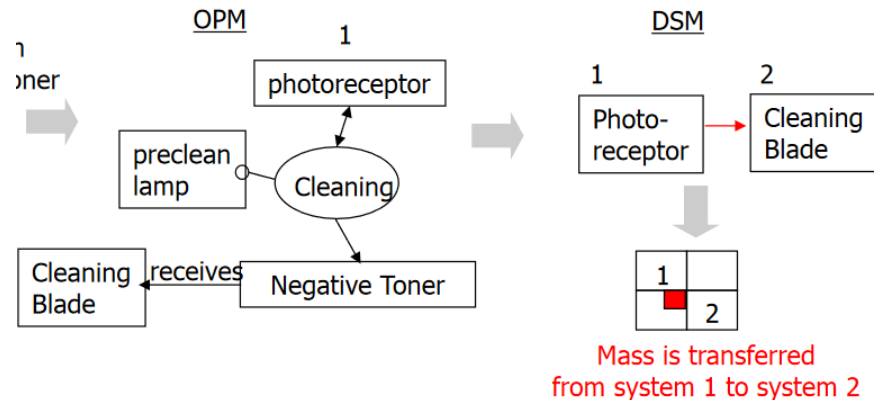
Physical Connection



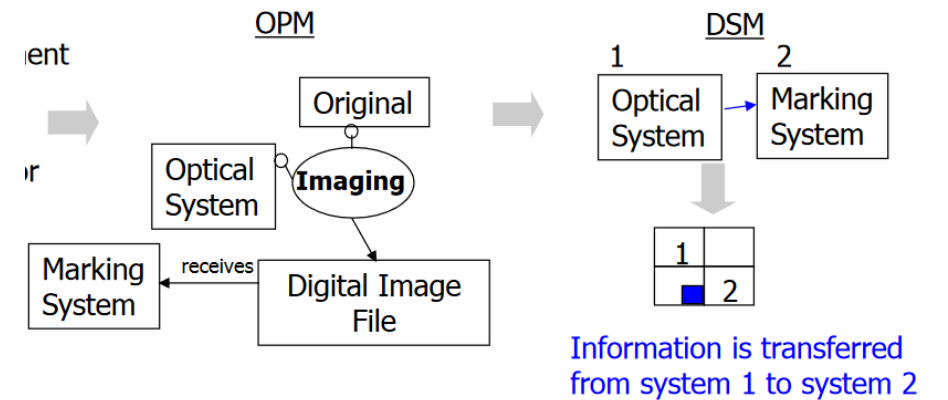
Energy Flow (electrical, thermal, RF, mechanical)



Mass Flow (fluids, gases, solids)



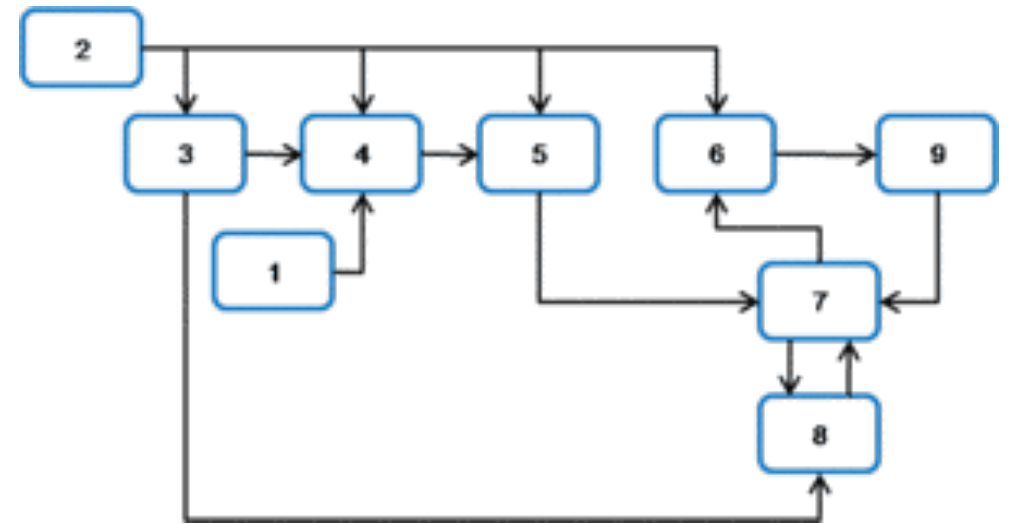
Information Flow (data)





MAS TAMBÉM LIDA COM TAREFAS

- Considere um conjunto de tarefas em um processo. Essas tarefas devem trabalhar juntas para cumprir o objetivo do processo geral.
- A troca de informações pode, portanto, ser representada como um dígrafo ou um DSM.
-





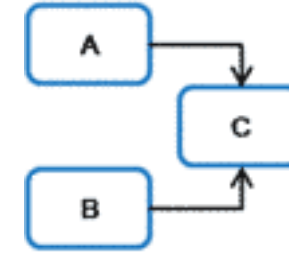
N2 SÃO USADOS PARA DEFINIR DESIGN STRUCTURE MATRIX (DSM)

- A pesquisa sobre o gerenciamento de complexidade baseado em matriz percorreu um longo caminho. Originando-se de um foco de processo com a primeira formulação publicada de uma **Matriz de Estrutura de Design (DSM) por Don Steward em 1981**, toda uma comunidade se desenvolveu em torno desta pesquisa.
- DSMs podem ter qualidades diferentes:
 - DSMs binários representam apenas a existência de uma relação,
 - enquanto os DSMs numéricos representam um valor numérico (também chamado de "peso") para representar a força de uma relação.
 - Os DSMs podem ser direcionados ou não dirigidos.



DESIGN STRUCTURE MATRIX (DSM)

- Por exemplo, considere um sistema que é composto de três elementos (ou subsistemas): elemento "A", elemento "B" e elemento "C".
- Um grafo pode ser desenvolvido para representar este sistema pictorialmente.
- O grafo do sistema é construído permitindo que um vértice/nó no gráfico represente um elemento do sistema e uma aresta unindo dois nós para representar a relação entre dois elementos do sistema.
- A direcionalidade da influência de um elemento para outro é capturada por uma seta em vez de um simples link.



	A	B	C
A			X
B			X
C			

- A representação matricial de um dígrafo (ou seja, grafo dirigido) tem as seguintes propriedades:
 - é binário (ou seja, uma matriz preenchida com apenas zeros e uns)
 - é quadrado (ou seja, uma matriz com igual número de linhas e colunas)
 - tem n linhas e colunas (n é o número de nós do dígrafo)
 - tem k elementos diferentes de zero, onde (k é o número de arestas no dígrafo)



BLOCOS DE CONSTRUÇÃO DO DSM

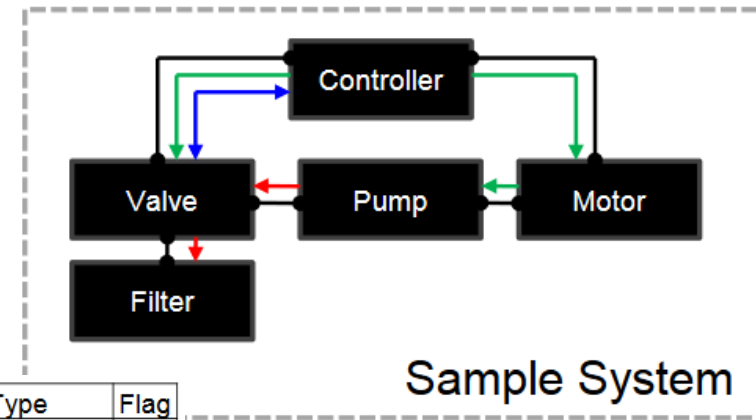
- Existem três blocos de construção básicos para descrever a relação entre os elementos do sistema: paralelo (ou simultâneo), sequencial (ou dependente) e acoplado (ou interdependente).

Configuration of Relationships	parallel	sequential	coupled
Graph and DSM representation	<p>The graph shows two separate nodes, A and B. The DSM is a 2x2 matrix with blue squares on the diagonal (A-A and B-B) and white cells elsewhere.</p>	<p>The graph shows node A pointing to node B. The DSM is a 2x2 matrix with blue squares on the diagonal (A-A and B-B), a blue square at (A, B), and a white square at (B, A).</p>	<p>The graph shows nodes A and B with bidirectional arrows between them. The DSM is a 2x2 matrix with blue squares on the diagonal (A-A and B-B), blue squares at (A, B) and (B, A), and white squares elsewhere.</p>



DESIGN STRUCTURE MATRIX

- Definição de Arquitetura: A incorporação do conceito e a alocação da função física/informacional (processo) aos elementos da forma (objetos) e definição de interfaces estruturais entre os objetos
-
- DSM captura a conectividade entre componentes => arquitetura
- O DSM fornece capacidade de análise não presente em um esquema tradicional



Number	Type	Flag
0	No Connection	0
1	Mechanical	
2	Flow	3
3	Information	7
4	Energy	15

Key

	Controller	Pump	Valve	Filter	Motor
Controller					
Pump					
Valve					
Filter					
Motor					

DSM



EXEMPLO

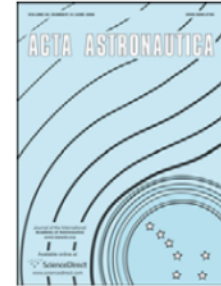
Acta Astronautica 66 (2010) 937–949



Contents lists available at ScienceDirect

Acta Astronautica

journal homepage: www.elsevier.com/locate/actaastro



An application of the Design Structure Matrix to Integrated Concurrent Engineering

Mark S. Avnet*, Annalisa L. Weigel

Massachusetts Institute of Technology, Cambridge, MA, USA



TROCA DE PARÂMETROS

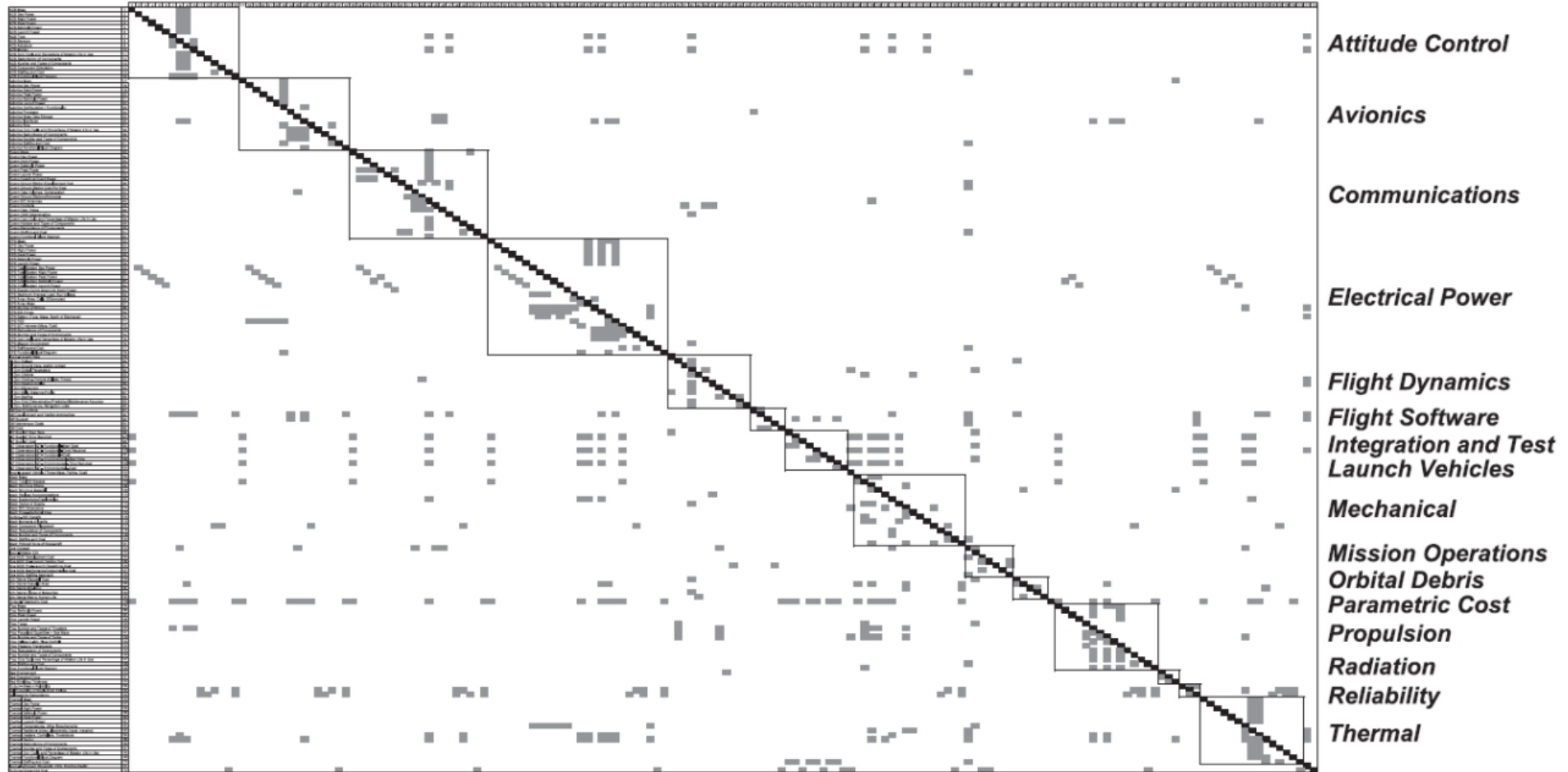


Fig. 2. Parameter-based DSM for the typical ICE design process. The DSM is organized as an alphabetical sequence of the 16 disciplines involved. The blocks along the diagonal encapsulate the work that is internal to each discipline, and the off-diagonal marks represent information flow across disciplines.



PROCESSO DE DESIGN

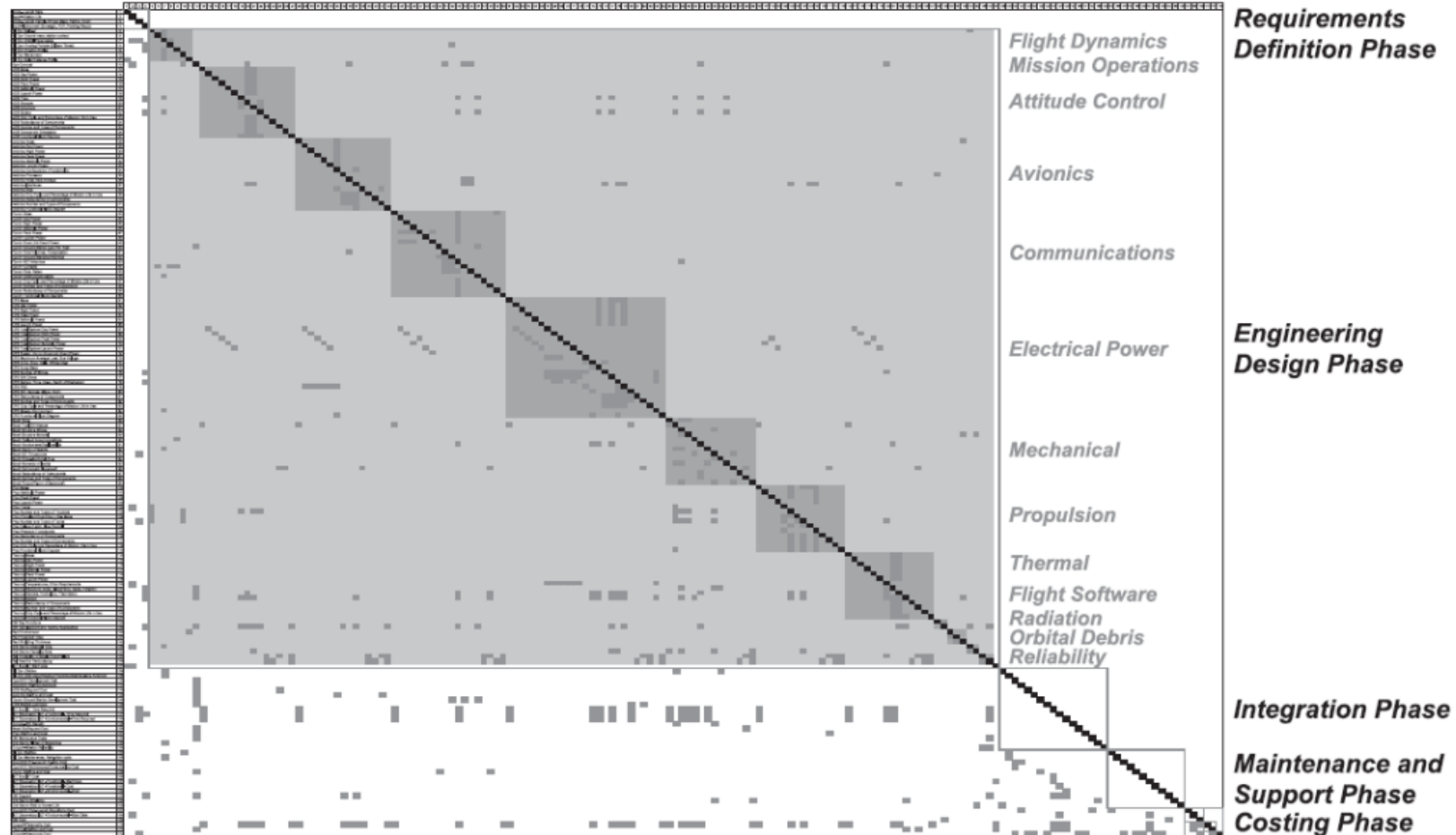


Fig. 3. Partitioned DSM for the ICE design process. The outlines show the phases of the design life cycle, the lightly shaded region highlights the single large coupled block corresponding to the Engineering Design Phase, and the darkly shaded blocks within the larger block contain the engineering tasks of each individual discipline.



DEPENDÊNCIAS ENTRE DISCIPLINAS

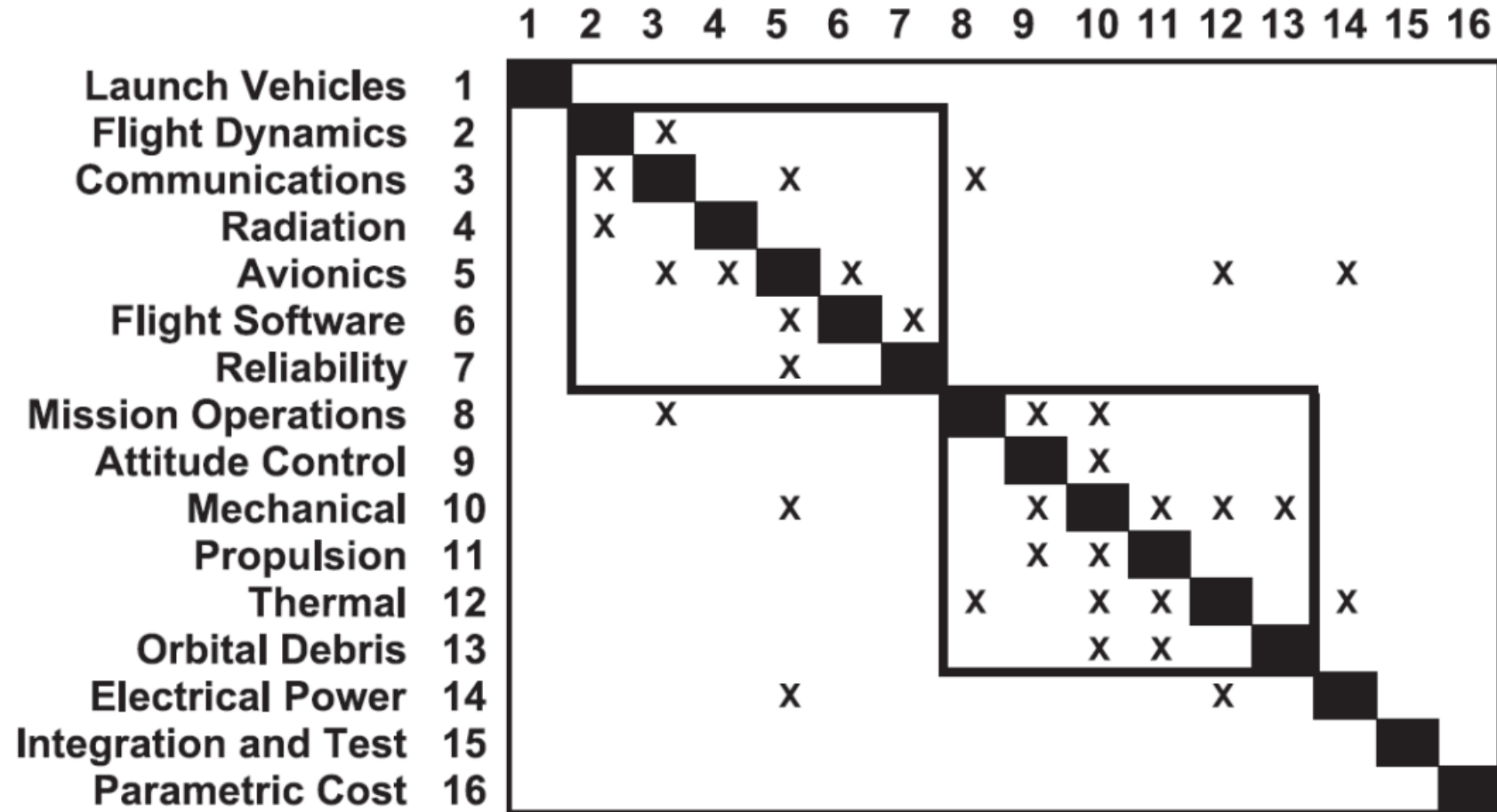


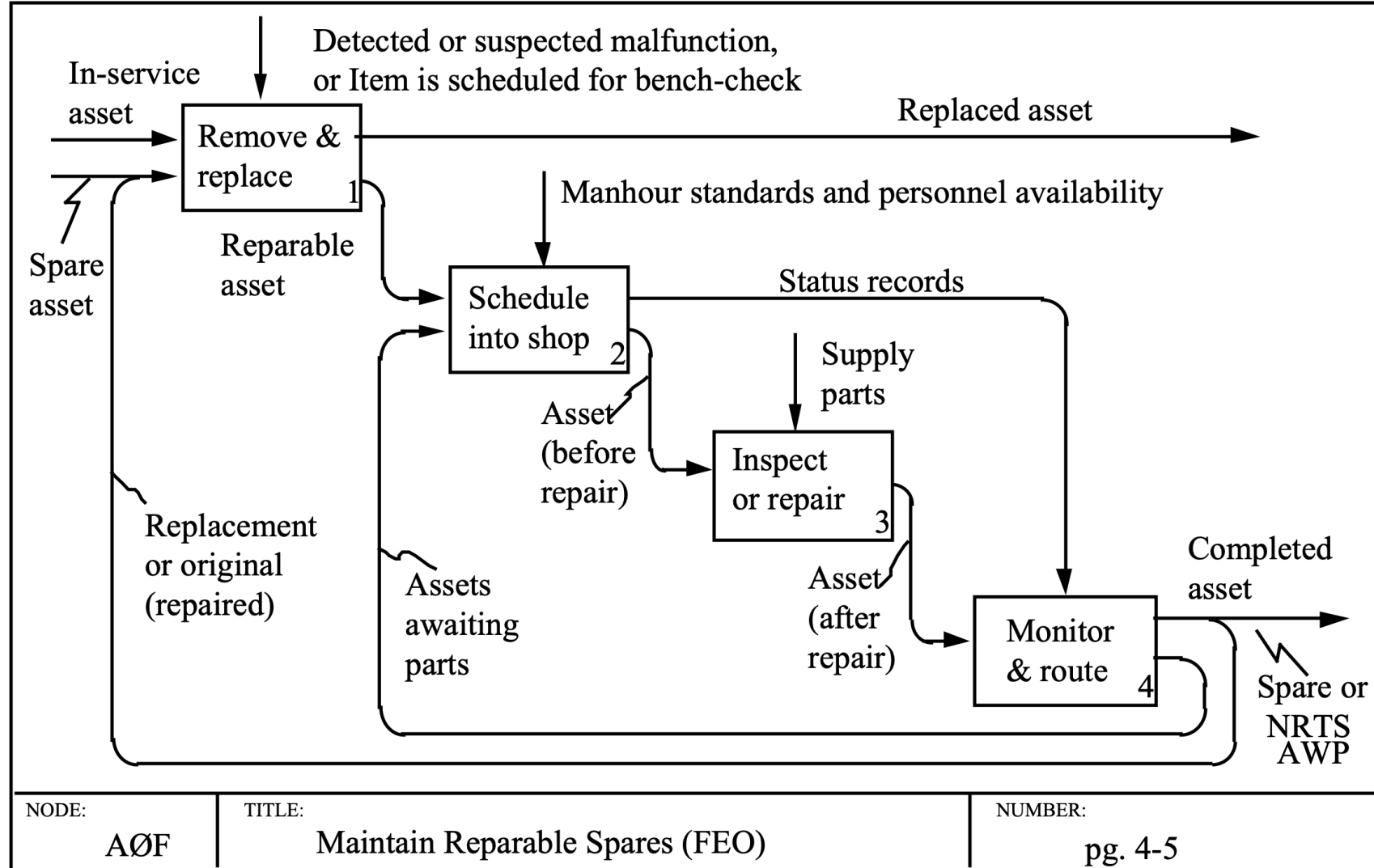
Fig. 4. Clustered team-based DSM for the ICE design process. Each mark indicates that the discipline in the row requires information from the discipline in the column due to one or more of the critical design trades (the loop types shown in Table 2).



IDEF0 DIAGRAM



IDEFO





INTEGRATION DEFINITION FOR FUNCTION MODELING (IDEFO)

- **IDEFO** está associado à metodologia estabelecida por Douglas T. Ross em 1974.
- Em paralelo, o programa **ICAM (Integrated Computer-Aided Manufacturing)** da Força Aérea dos EUA adotou a abordagem e ordenou uma definição para o domínio público chamada IDEF0 (1975).
-
- **IDEFO** ("Icam DEFinition for Function Modeling"), é uma metodologia de modelagem de funções para descrever funções de fabricação, que oferece uma linguagem de modelagem funcional para a análise, desenvolvimento, reengenharia e integração de sistemas de informação; processos de negócios; ou análise de engenharia de software.
- **IDEFO** faz parte da família IDEF de linguagens de modelagem no campo da engenharia de software e é construído sobre a **Structured Analysis and Design Technique (SADT)**.



BENEFÍCIOS RESULTANTES DO USO DO IDEF0

- **Identifica as necessidades:** Um bom modelo IDEF0 parecerá muito simples e fácil de entender. Necessidades e oportunidades de melhoria reveladas durante o esforço de modelagem podem parecer óbvias no início, mas teriam passado despercebidas de outra forma..
- **Constrói consenso:** A diversidade de conhecimentos, treinamento, habilidades e conhecimentos de uma equipe pode prejudicar a comunicação. O método de crítica do modelo leitor/autor pode estabelecer uma base comum bem definida para a compreensão por toda a equipe.



BENEFÍCIOS RESULTANTES DO USO DO IDEFO

- **Melhora a visão:** A imagem gráfica condensada fornecida pelo modelo apresenta uma linha de base instigante sobre a qual considerar melhorias. Um analista ou um planejador corporativo que considere o uso potencial de uma nova tecnologia ou novo método à luz dessa linha de base pode identificar oportunidades de aplicação específicas.
- **Fornece uma base para uma arquitetura aberta:** A estrutura do modelo pode ser usada para definir as interfaces entre os elementos do sistema e para identificar interfaces precisas para a definição de uma arquitetura de sistemas abertos. O modelo limita o escopo com precisão, mostrando onde o sistema modelado se encaixa no quadro maior.



BENEFÍCIOS RESULTANTES DO USO DO IDEFO

- **Suporta o controle de gerenciamento por meio de métricas:** O modelo pode servir como uma linha de base para custo, tempo, capacidade de fluxo e outras métricas relevantes para o projeto de um novo sistema. Se os detalhes do modelo não forem suficientes para anexar métricas específicas, a falta de detalhes deve ser usada para identificar onde é necessária mais decomposição. Detalhes mais finos na modelagem facilitarão uma medição mais precisa.
- **Define variantes para uso mais amplo do sistema de suporte:** Os sistemas de suporte que são aplicados a várias atividades no modelo podem se beneficiar de uma análise cuidadosa da funcionalidade real necessária em cada ponto de uso. Variantes e versões do sistema de suporte podem economizar custos consideráveis se o sistema de suporte puder ser ajustado para cada ponto de uso específico.



IDEFO EM POUCAS PALAVRAS

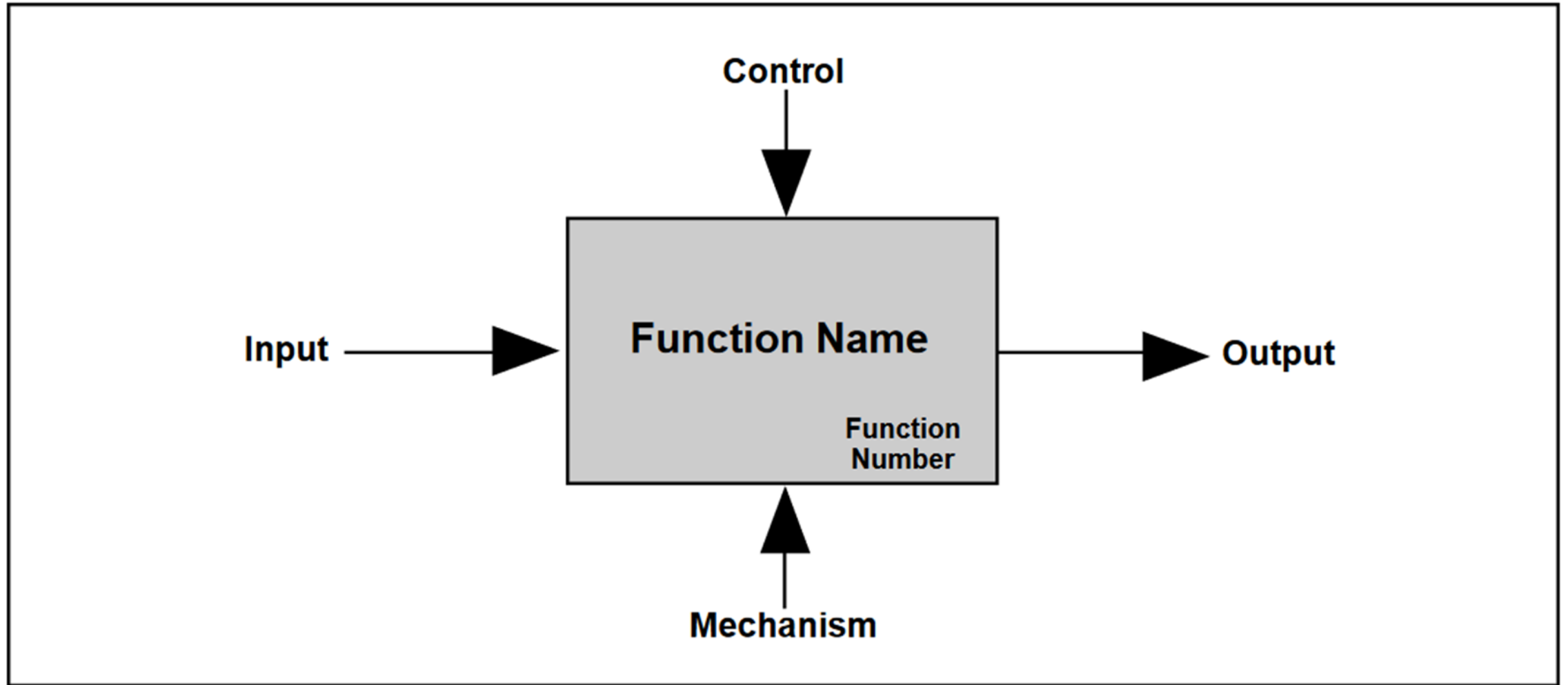
- IDEF0 é uma técnica de modelagem comum para a análise, desenvolvimento e integração de sistemas de informação; processos de negócios; ou análise de engenharia de software.
- O IDEF0 é usado para mostrar o fluxo de dados, o controle do sistema e o fluxo funcional dos processos do ciclo de vida.
- O IDEF0 é capaz de representar graficamente uma ampla variedade de negócios, manufatura e outros tipos de operações corporativas em qualquer nível de detalhe.

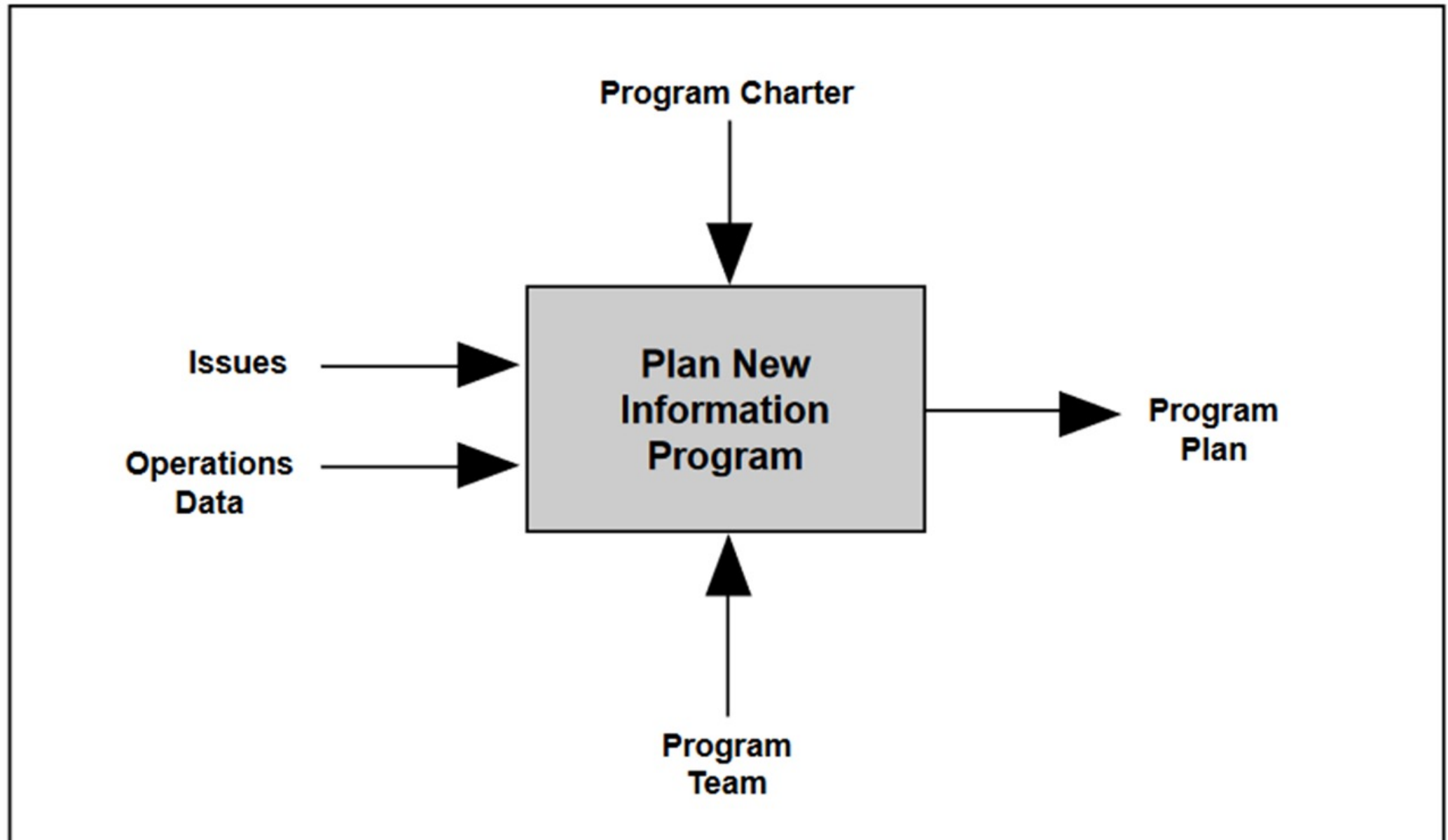
Ele fornece uma descrição rigorosa e precisa e promove a consistência do uso e da interpretação. Ele pode ser gerado por uma variedade de ferramentas.

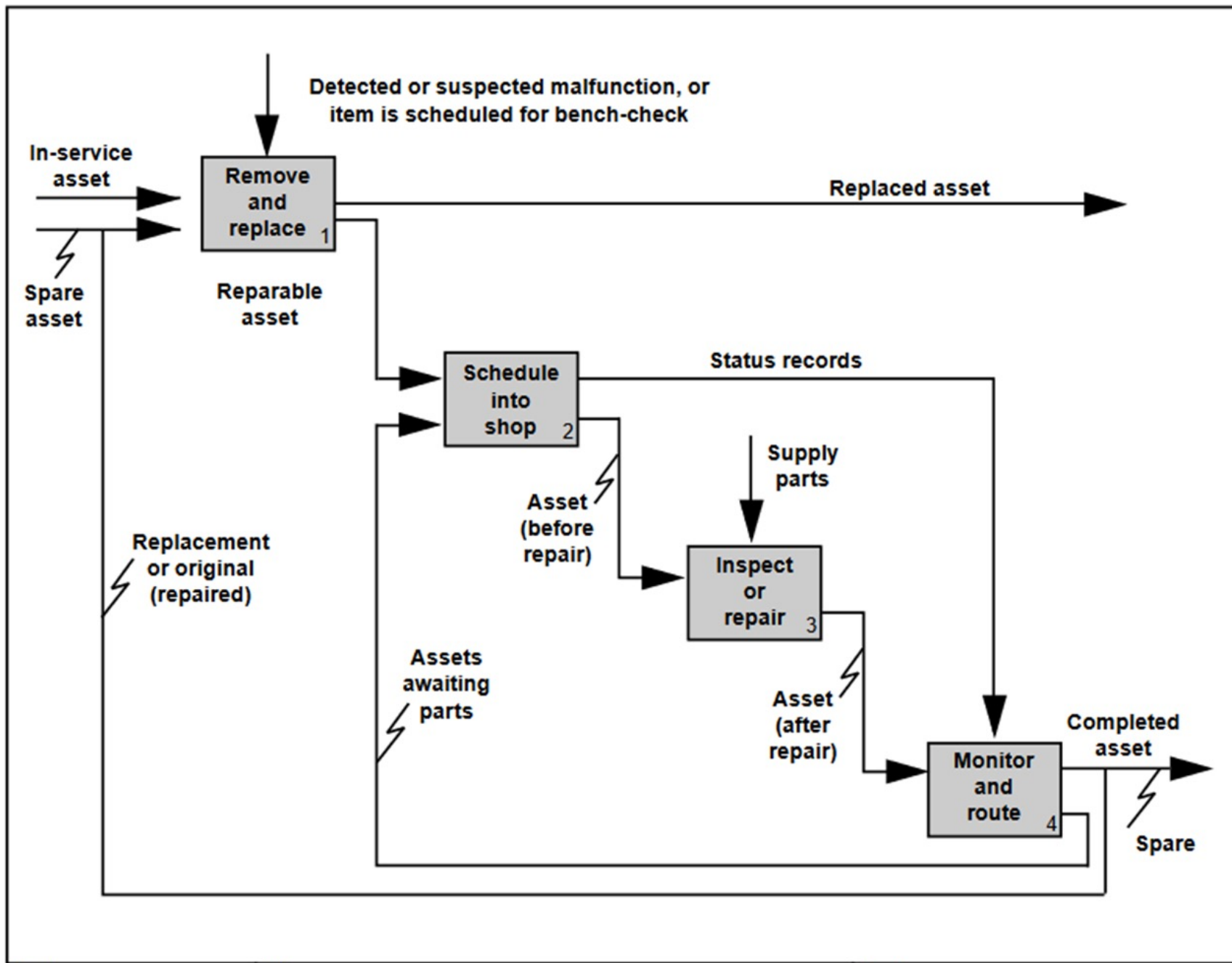


A POSIÇÃO NA QUAL A SETA SE CONECTA

- **Os controles entram na parte superior** da caixa.
- **As entradas, os dados ou objetos** atuados pela operação, digite a caixa a partir da **esquerda**.
- **As saídas da operação saem do lado direito** da caixa.
- **As setas do mecanismo** que fornecem meios de suporte para executar a função se unem (apontem para) a parte **inferior** da Caixa.



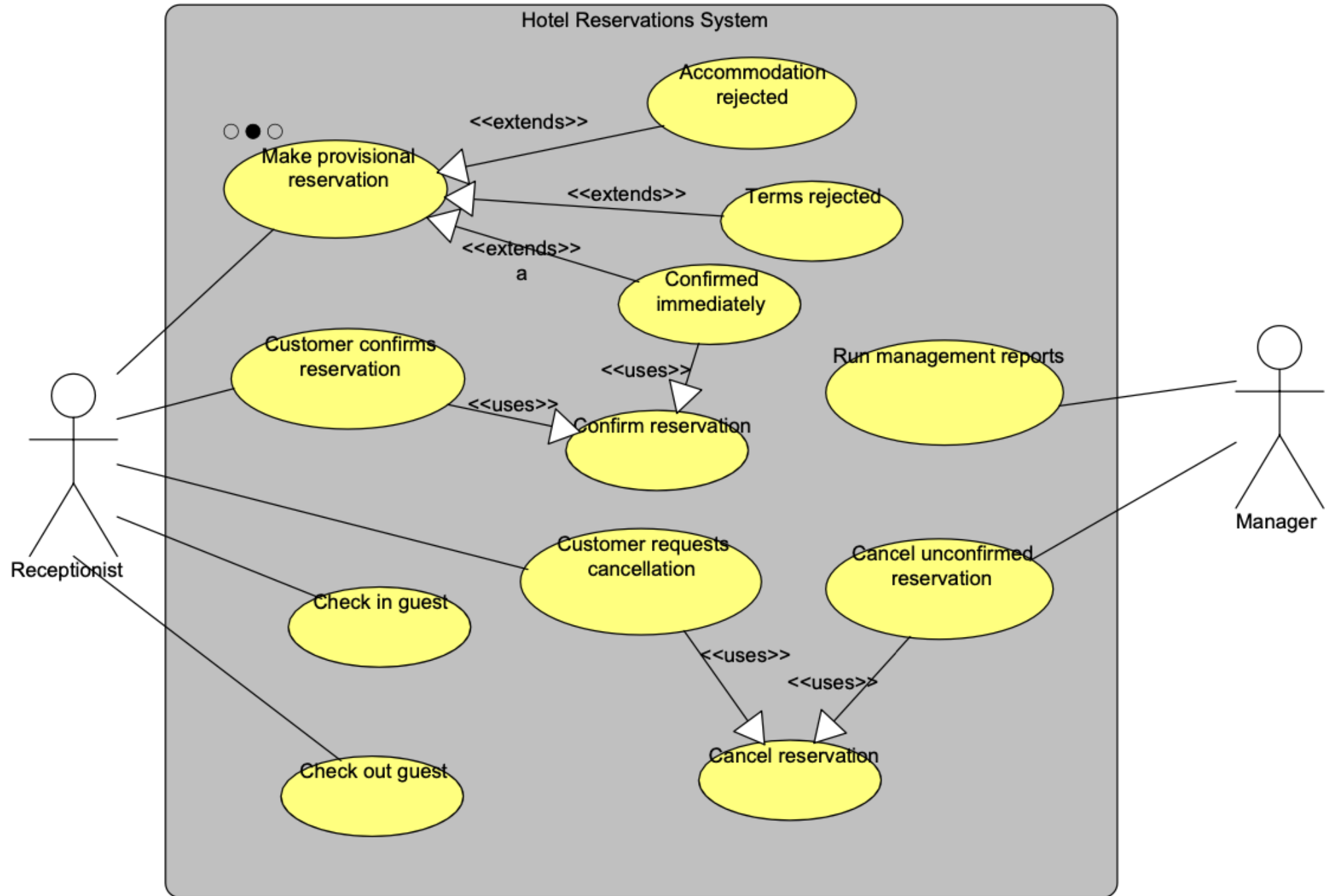






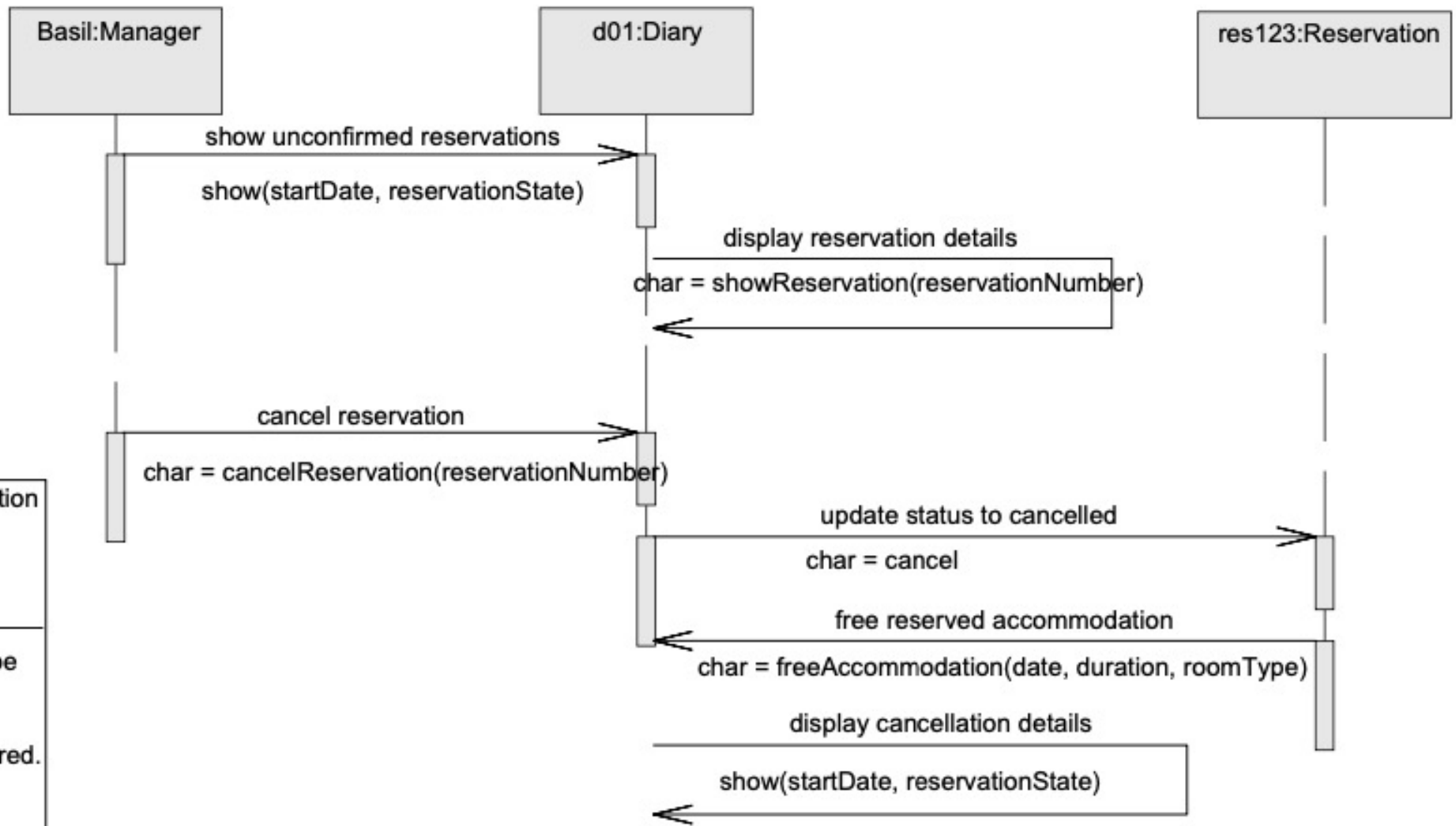
USE CASE

Veremos ao longo do curso





SEQUENCE DIAGRAM



Cancel unconfirmed reservation
(UML Sequence)
SA/2001
Wed Jun 16, 1999 13:44
-Comment-
Note: The customer MUST be notified if a reservation is cancelled because the confirmation period has expired.



BEHAVIOR DIAGRAM (BD)



BEHAVIOR DIAGRAM

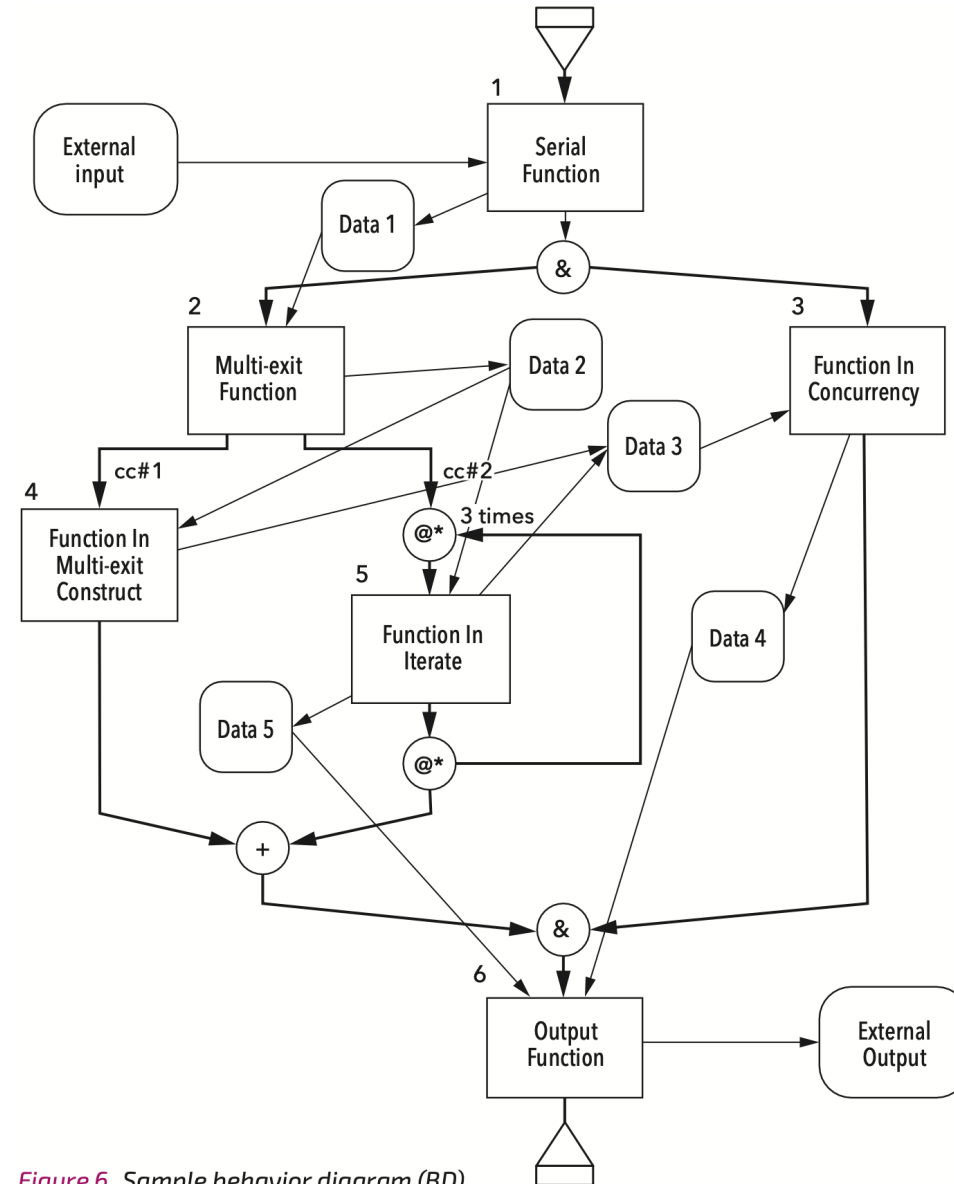


Figure 6. Sample behavior diagram (BD)

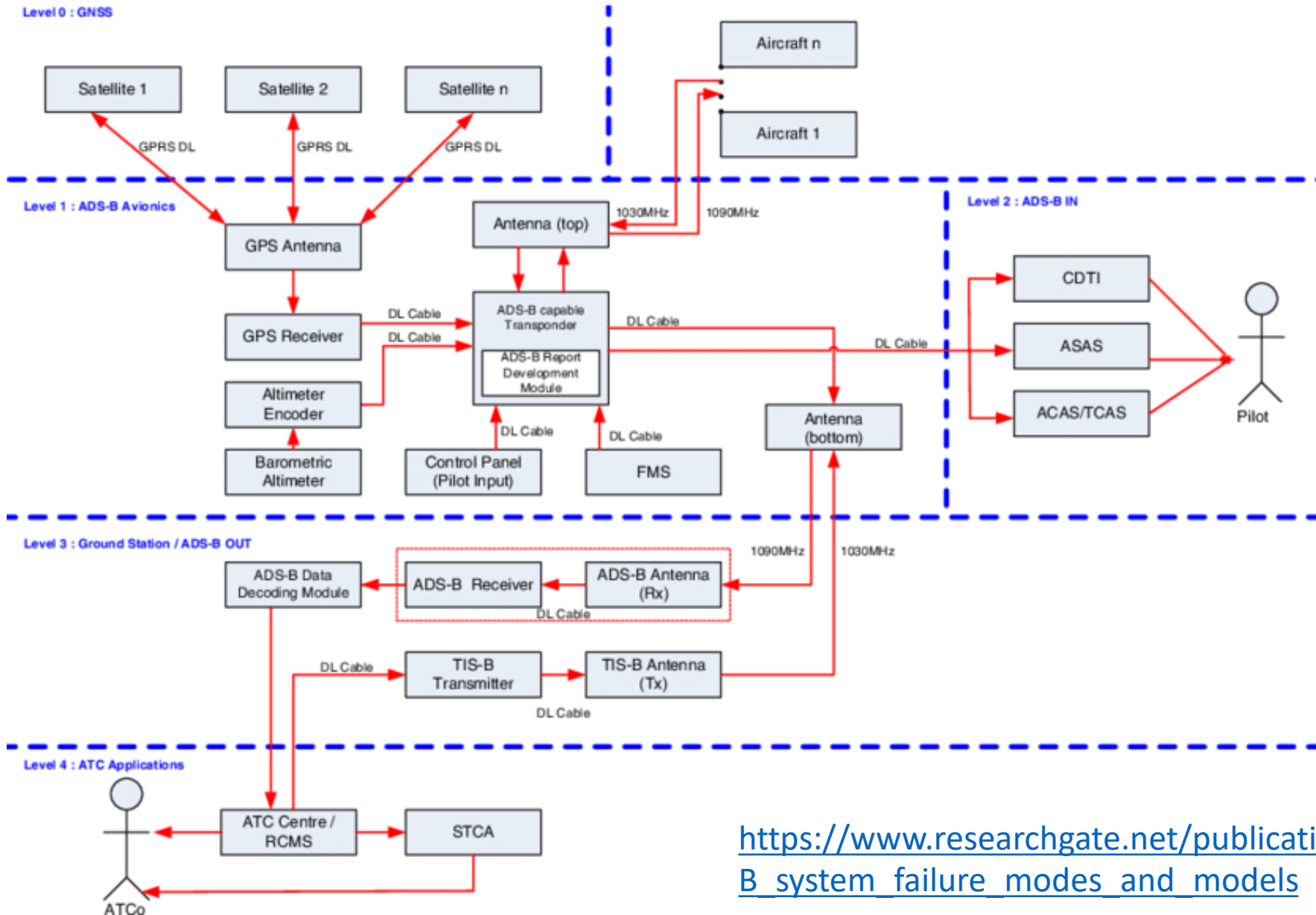


DIAGRAMA DE BLOCOS

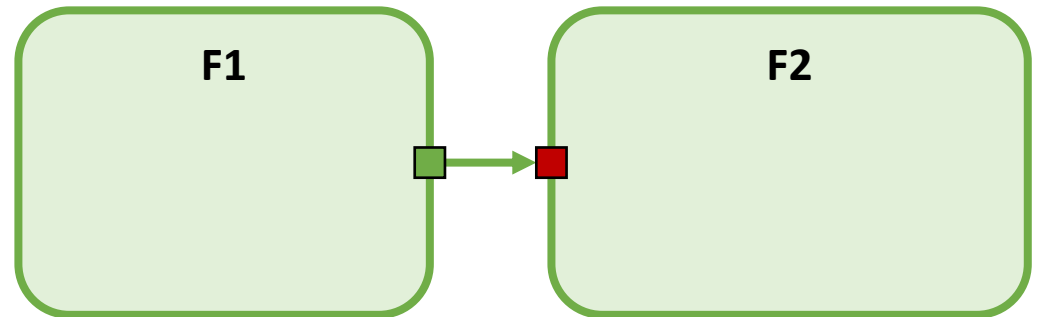
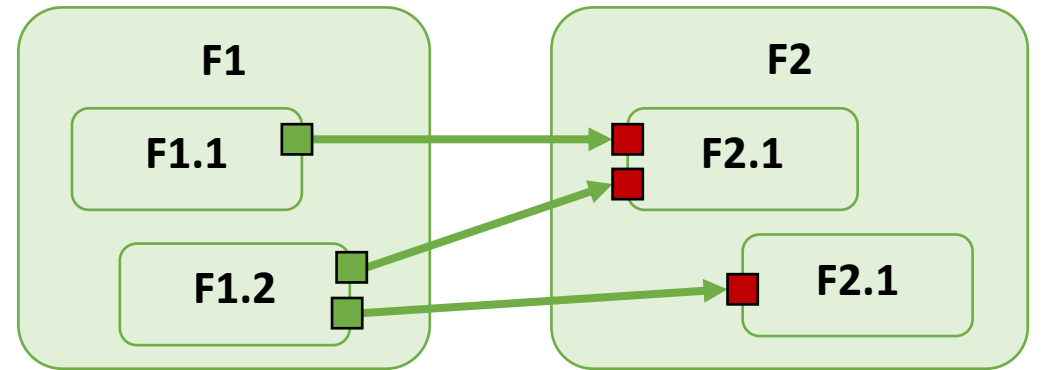
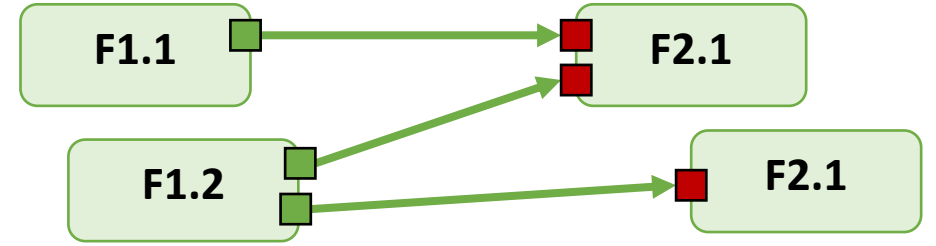
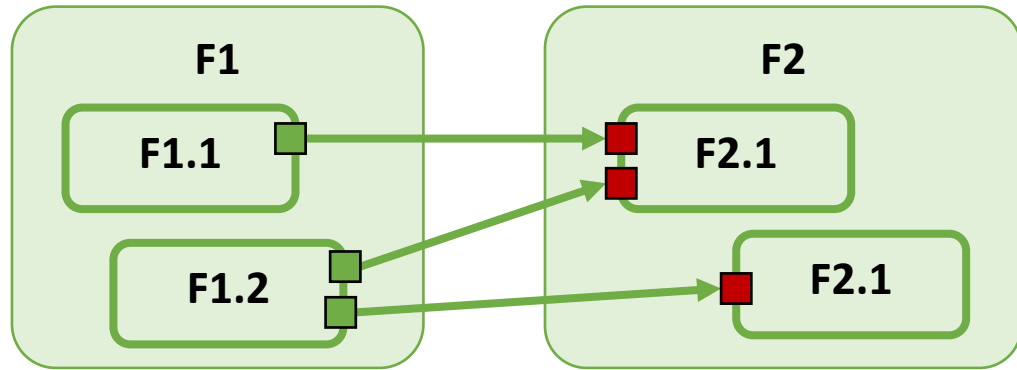
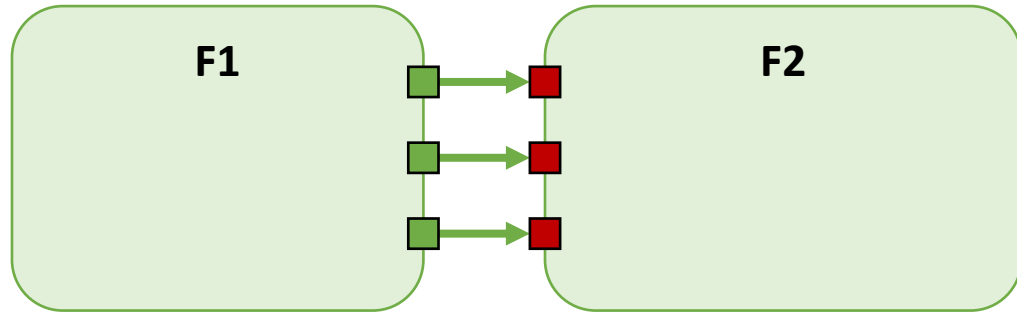
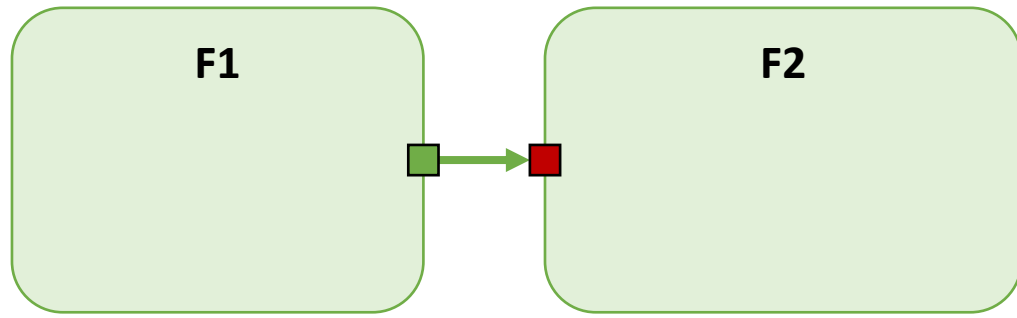
Block Diagram



DIAGRAMAS DE BLOCO



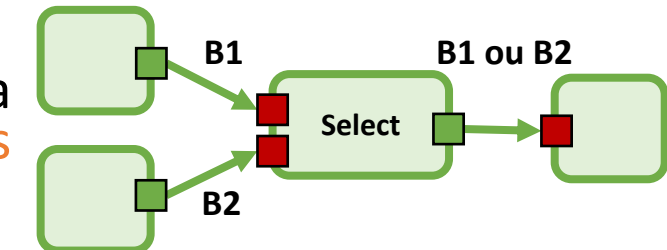
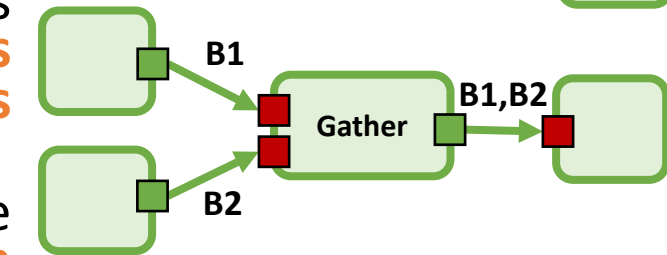
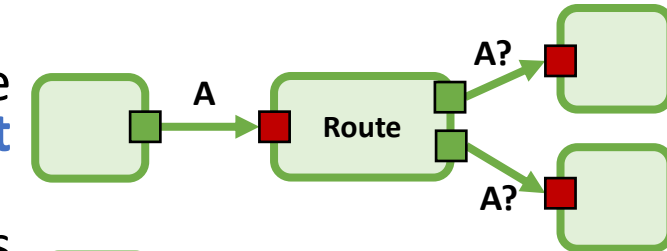
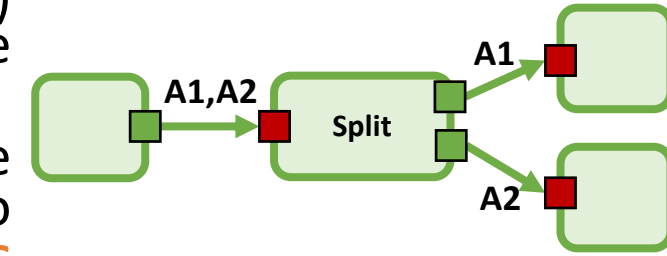
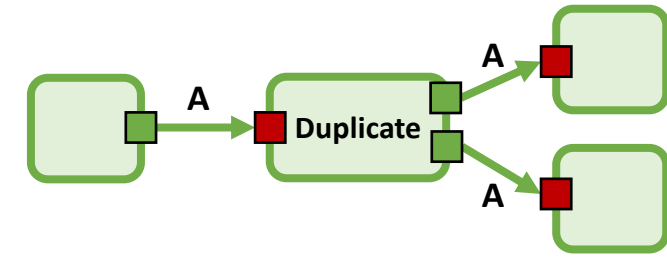
https://www.researchgate.net/publication/272037785_ADS-B_system_failure_modes_and_models





CONTROLES DE FLUXO

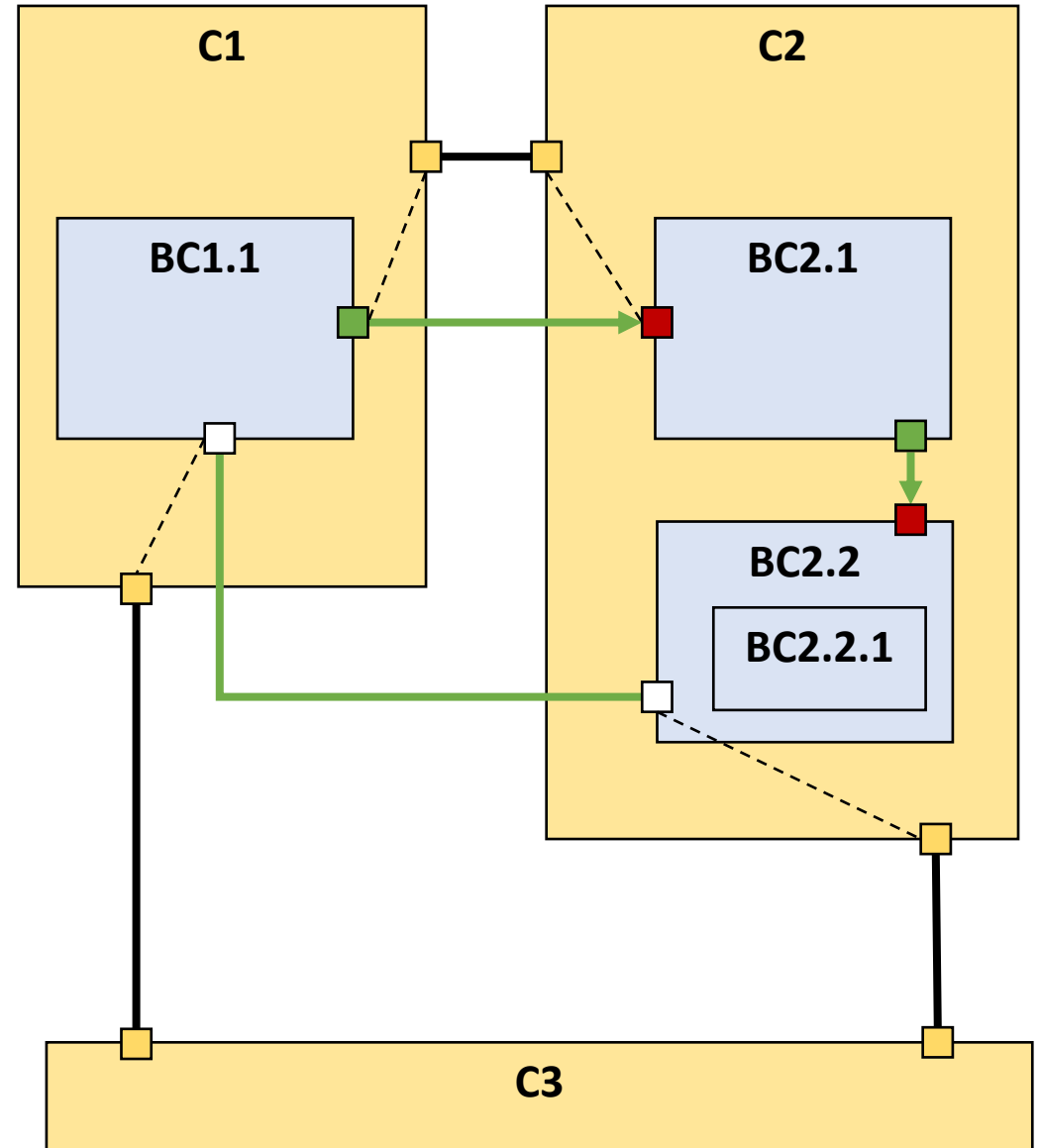
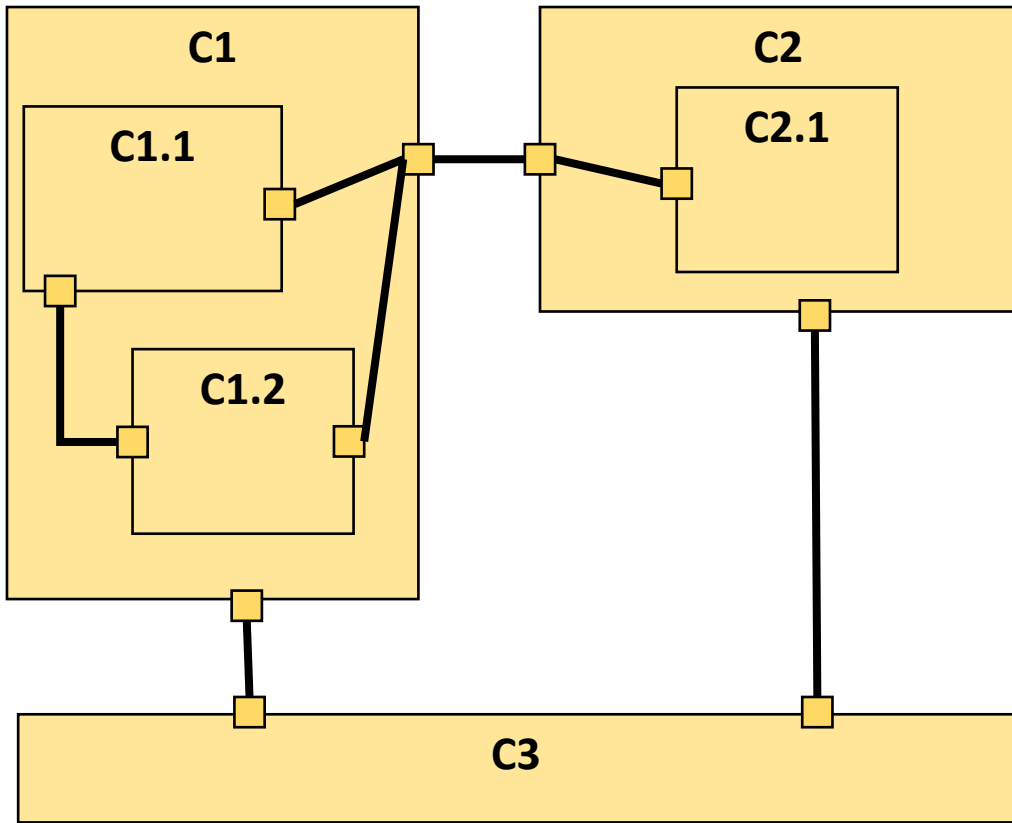
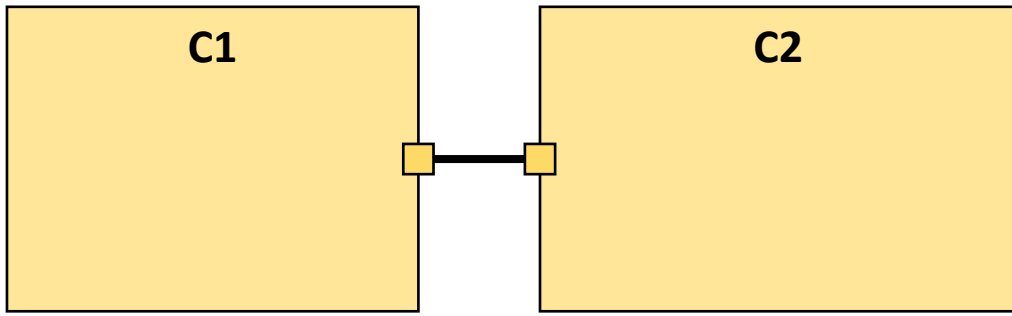
- Funções de controle de fluxo são intermediários entre a(s) fonte(s) e o(s) destinatário(s), responsáveis por controlar as condições de interação:
 - para especificar uma **difusão simultânea** de uma troca de origem para vários destinatários, definimos uma função **Duplicate** que **transmite os mesmos itens de troca para todos os destinatários**;
 - para especificar a **difusão simultânea** de alguns dos itens de troca para cada destinatário seletivamente, uma função **Split** que **roteia cada parte para um destinatário separado**;
 - para especificar a seleção de um entre vários destinatários potenciais, uma função **Route** que **transmite (na maioria das vezes sujeita a condições) para cada destino apenas alguns dos itens de troca recebidos**;
 - para especificar a combinação de itens de várias trocas de diferentes fontes, uma função **Gather** pode ser um **único item de troca combinando aqueles recebidos de diferentes fontes**;
 - para especificar a seleção de uma fonte entre muitas, uma função **Select** que **direciona apenas os elementos provenientes da fonte selecionada (na maioria das vezes sujeita a condições)**





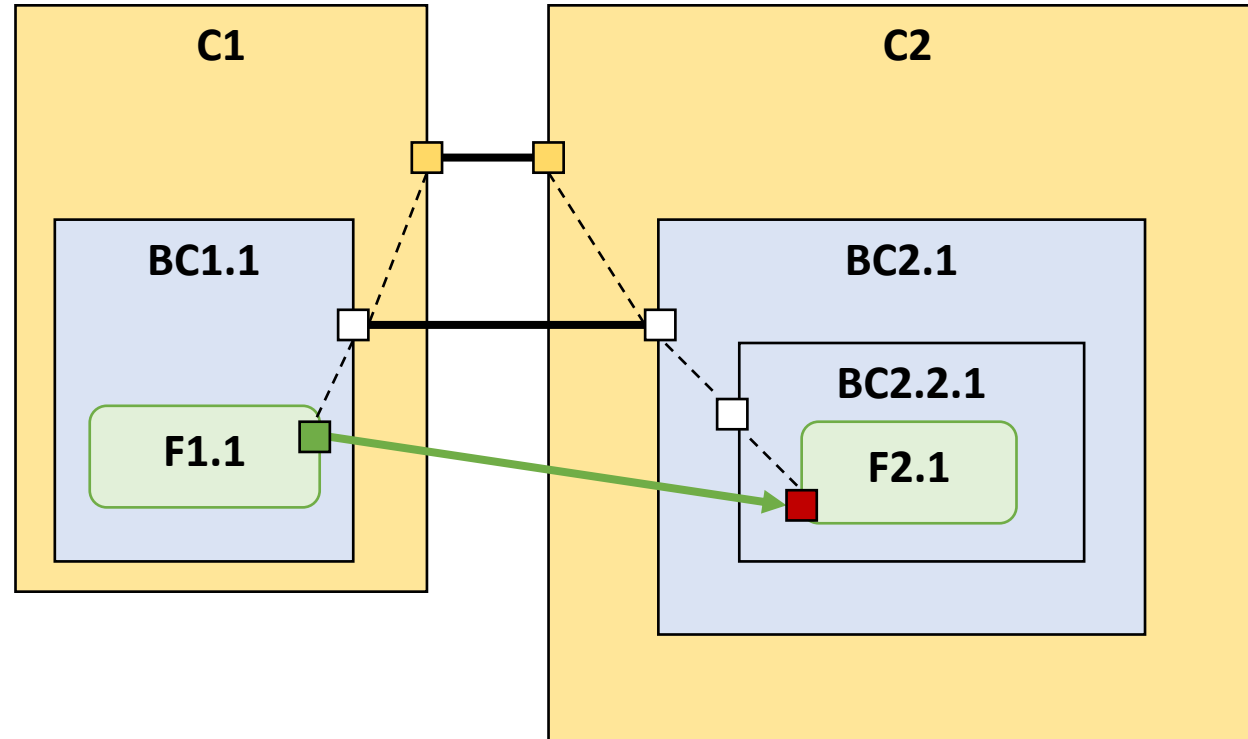
ORGANIZAÇÃO DAS FORMAS

- O **sistema** é um conjunto ordenado de elementos que funcionam como um todo, respondendo à demanda e necessidades do cliente e do usuário, e sujeitos à engenharia apoiada pela Arcadia.
- Um **ator** é uma entidade que é externa ao sistema (humano ou não), interagindo com ele, especialmente através de suas interfaces.
- Um **componente** é uma parte constituinte do sistema, contribuindo para o seu comportamento e/ou propriedades, juntamente com outros componentes e atores externos ao sistema.
 - Um componente **pode ser dividido em subcomponentes**.
 - Para generalizar, um componente também pode ser alocado a um ator, para definir suas interações e conexões com o sistema ou outros atores.
- Um **componente comportamental** é um componente do sistema, responsável por realizar algumas das funções devolvidas ao sistema, interagindo com seus outros componentes comportamentais e atores externos.
- Um **componente físico** hospeda um componente comportamental, fornecendo-lhe os recursos necessários para funcionar e interagir com seu ambiente..
 - Um **componente comportamental é hospedado por um e apenas um componente físico**.





INSTANCIANDO FUNÇÕES EM BLOCOS





ITENS DE TROCAS (EXCHANGE ITENS)

- Um item de troca é um **conjunto ordenado de referências a elementos roteados** juntos, durante uma interação ou troca entre funções, componentes e atores.
- Os itens são roteados simultaneamente, nas mesmas condições, com as mesmas propriedades não funcionais. Esses itens são chamados de dados e são caracterizados pela **classe** à qual pertencem.
- **Um item de troca é definido por:**
 - um nome;
 - A lista de elementos do item de troca; cada elemento é definido no item de troca por um nome, e a classe à qual ele pertence, e se a troca é bidirecional, a direção de transmissão (por convenção, "in" na direção da troca por padrão, "out" na direção oposta, ou "in/out");
 - a descrição das condições de comunicação, se necessário, por exemplo, serviço, mensagem, evento, fluxo de dados, dados compartilhados, fluxo de material, quantidade física, etc.



TROCAS FUNCIONAIS



- **Pelo menos um item de troca** deve ser alocado para cada porta funcional em uma função para caracterizar o conteúdo que a função pode produzir ou que ela precisa.
 - Este item de troca **pode ser compartilhado por várias portas**.
- Se uma porta transporta **vários itens de troca, então precisamos especificar, em cada uma das trocas funcionais conectadas a ela, o(s) item(ns) realmente roteado(s)**, que deve ser coerente com os das portas conectadas pela troca. Além disso, por conveniência, é possível começar alocando um item para uma troca, antes de propagá-lo para as portas conectadas a ele.
- **Recomenda-se definir apenas um único item em cada troca funcional.**

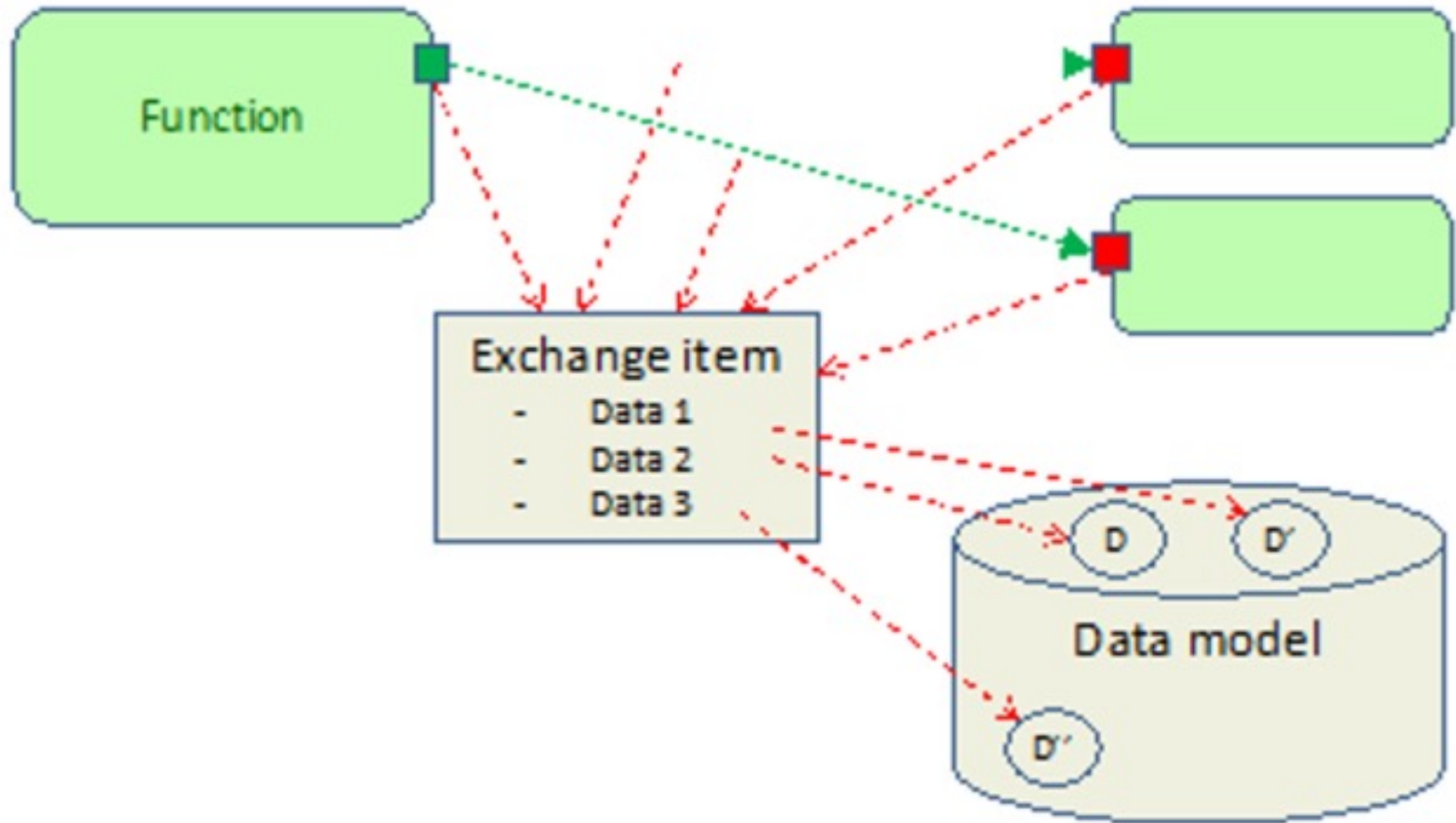
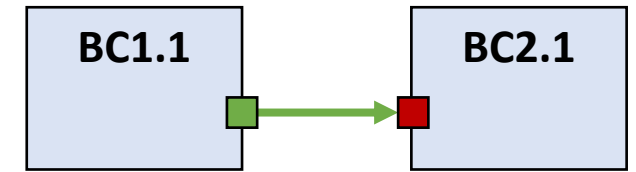


Figure 21.2. Allocation of exchange items to functional ports and exchanges



TROCAS ENTRE COMPONENTES COMPORTAMENTAIS (CONCEITUAIS)



- O conteúdo de uma troca entre componentes comportamentais já é definido pelos itens de troca transportados pelas trocas funcionais que ela implementa.
- **Uma interface** é um conjunto de itens de troca que permite que dois componentes (e o sistema e os atores), se comuniquem, de acordo com um "contrato" de comunicação compartilhado entre eles..
- **Várias interfaces podem ser agrupadas em uma única interface.**

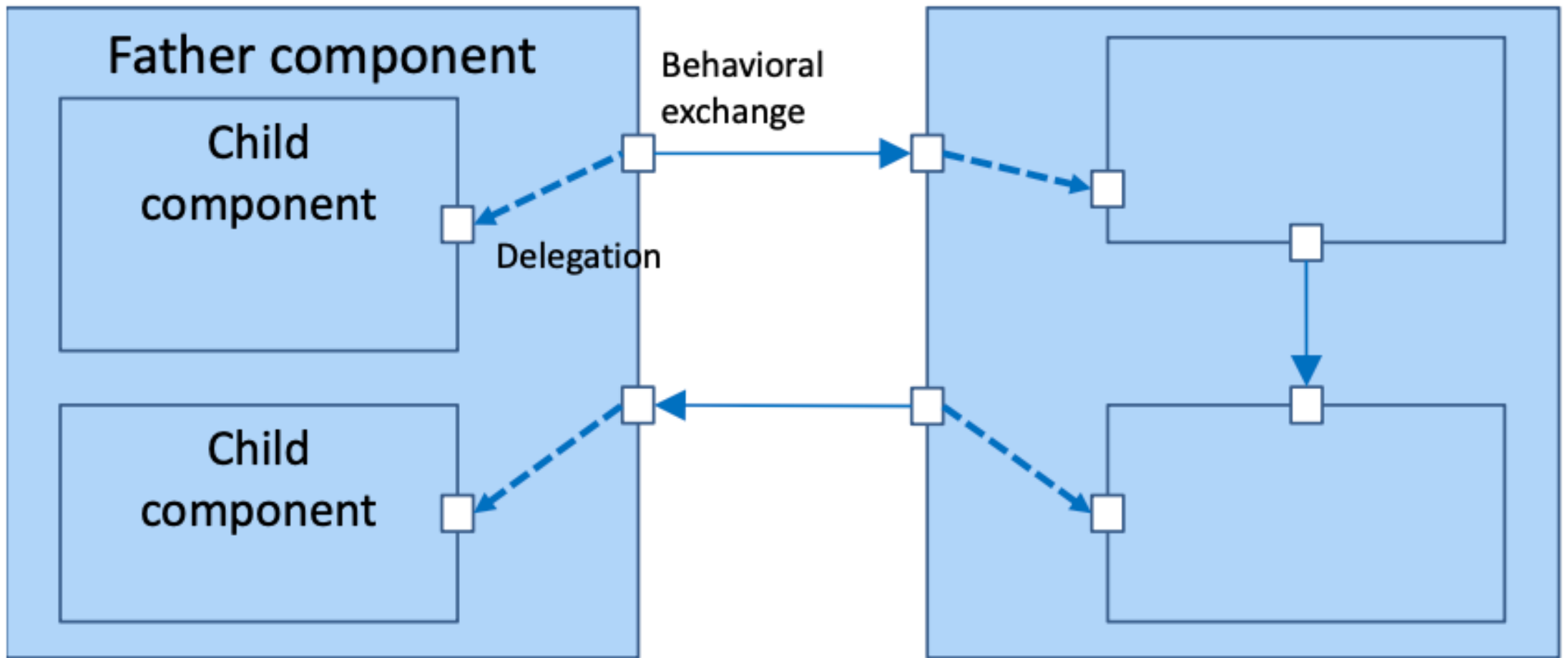
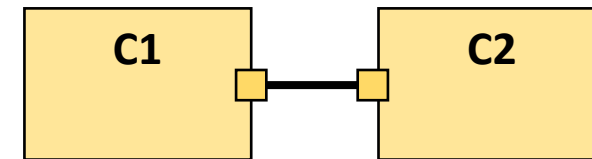


Figure 19.2. Behavioral components, ports, exchanges and delegation



TROCAS ENTRE COMPONENTES FÍSICOS



- Um **link físico** é um meio de comunicação, transporte ou roteamento entre dois componentes físicos de hospedagem, usado como **suporte para trocas comportamentais**.
- Um link físico (interface) vincula duas portas físicas de dois componentes. Ele pode vincular diretamente alguns subcomponentes uns aos outros (através de suas portas).
- **Um link físico faz referência às trocas comportamentais que transporta.**

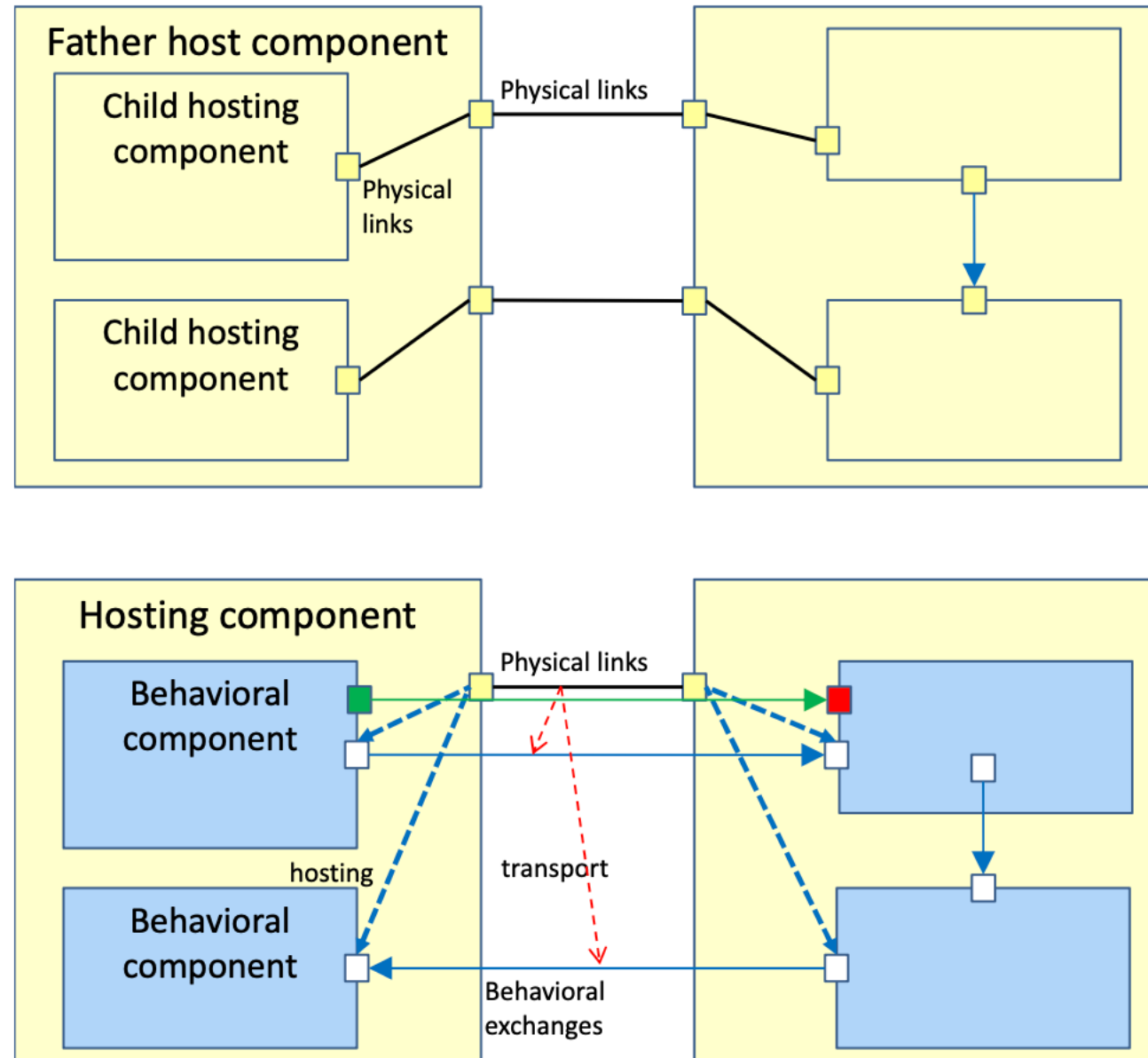


Figure 19.3. Breakdown of a hosting component and hosting of behavioral components



CONSIDERAÇÕES FINAIS



ATIVIDADES

- AI-02 - Leitura/Resumo paper sobre representações clássicas.
 - Lembrem que o exercício é para ser escrito (papel/tablet).
 - Façam uma página com um mapa mental sobre cada tipo de diagrama contendo a natureza do que o diagrama representa, origem do diagrama.
 - Em uma segunda/terceira página, exercitem os diagramas (eFFBD, DFD, N2, BLOCOS) para dois exemplos:

