Establishment of a methodological language for Systems Engineering

[2024]

Prof. Dr. Christopher S. Cerqueira

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

Note:

- Last two weeks we visited several universities that are going to became partners into the Systems Engineering Topic.
  - Harvard / MIT / George Mason University / Illinois State University / Purdue
- I'll lecture next year joint courses with Purdue / GMU – so...
- I'm changing (again) all my material to English (last year's in portuguese is available at my website).

Summary

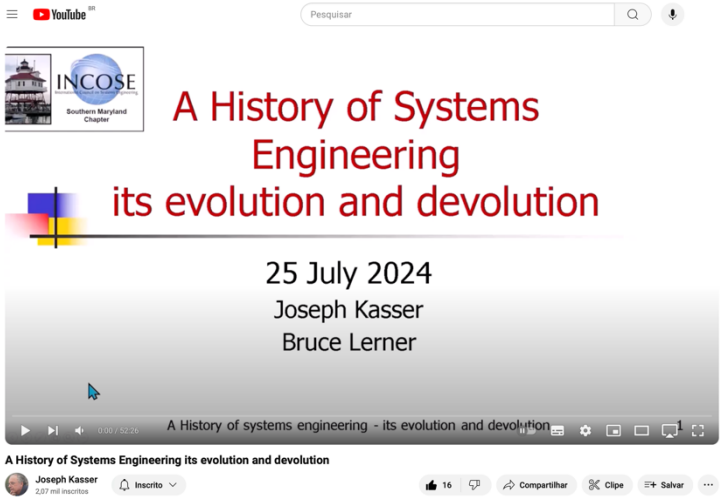

Motivation
Language Perspective
Digitalization of the Activities
Model Based Systems Engineering
Creating a system model
OPM
ARCADIA
Last Considerations

Motivation

Teasing: https://www.youtube.com/watch?v=Gmnc-78TUuM

Nine perspectives:

1. The introductory phase: the **early systems and the start of systems engineering** postgraduate education
2. Changes in the **definitions** of systems engineering: the changes in a sample of definitions of systems engineering between the 1950's and 2024.
3. Changes in the **application** of the systems approach in systems engineering: starting with the General Systems Theory, the changes in the meaning of the systems approach.
4. Changes in the systems engineering **tools**: a look at the tools of systems engineering in the 1950's and 1960's and different set of tools in the early 2002's before Model-based Systems Engineering (MBSE).
5. Changes in the systems engineering **roles**: a look at how the roles of the systems engineer changed between 1969 and 2024 with samples from1969, 1988, 1994, 1996, 1997, 2000, 2017, 2019 and 2024.
6. The two systems engineering **paradigms**: a brief overview of the original "A" paradigm and the devolved currently widely-practiced "B" paradigm which seems to have burst on the scene in the 1990's.
7. The early "**Standards**" for systems engineering: a brief look at MIL-STD 499, EIA-632, IEEE-1220 and ISO-IEC 15288:2002 showing why they are not actually standards for the performance of systems engineering.
8. Historical sketch of INCOSE: highlights a few milestones from its beginning as the **National Council on Systems Engineering** to the introduction and singular focus on MBSE.
9. The nine perspectives of systems engineering: shows how the differences in the contents of textbooks, and journal and conference papers were grouped into nine perspectives.



SPICY SE

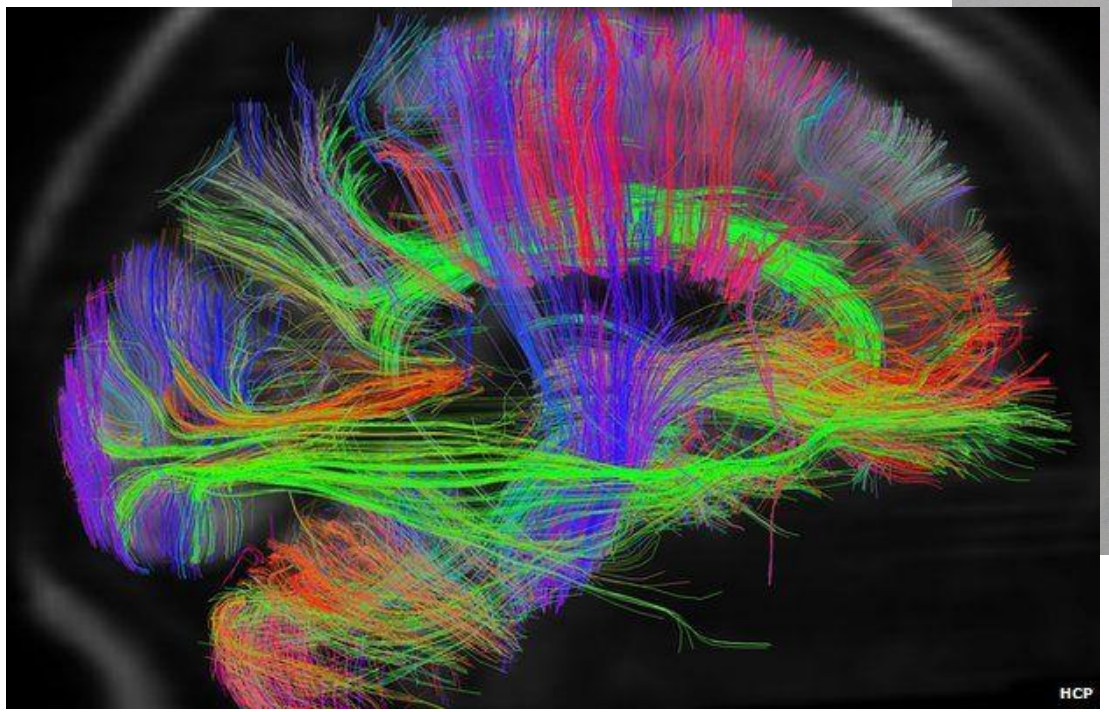
"Systems Engineering tools look like they are from the 90s!"

https://spicy-se.com

Language Perspective

Reducing to a function - we are standard recognizers

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

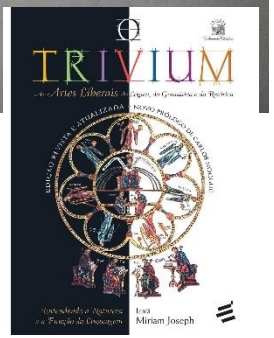


feita de tijolos, pedras ou de

FUNÇÃO DA GRAMÁTICA

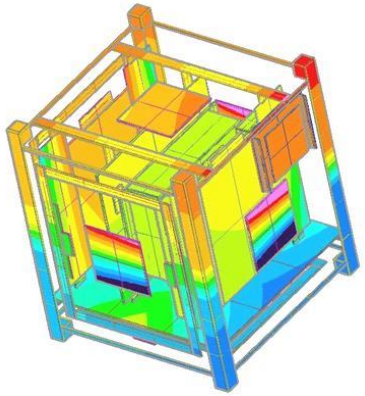
A função fundamental da gramática é estabelecer leis para relacionar símbolos de modo a expressar pensamento. Uma frase expressa um pensamento – uma relação de ideias – numa declaração, numa pergunta, numa ordem, num desejo, numa prece ou numa exclamação. Símbolos categoremáticos são aqueles que são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

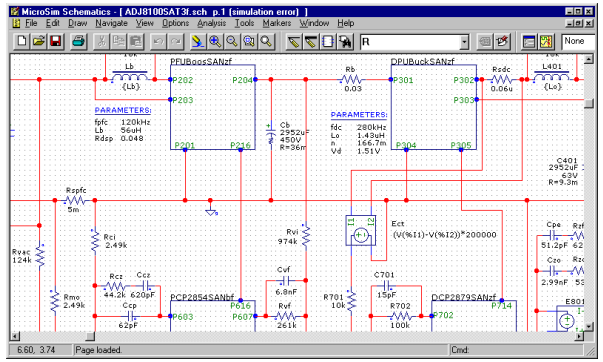


Each engineering has its own language

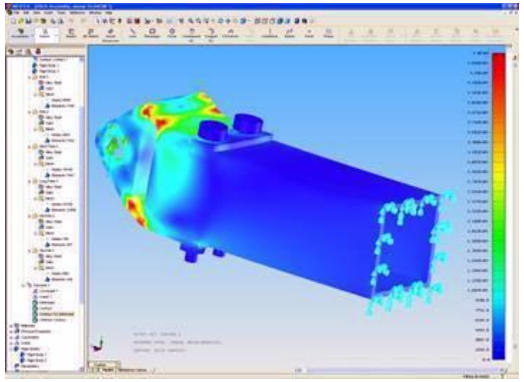
Thermal Eng.



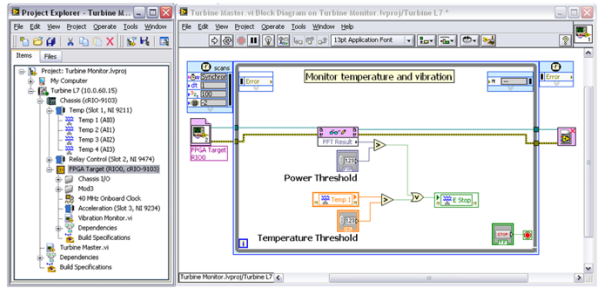
Electrical Eng.



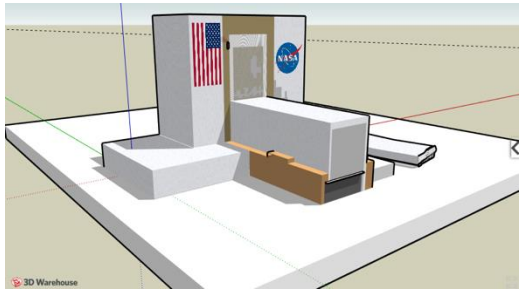
Mechanical Eng.



Control Eng.

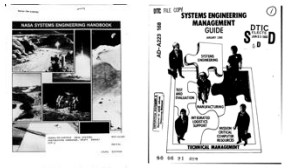


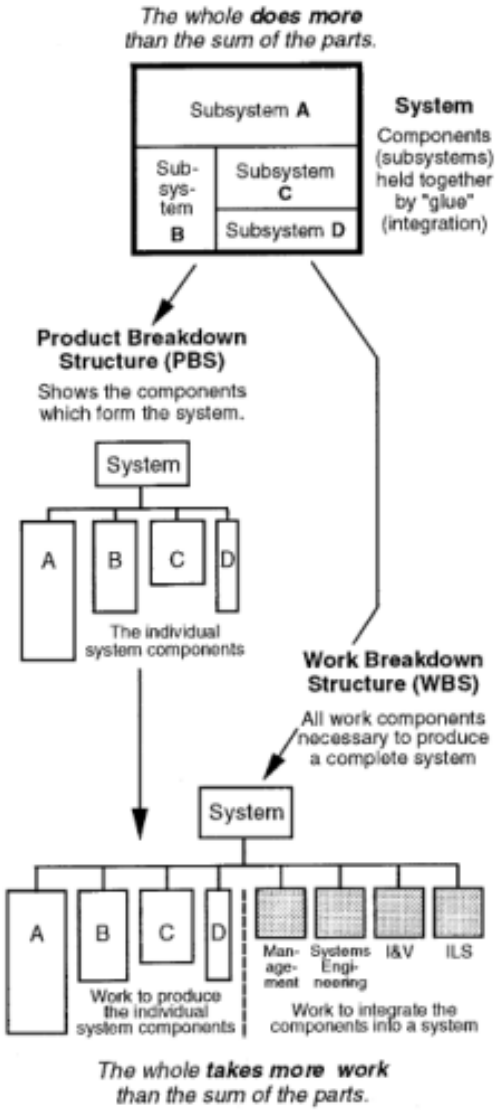
Infrastructure Eng.

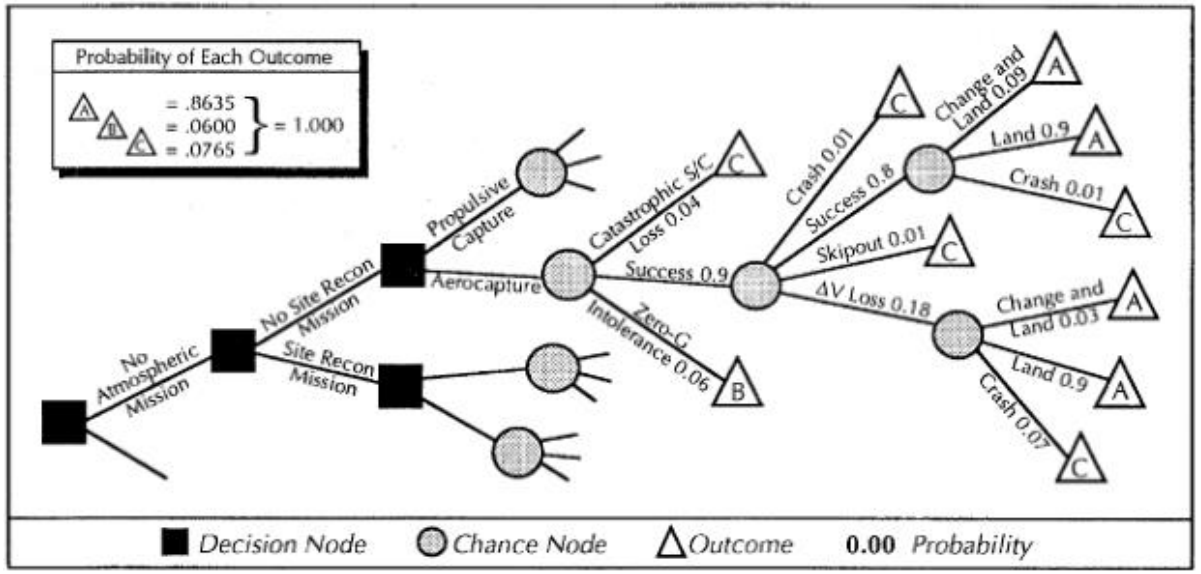


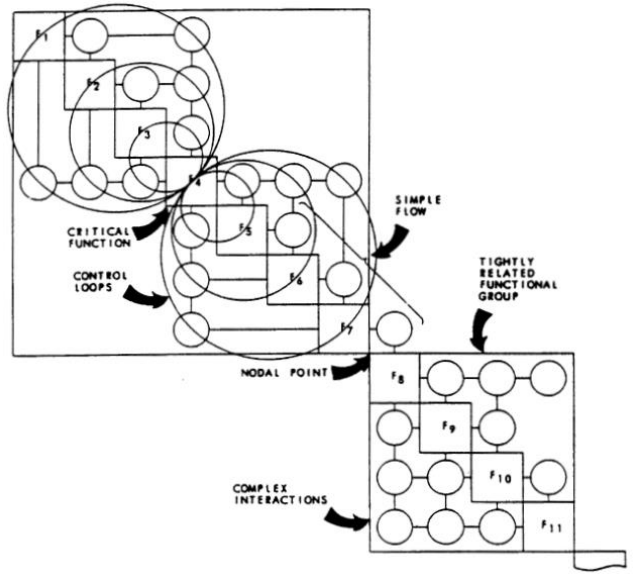
Systems Eng ???

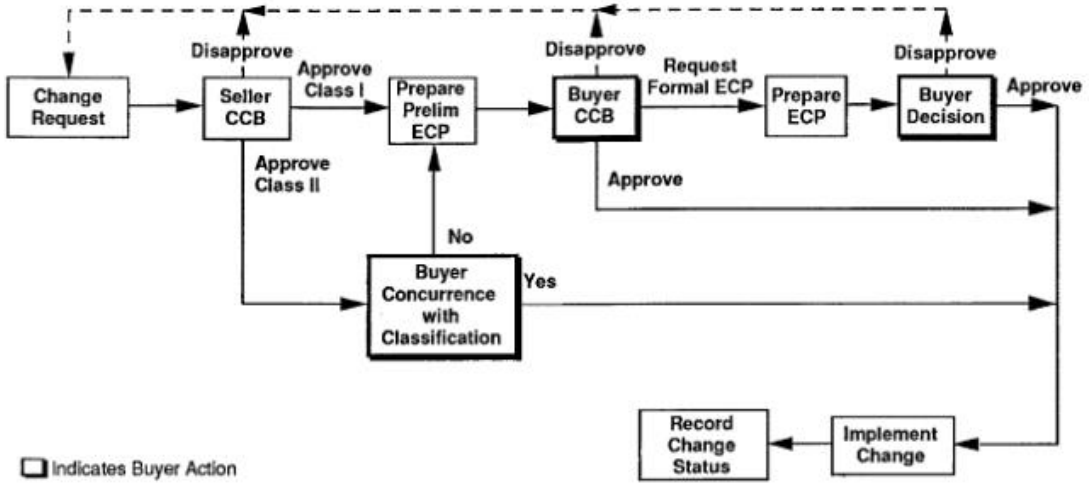
What is Systems Engineering's language?

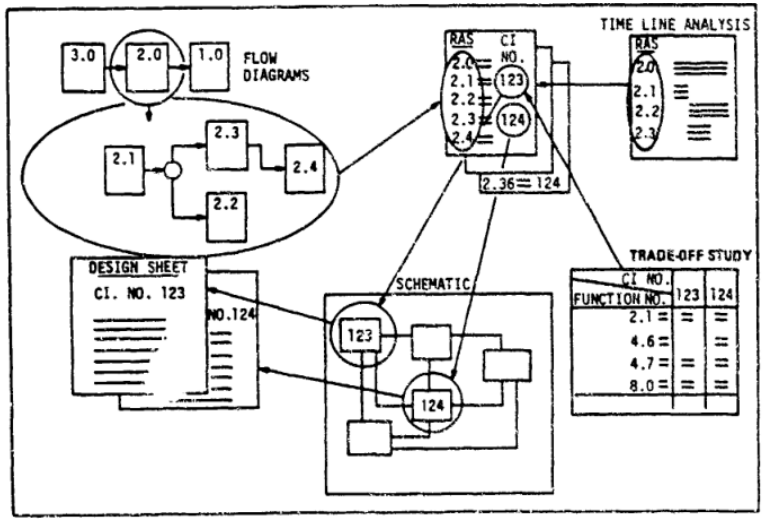




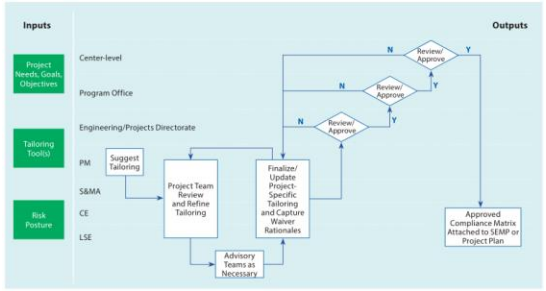


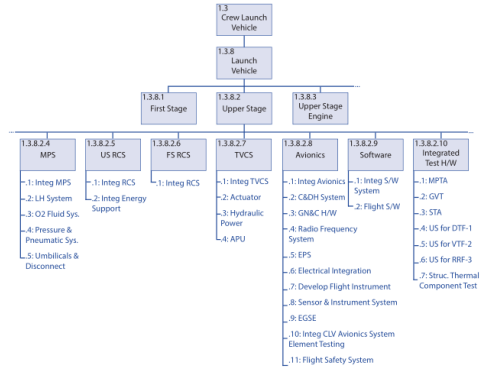


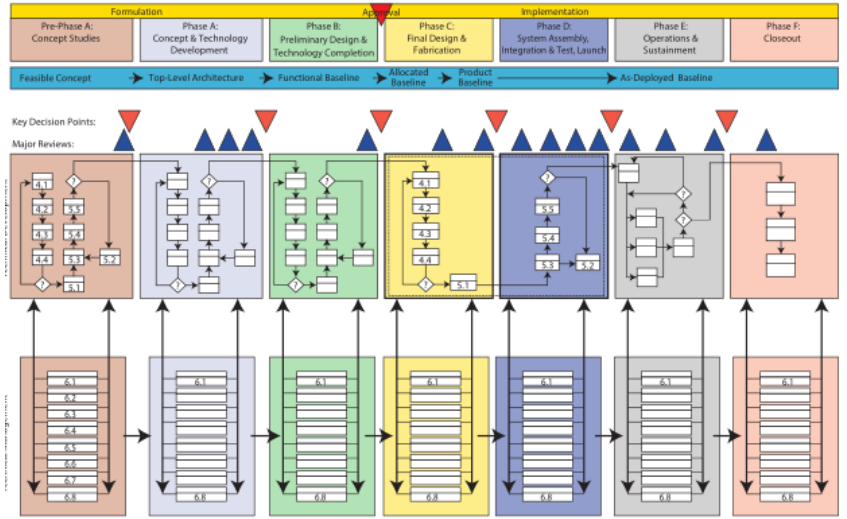


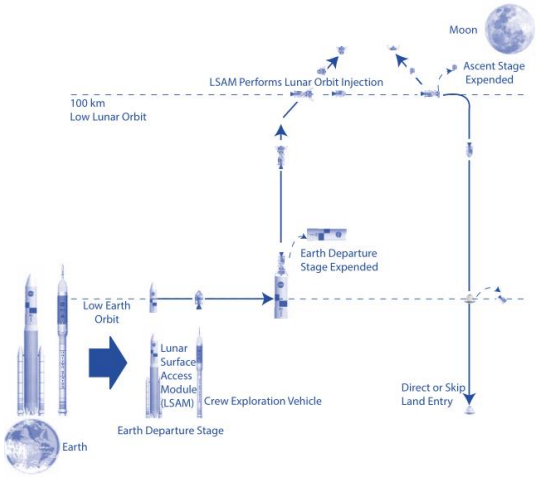


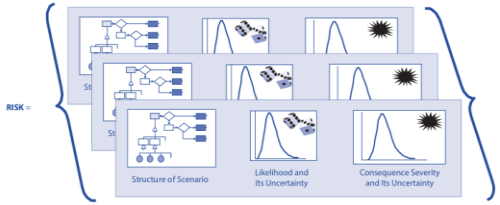
Over time… They became colorful











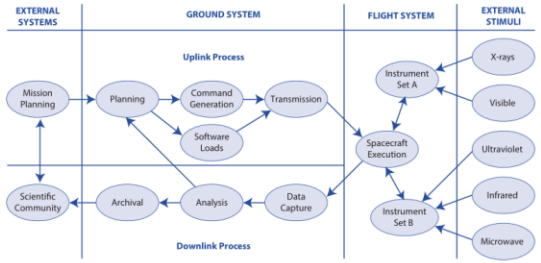


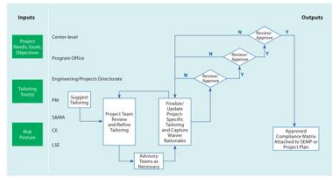
TABLE D-1 Requirements Verification Matrix

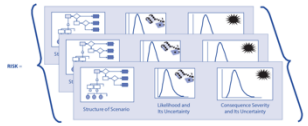
Requirement No.	Document	Paragraph	Shall Statement	Verification Success Criteria	Verification Method	Facility or Lab	Phase*	Acceptance Required?	Preflight Acceptance?	Performing Organization	Results
Unique identifier or each requirement	*Document number the requirement is contained within*	*Paragraph number of the requirement*	*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 initial acceptance testing of each unit.*	*Indicate whether this requirement is also verified during any pre-flight or recurring 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 locks to forward link at the min and max data rate tolerances 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-i	xxx	Other paragraphs	Other "shalls" in PTRS	Other criteria	xxx	xxx	xxx	Yes/No	Yes/No	xxx	Memo xxx
S-i or other unique designator	xxxxx (other specs, ICDs, etc.)	Other paragraphs	Other "shalls" in specs, ICDs, etc.	Other criteria	xxx	xxx	xxx	Yes/No	Yes/No	xxx	Report xxx

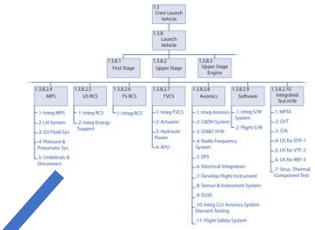
Red = Below TRL 3
Yellow = TRL 3, 4 & 5
Green = TRL 6 and above
White = Unknown
X = Exists

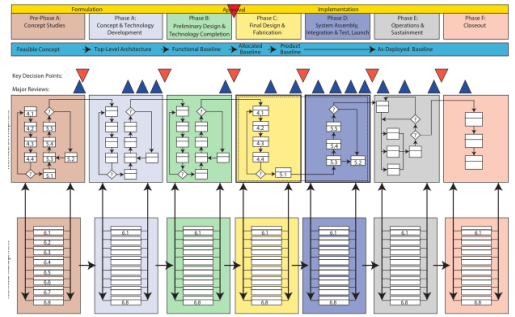
	Concept	Demonstration Units: Breadboard	Demonstration Units: Brassboard	Demonstration Units: Developmental Model	Demonstration Units: Prototype	Demonstration Units: Flight Qualified	Environment: Laboratory Environment	Environment: Relevant Environment	Environment: Space Environment	Environment: Space Launch Operation	Unit Description: Form	Unit Description: Fit	Unit Description: Function	Unit Description: Appropriatae Scale	Overall TRL
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															
1.1.6 Thermal Systems							X				X	X			
1.1.7 Fluid Systems		X													
1.1.8 Optical Systems															
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															

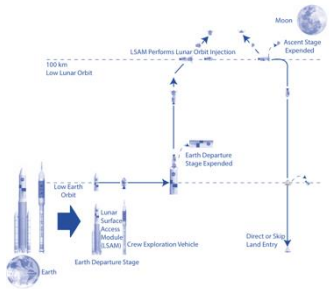
The Systems Engineering Team is the integrator

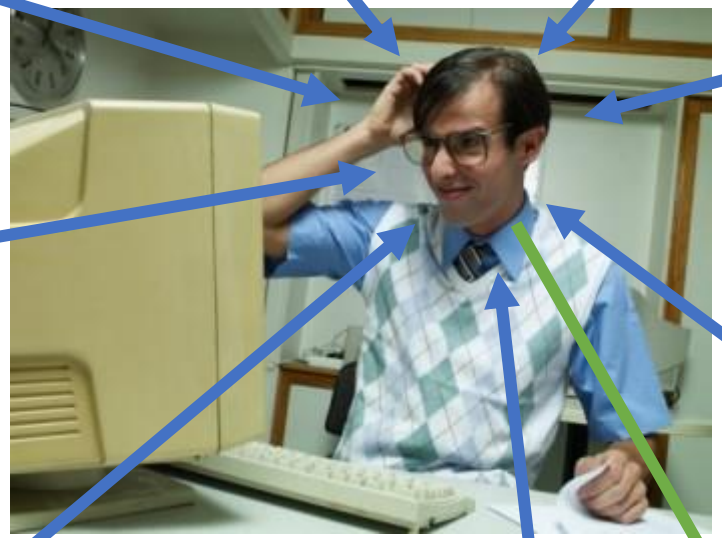


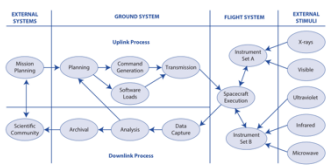


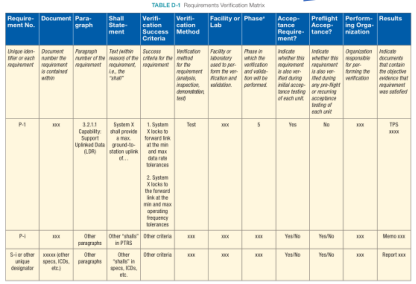


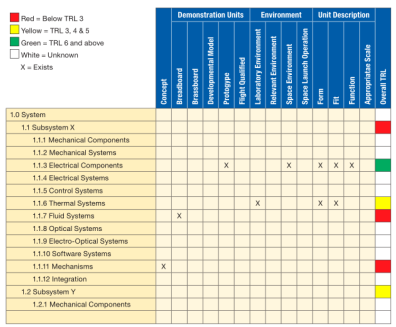






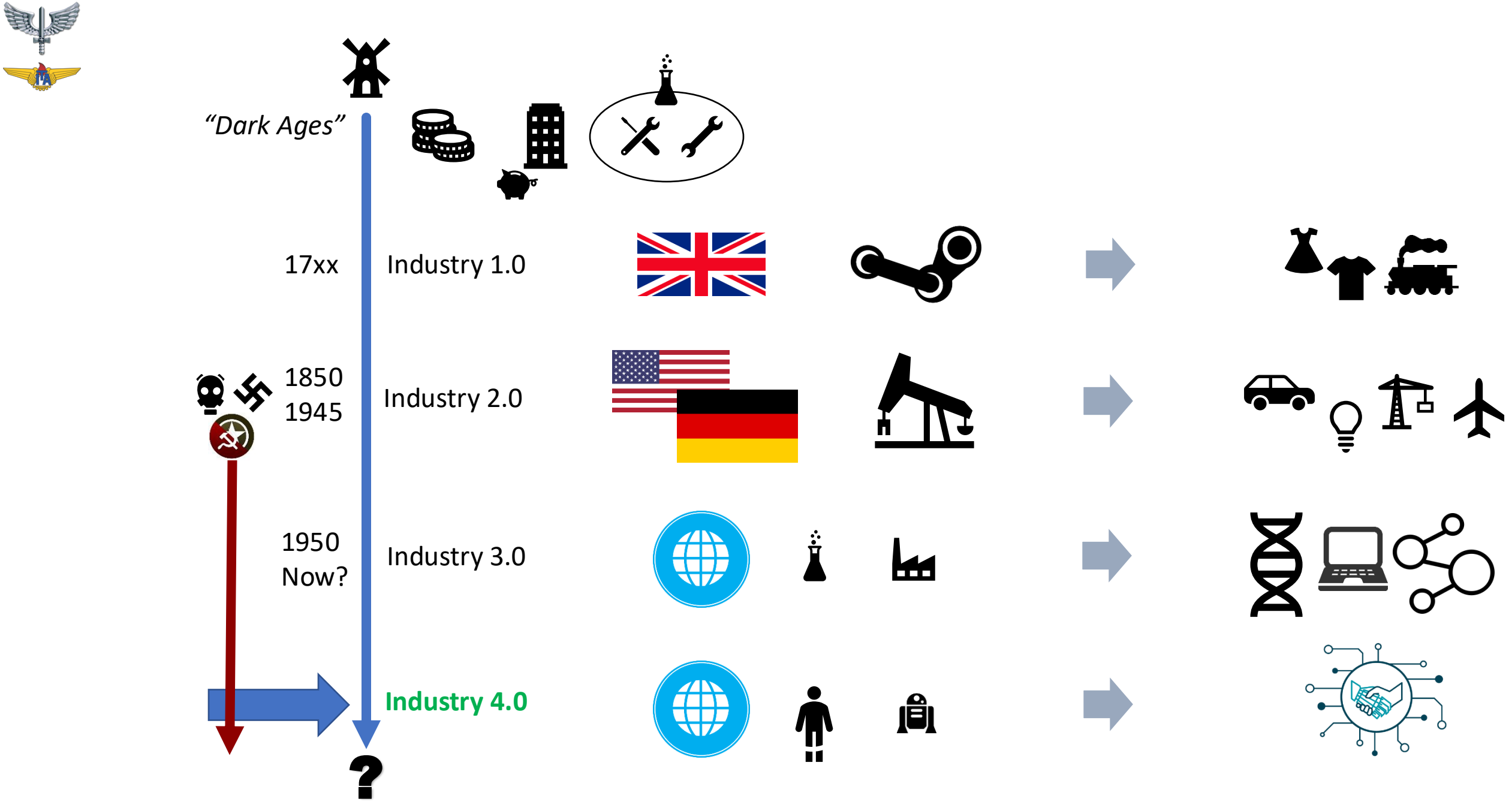




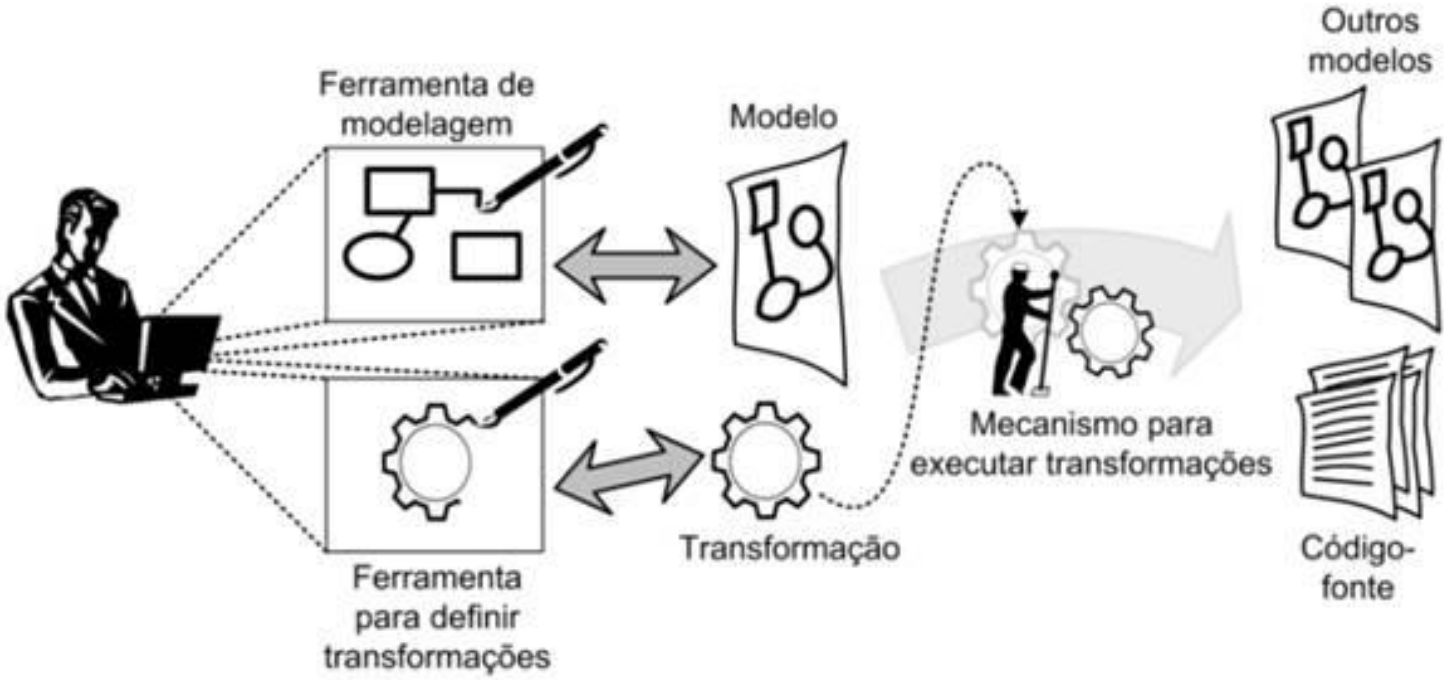


Integrator

Digitalization of the Activities


"Dark Ages"
17xx
Industry 1.0
1850
1945
Industry 2.0
1950
Now?
Industry 3.0
Industry 4.0
?

Model Driven Development ...



- **MDD** – Model Driven Development
- **MDSD** – Model Driven Software Development
- **MDA** – Model Driven Architect
- **MDSE** – Model Driven Software Engineering
- **MDRE** – Model Driven Reverse Engineering
- **MM** – Model Management
- **ADM** – Architecture Driven Modernization
- **DDD** – Domain Driven Design
- **MBD** – Model Based Development

http://www.devmedia.com.br/model-driven-architecture-com-enterprise-architect/33898

Model (driven x based x centric x oriented)

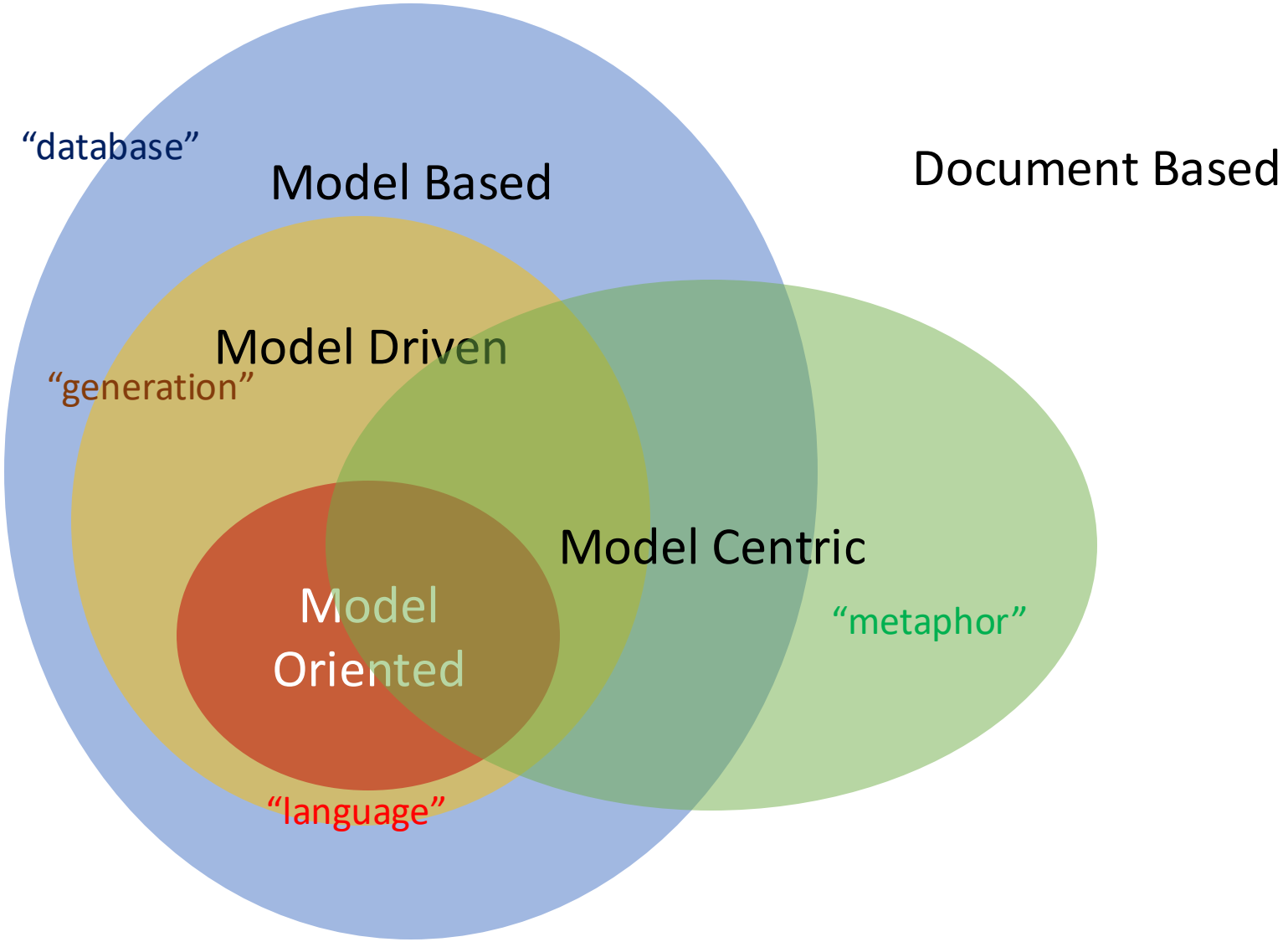
- **Driven:** models should be used to **generate executable systems**. Automatic code generation.
- **Based:** models are the **source of data** for lifecycle activities. Information storage.
- **Centric:** models are the **metaphor** of the tool. CAD tools.
- **Oriented:** models and the system (code) are **indistinguishable**. Scripts.

http://onlinelibrary.wiley.com/doi/10.1002/spe.1155/abstract#spe1155-note-0001

http://www.3dcadworld.com/why-you-need-to-understand-model-based-engineering/

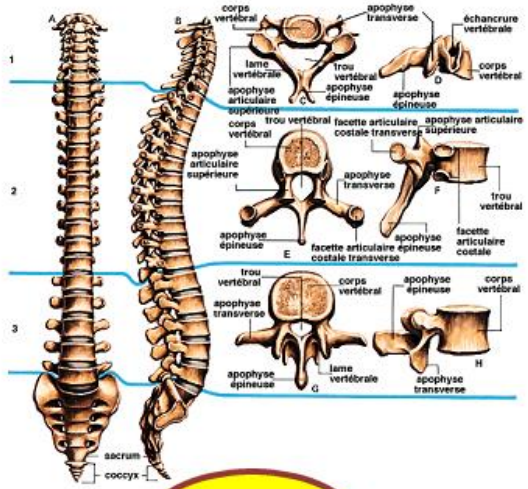
https://www.youtube.com/watch?v=VjGmNjg5cro

http://cruise.eecs.uottawa.ca/umple/

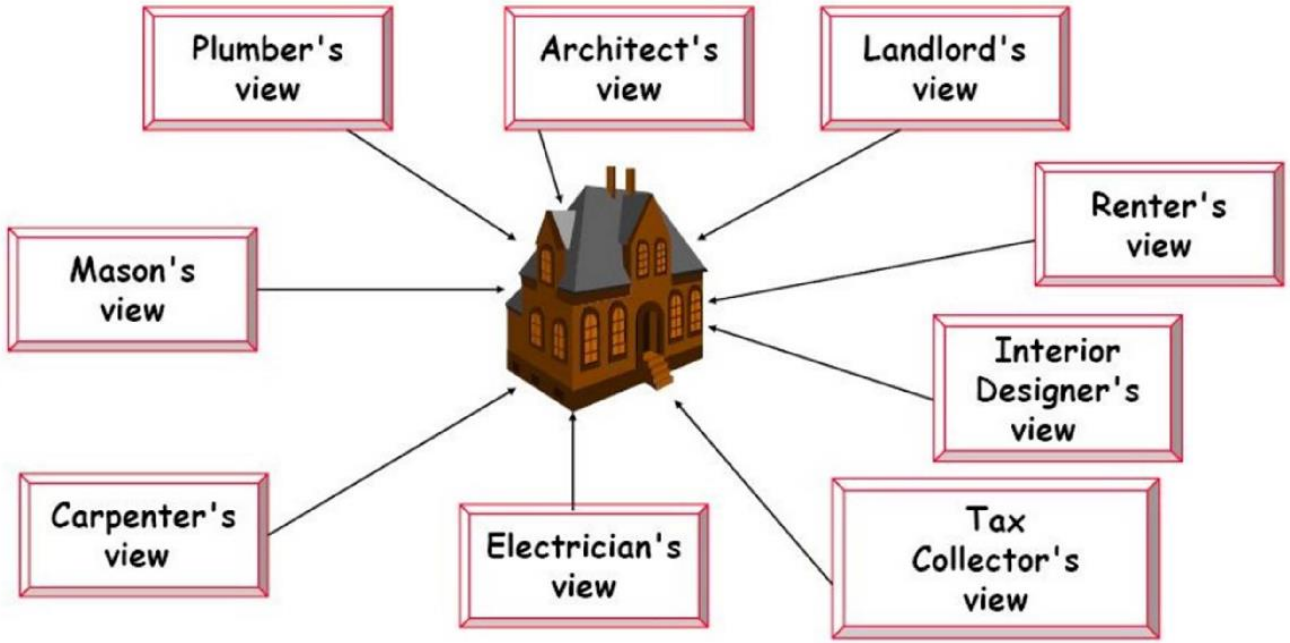

“database”
Model Based
Document Based
Model Driven
“generation”
Model Centric
Model
Oriented
“metaphor”
“language”

Models

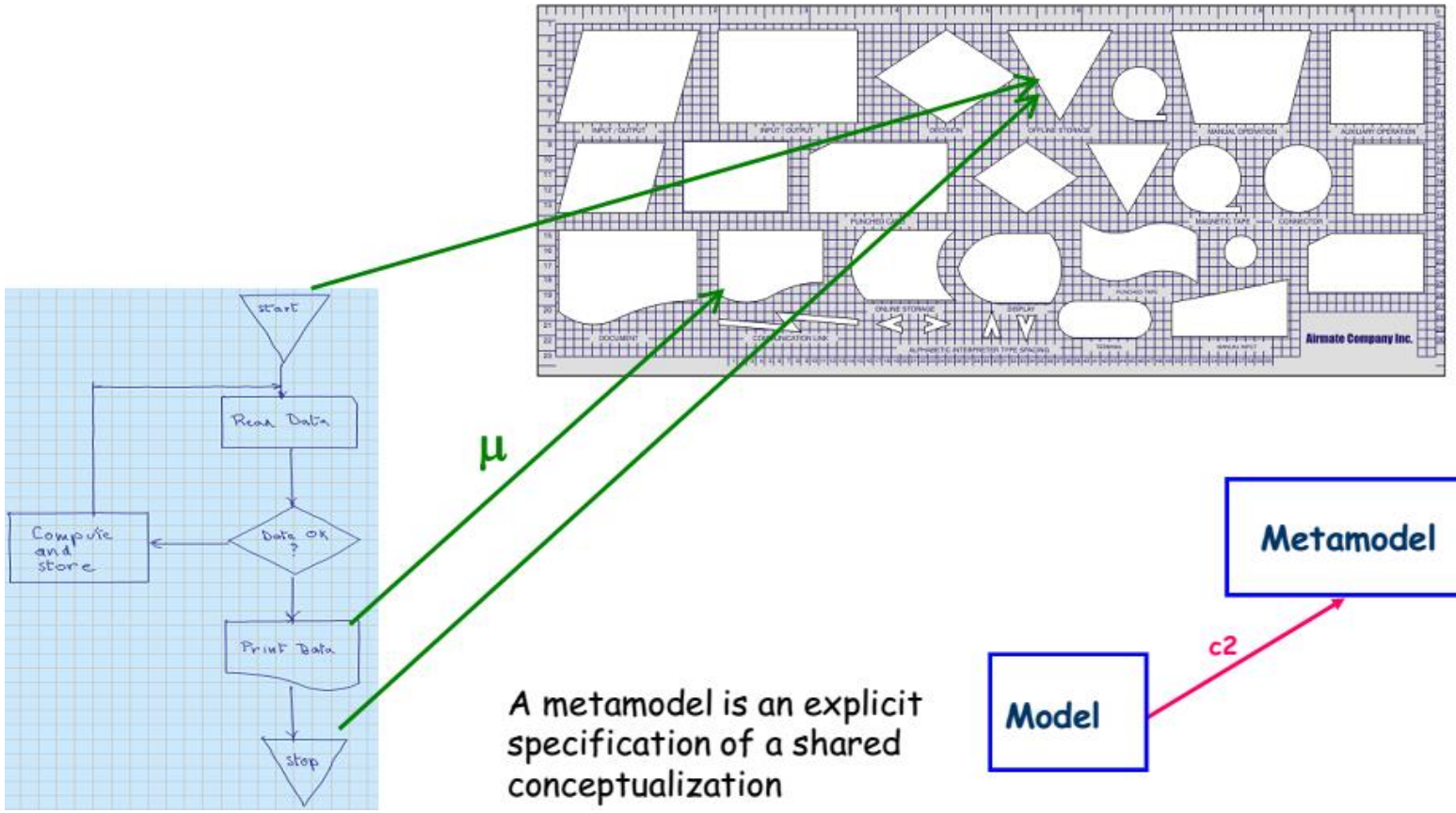
- A model is a **simplified** image of a system.
- "*modullus*", "*modus*" (measure)
  - **What**? (**mapping**)
  - **How**? (**reduction**)
  - **Whom, when, to what**? (**pragmatic**)
- Quite old idea



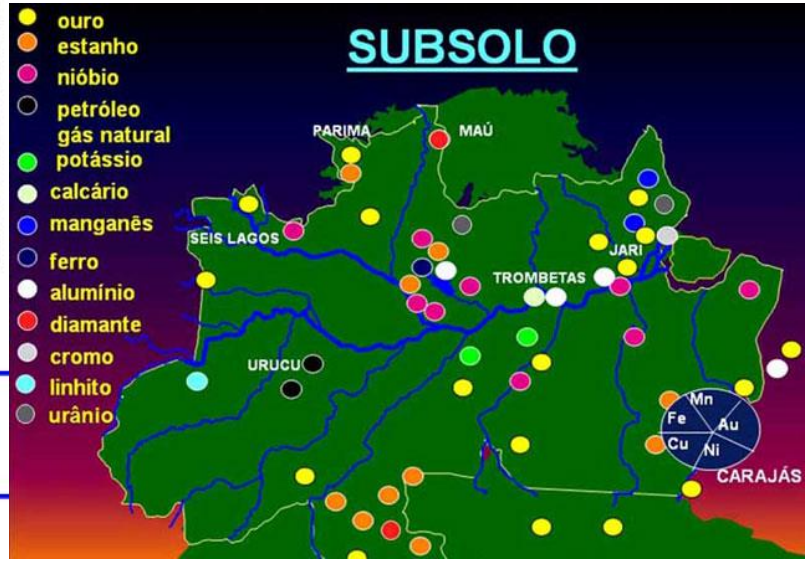
J. Bézivin, Model Engineering for Software Modernization, 2004.



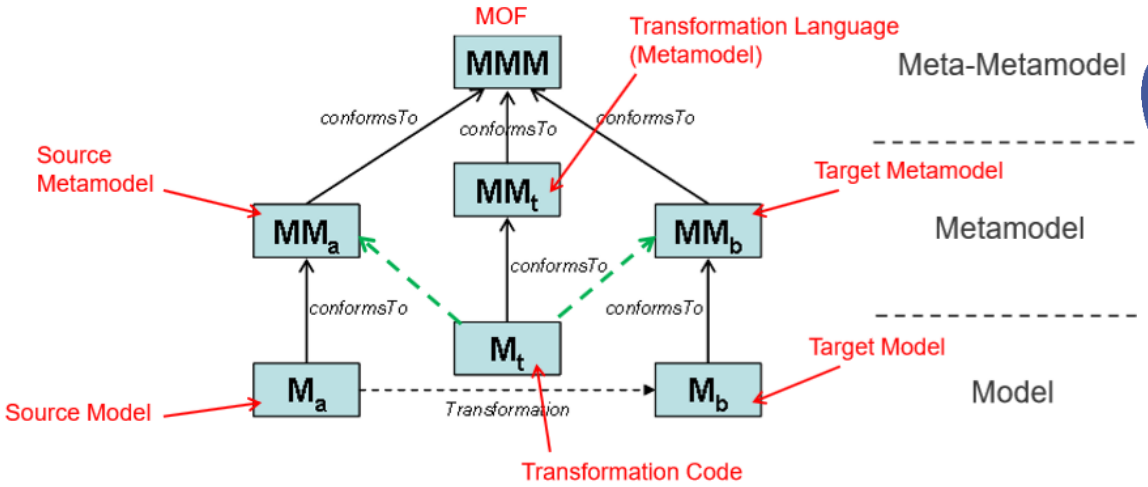
Meta-models

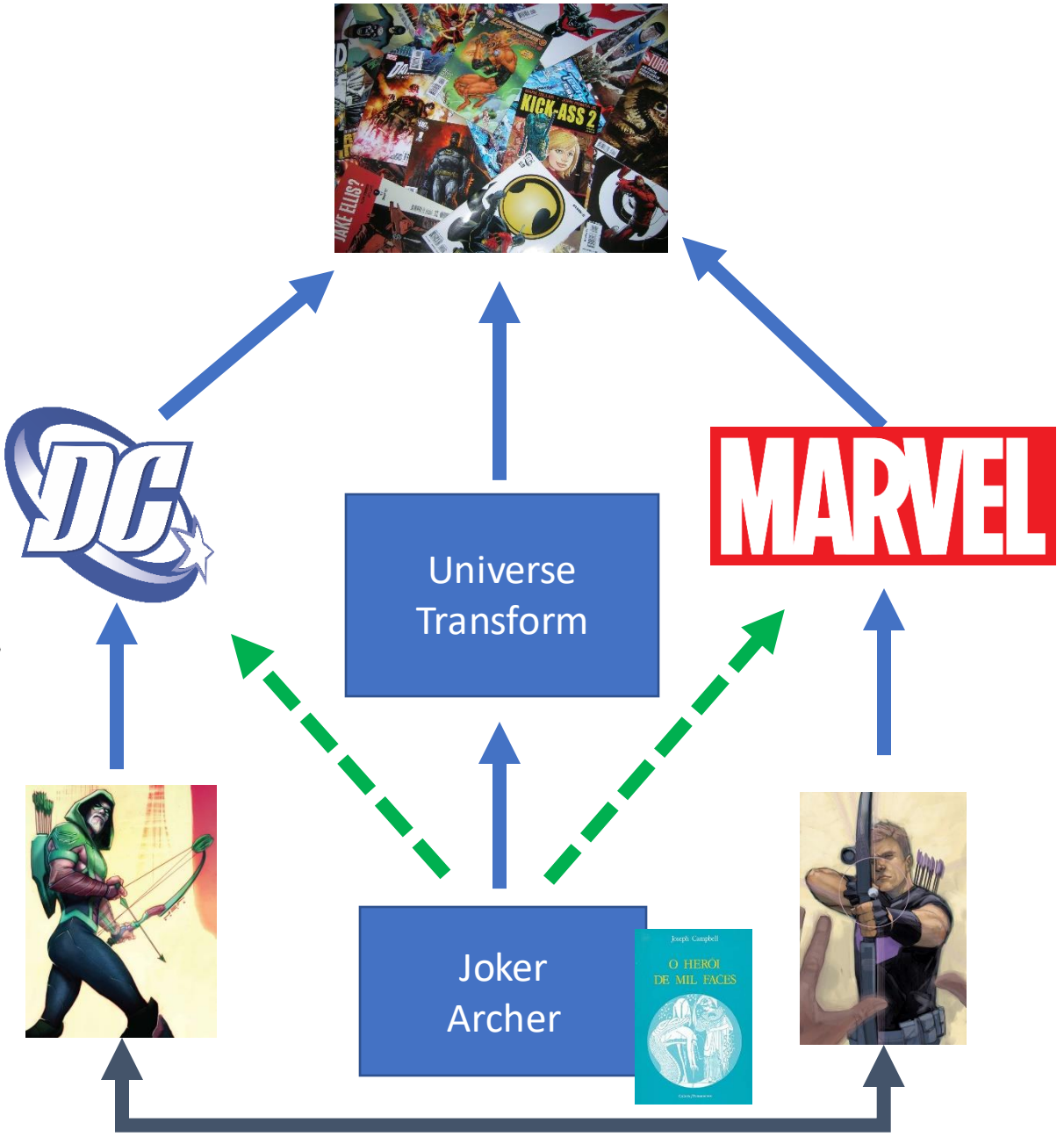


A metamodel is an explicit specification of a shared conceptualization

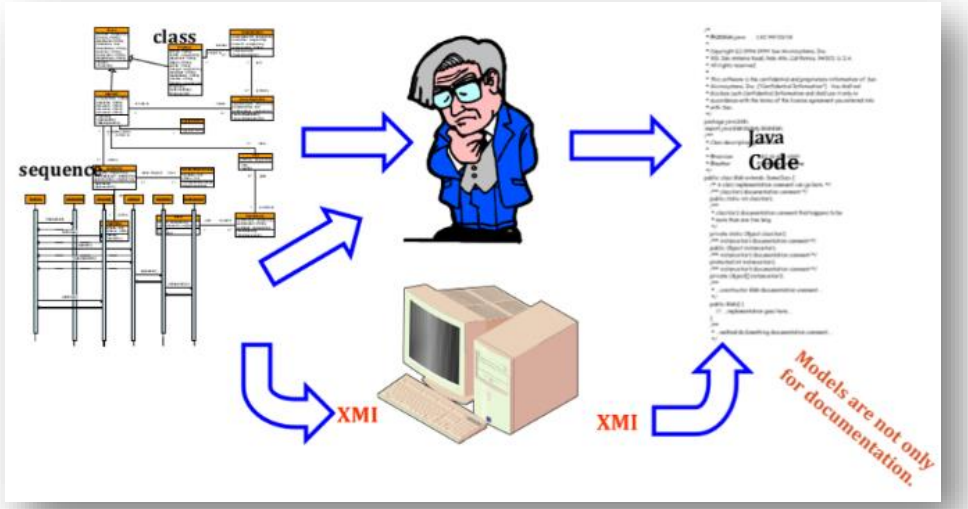


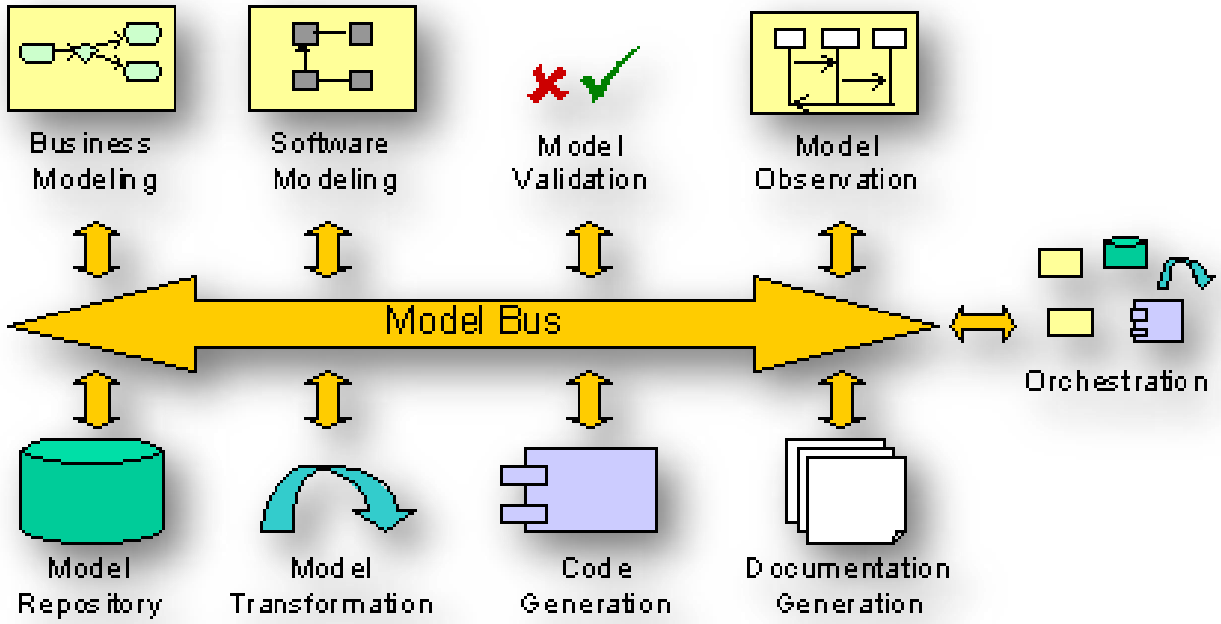
Model Transformation

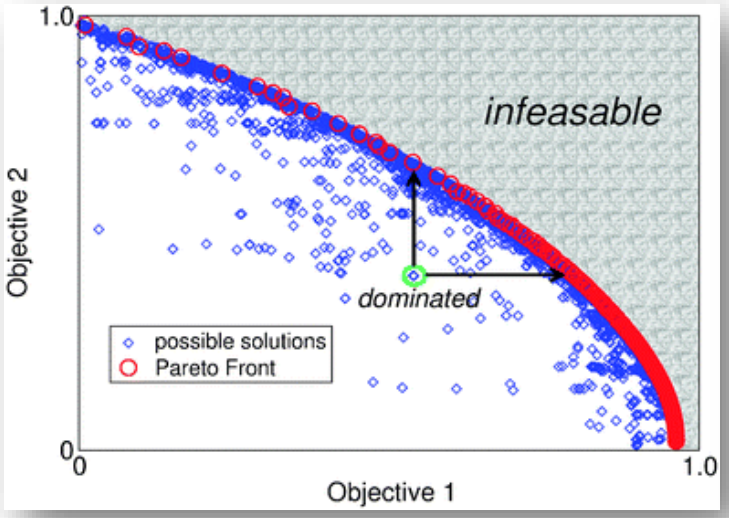


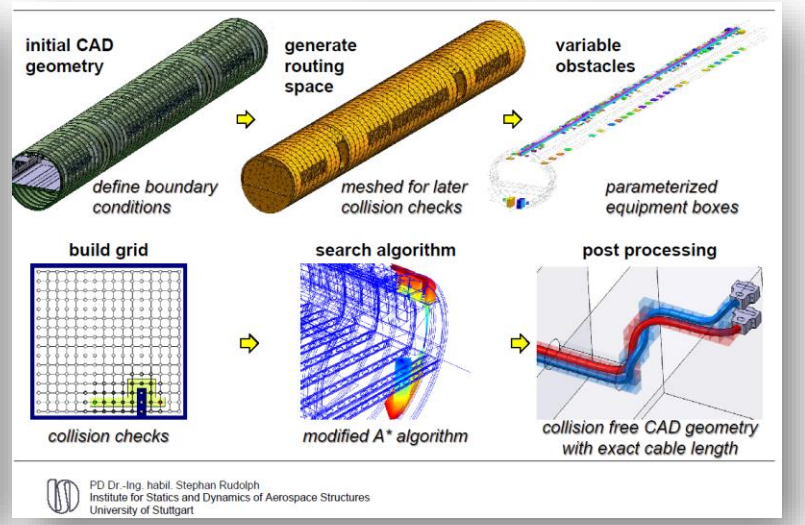


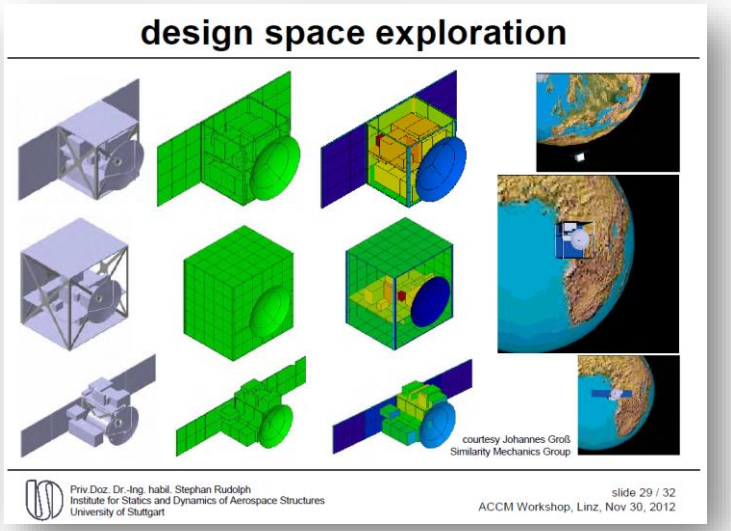
Benefits











http://www.eclipse.org/proposals/eclipse-mddi/

- This is not randon...

- It is a consequence of the computational technology being appropriated by the engineerings

Model Based Systems Engineering

MBSE

“Model-based systems engineering (MBSE) is a systems engineering methodology that focuses on creating and exploiting domain models as the primary means of information exchange between engineers, rather than on document-based information exchange.”

“the formalized application of modelling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases. MBSE is part of a long-term trend **toward model-centric approaches** adopted by other engineering disciplines, including mechanical, electrical and software. In particular, MBSE is expected to **replace the document-centric** approach that has been practiced by systems engineers in the past and to influence the future practice of systems engineering by being fully integrated into the definition of systems engineering processes.”

http://www.omgwiki.org/MBSE/doku.php

So… MBSE requires a methodology… Why?

- **The tool will not provide an universal language**, it filter the "universe" of symbols into a set of options.

- **Methodology** - Defined as a collection of related processes, methods, and tools.
  - **Process** - A logical sequence of tasks performed to achieve a particular objective. A process defines the "**WHAT**" is to be done, without specifying the "HOW" each task is to be performed.
  - **Method** - Consists of techniques for performing a task, the "**HOW**" of each task. The terms "method," "technique," practice," and "procedure" can be used interchangeably in this context.
  - **Tool** - An instrument that, when applied to a particular method, can enhance the efficiency of a task. Thus, methods help bridge the gap between process and tools. The purpose of the tool should be to facilitate the accomplishment of the "HOWs".

INCOSE METHODOLOGIES

List of Methodologies and Methods

Methodologies Surveyed in INCOSE 2008 Report

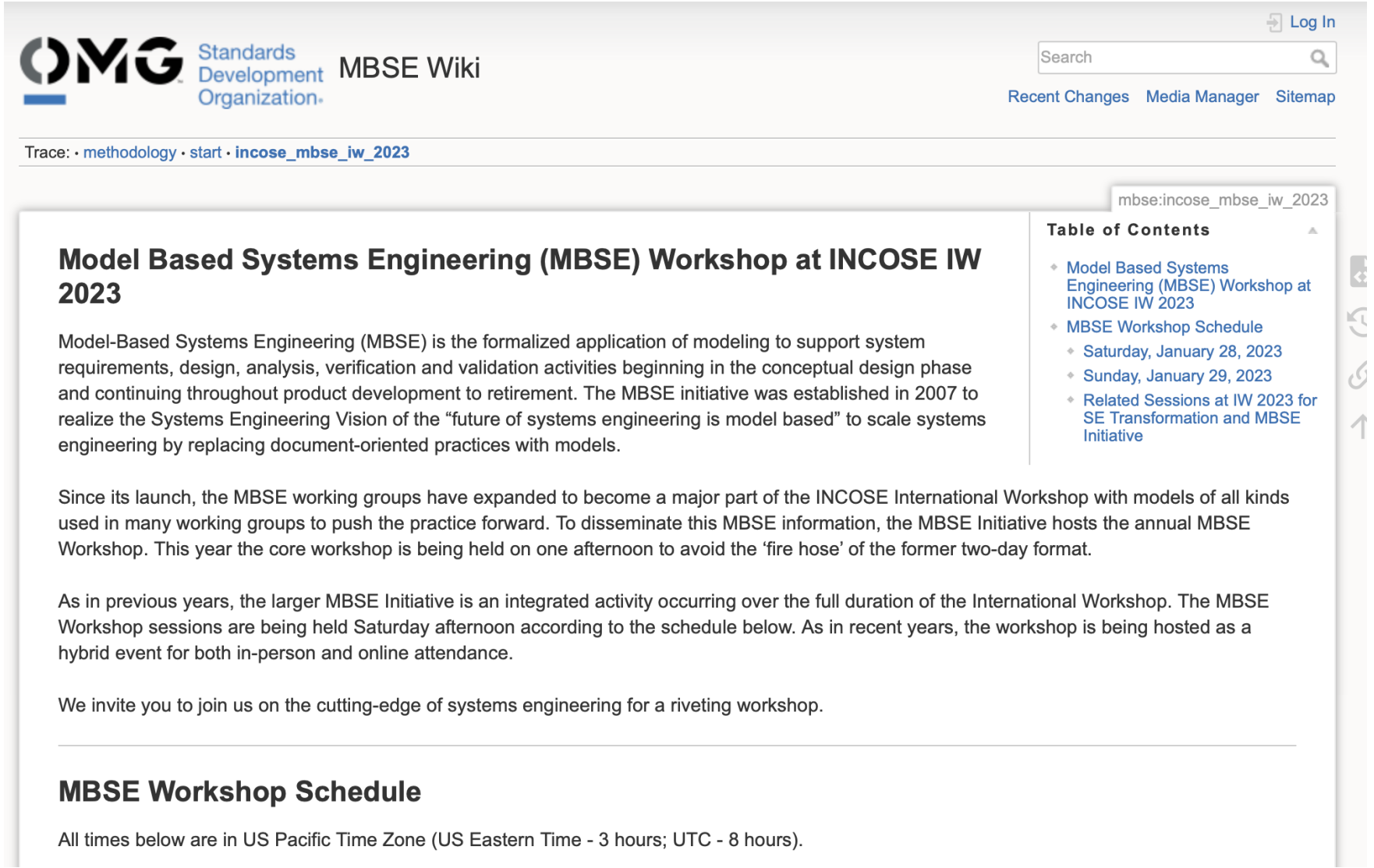
Name	Primary Point of Contact
INCOSE Object-Oriented Systems Engineering Method (OOSEM)	safriedenthal@gmail.com
IBM Rational Telelogic Harmony-SE	peter.hoffmann@telelogic.com
IBM Rational Unified Process for Systems Engineering (RUP-SE)	mcantor@us.ibm.com
Vitech Model-Based Systems Engineering (MBSE) Methodology|Vitech	jlong@vitechcorp.com
JPL State Analysis (SA) Methodology|JPL State Analysis (SA)	Robert.D.Rasmussen@jpl.nasa.gov
Dori Object-Process Methodology (OPM)	dori@ie.technion.ac.il

Additional Methodologies Identified as Gaps Since 2008 INCOSE Survey

Name	Primary Point of Contact
Weilkiens Systems Modeling Process (SYSMOD)	Tim.Weilkiens@oose.de
Fernandez Process Pipelines in OO Architectures (PPOOA)	joselfernandez@telefonica.net
An Ontology for State Analysis: Formalizing the Mapping to SysML	nicolas.f.rouquette@jpl.nasa.gov
ISO-15288, OOSEM and Model-Based Submarine Design	Paul.Pearce@deepbluetech.com.au
Alstom ASAP methodology	marco.ferrogalini@transport.alstom.com
Pattern-Based Systems Engineering (PBSE)	schindel@ictt.com
Arcadia, a model-based engineering method	Polarsys/Capella

https://www.omgwiki.org/MBSE/doku.php?id=mbse:methodology

Workgroup for MBSE



OMG Standards Development Organization · MBSE Wiki

Log In

Search

Recent Changes · Media Manager · Sitemap

Trace: · methodology · start · incose_mbse_iw_2023

mbse:incose_mbse_iw_2023

Model Based Systems Engineering (MBSE) Workshop at INCOSE IW 2023

Model-Based Systems Engineering (MBSE) is the formalized application of modeling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout product development to retirement. The MBSE initiative was established in 2007 to realize the Systems Engineering Vision of the "future of systems engineering is model based" to scale systems engineering by replacing document-oriented practices with models.

Table of Contents

Since its launch, the MBSE working groups have expanded to become a major part of the INCOSE International Workshop with models of all kinds used in many working groups to push the practice forward. To disseminate this MBSE information, the MBSE Initiative hosts the annual MBSE Workshop. This year the core workshop is being held on one afternoon to avoid the 'fire hose' of the former two-day format.

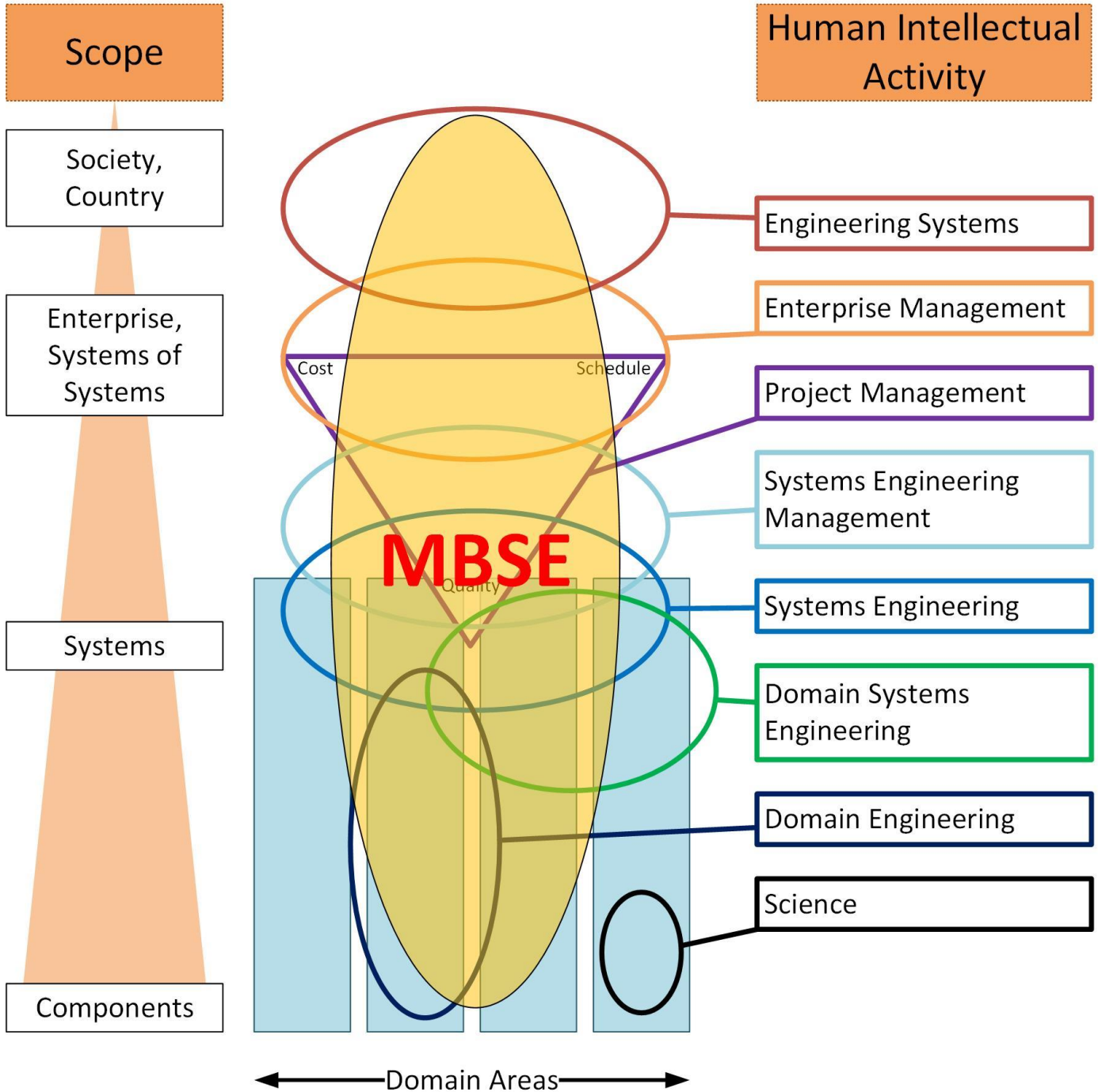
As in previous years, the larger MBSE Initiative is an integrated activity occurring over the full duration of the International Workshop. The MBSE Workshop sessions are being held Saturday afternoon according to the schedule below. As in recent years, the workshop is being hosted as a hybrid event for both in-person and online attendance.

We invite you to join us on the cutting-edge of systems engineering for a riveting workshop.

MBSE Workshop Schedule

All times below are in US Pacific Time Zone (US Eastern Time - 3 hours; UTC - 8 hours).

https://www.omgwiki.org/MBSE/doku.php?id=mbse:incose_mbse_iw_2023


Scope
Human Intellectual Activity
Society, Country
Enterprise, Systems of Systems
Systems
Components
Cost
Schedule
Quality
MBSE
Engineering Systems
Enterprise Management
Project Management
Systems Engineering Management
Systems Engineering
Domain Systems Engineering
Domain Engineering
Science
Domain Areas

Creating a system model

Simplified systems engineering technical processes.

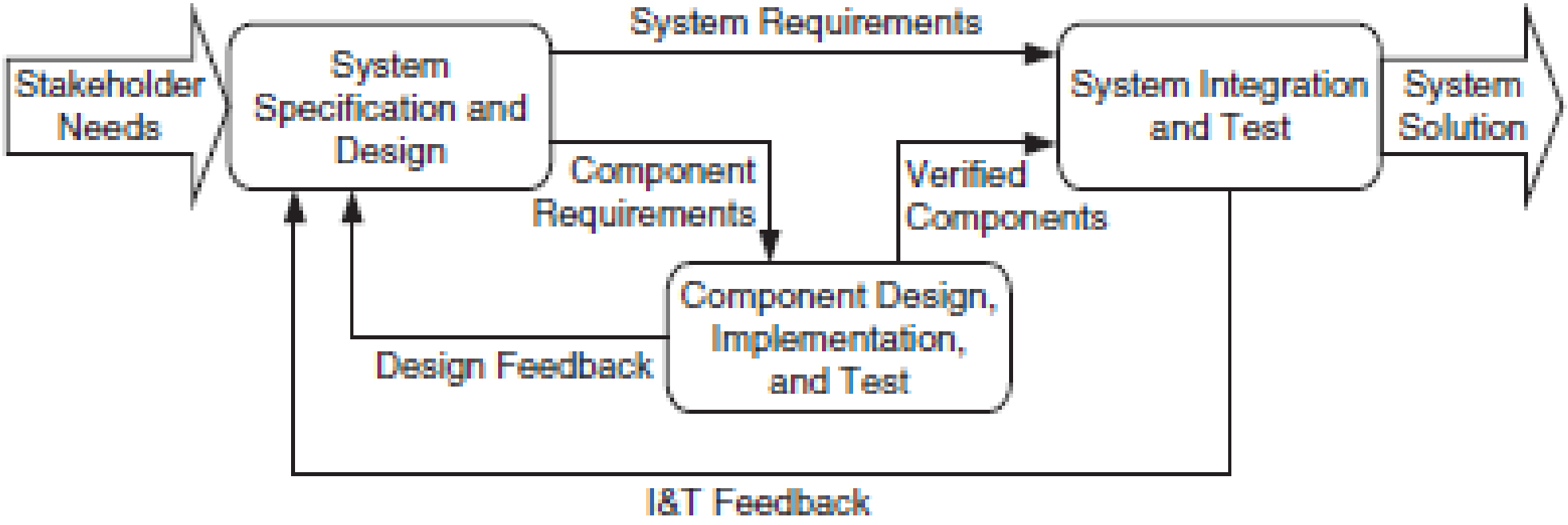


FIGURE 1.1

Simplified systems engineering technical processes.

Typical application of the systems engineering process

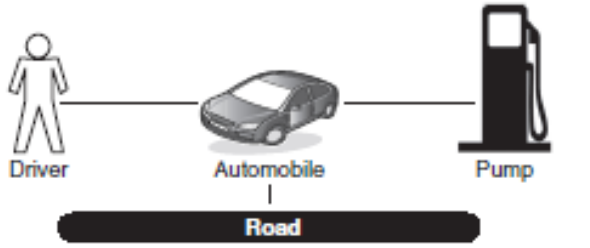


FIGURE 1.2
Defining the system boundary.

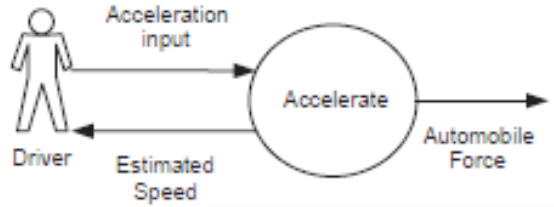


FIGURE 1.3
Specifying the functional requirements.

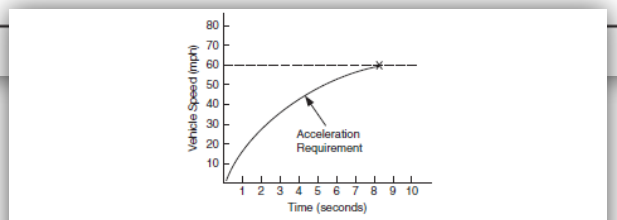


FIGURE 1.4
Automobile performance requirements.

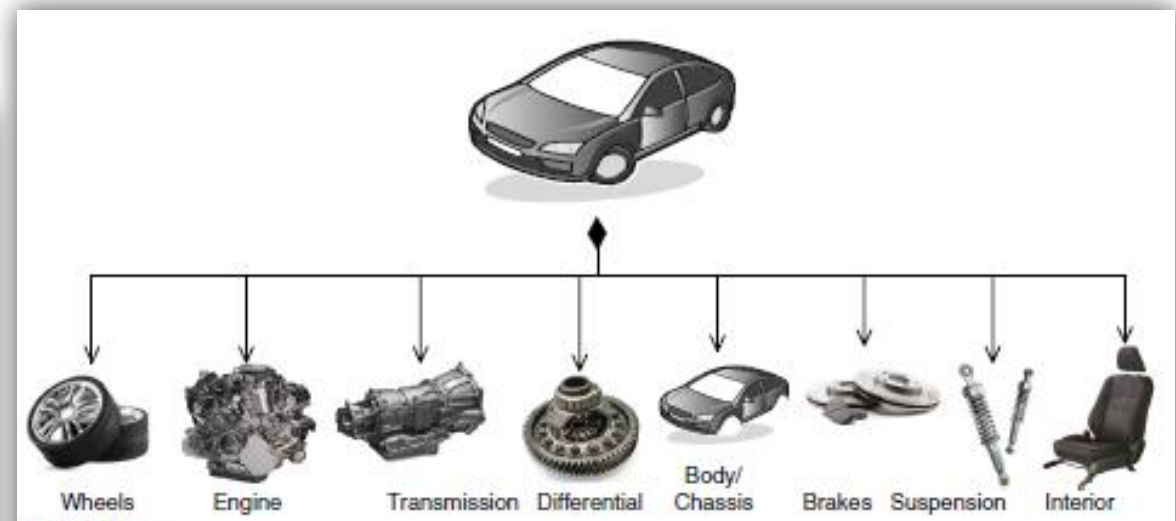


FIGURE 1.5
Automobile system decomposition into its components.

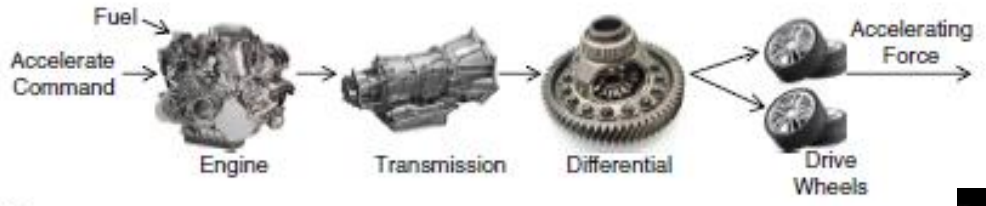
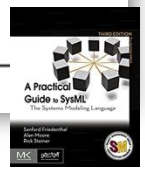
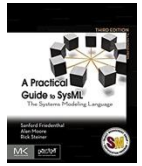


FIGURE 1.6
Interaction among components to achieve the system functional and performance requirements.



The System Model

- The **system model** is generally created using a modeling tool and stored in a model repository.
- The system model **includes system specifications, design, analysis, and verification** information.
- The model consists of model elements that represent **requirements, design, test cases, design rationale, and their interrelationships**.



The System Model

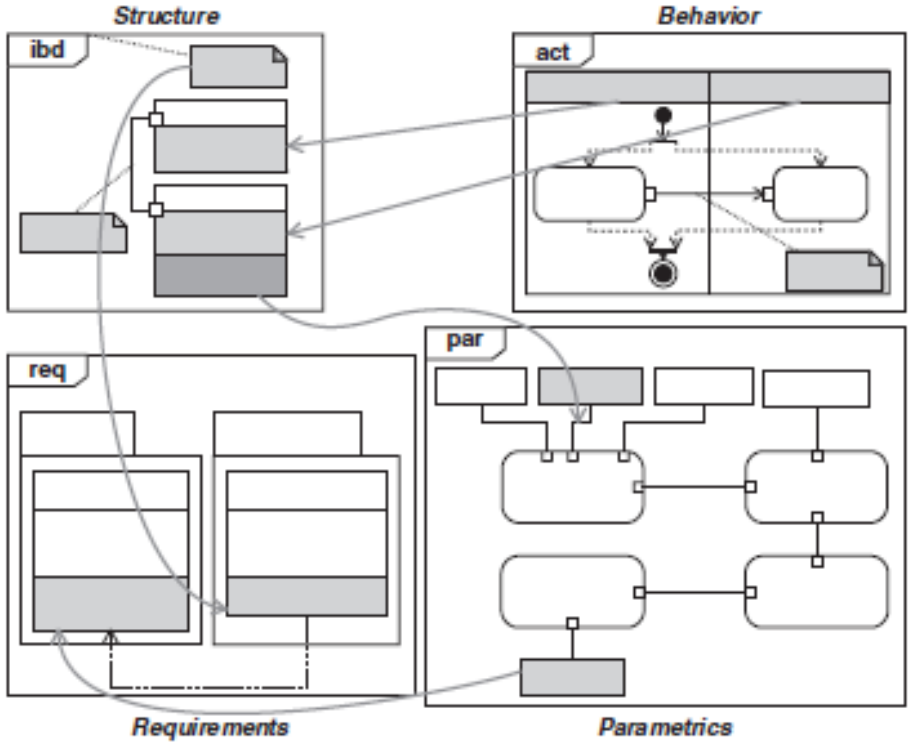


FIGURE 2.1

Representative system model example in SysML. (Specific model elements have been deliberately obscured and will be discussed in subsequent chapters.)

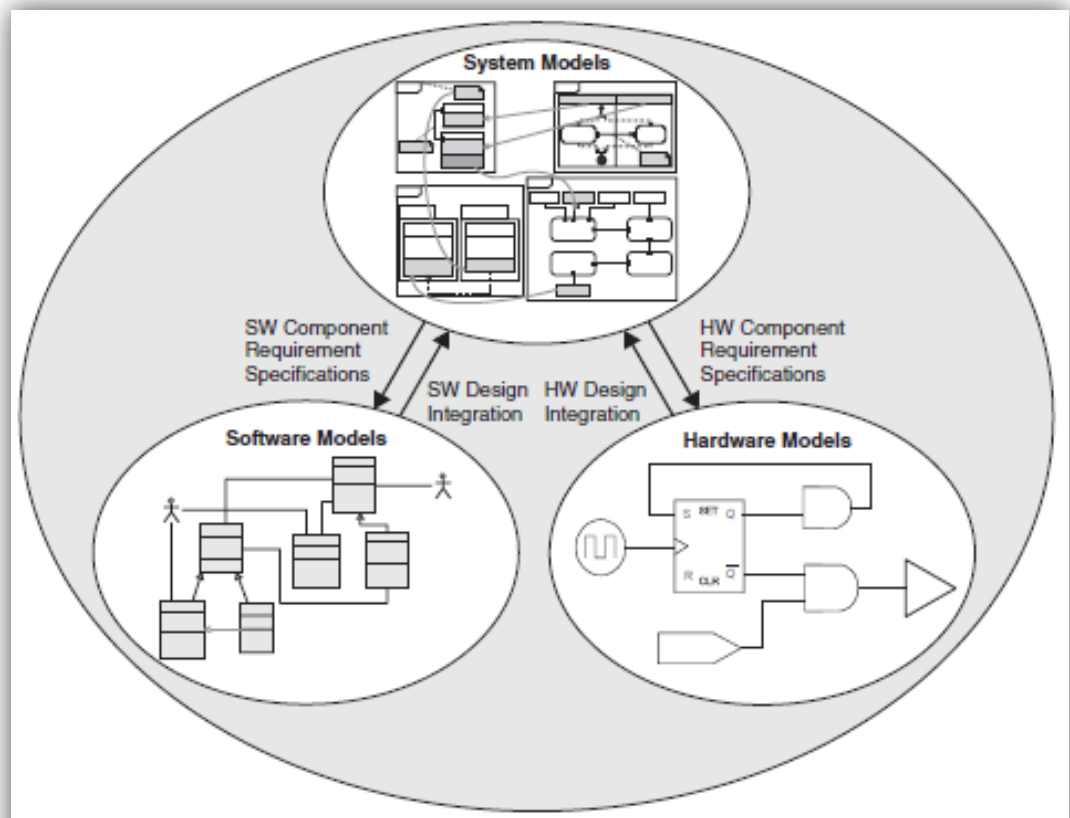
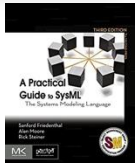


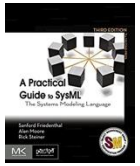
FIGURE 2.2

The system model is used to specify the components of the system.



Purpose (why – pragmatics)

- The **intended use** for modeling a system is associated with the **systems engineering activities** the model is intended to support across the system lifecycle, and may include the following uses:
  - Characterize and assess an existing system
  - Specify and design a new or modified system
  - Evaluate the system
  - Train users on how to operate or maintain a system
  - Support system maintenance and/or diagnostics

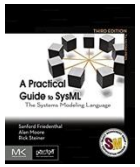


Validating the model

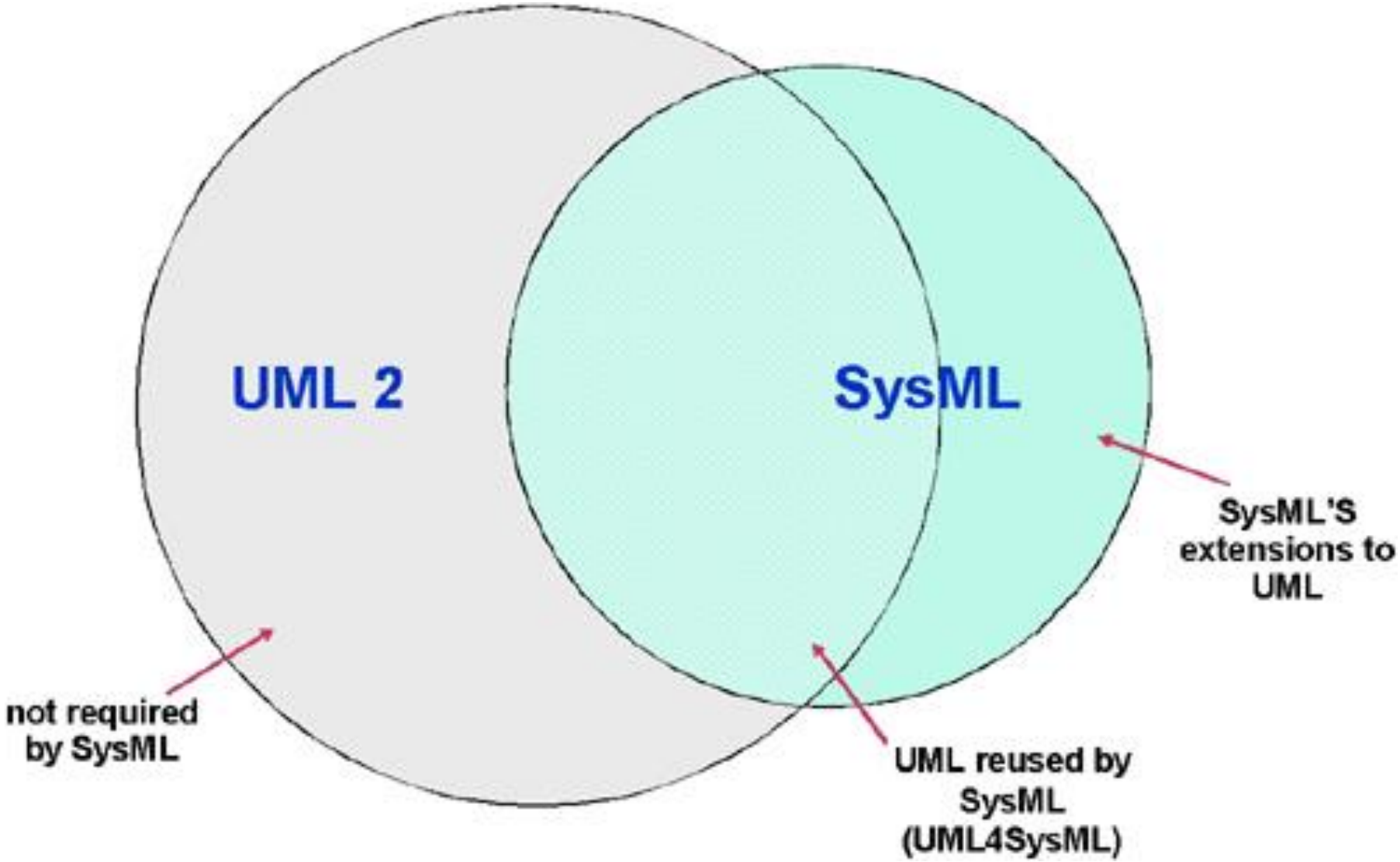
- **Model validation** is the process for determining the extent to which **the model accurately represents the domain of interest** (e.g., the system and its environment) to meet the model's intended use.
- The model's accuracy is dependent on the **quality of the source information** used to generate the model, the **validity of the assumptions** regarding the applicability of the source information, and the extent to which the source information and assumptions are **properly captured** in the model.
- As with analysis models, the system model validation can be performed by a combination of **model checks and domain expert review**.

SYSML

- SysML is a general-purpose graphical modeling language that ***(,proposes itself to be able to,)*** supports the analysis, specification, design, verification, and validation of complex systems. These systems may include hardware and equipment, software, data, personnel, procedures, facilities, and other elements of humanmade and natural systems.
- SysML can represent the following aspects of systems, components, and other entities:
  - Structural composition, interconnection, and classification;
  - Flow-based, message-based, and state-based behavior;
  - Constraints on the physical and performance properties;
  - Allocations between behavior, structure, and constraints; and
  - Requirements and their relationship to other requirements, design elements, and test cases.



Adaptation of UML to systemic domain



SysML diagram taxonomy.

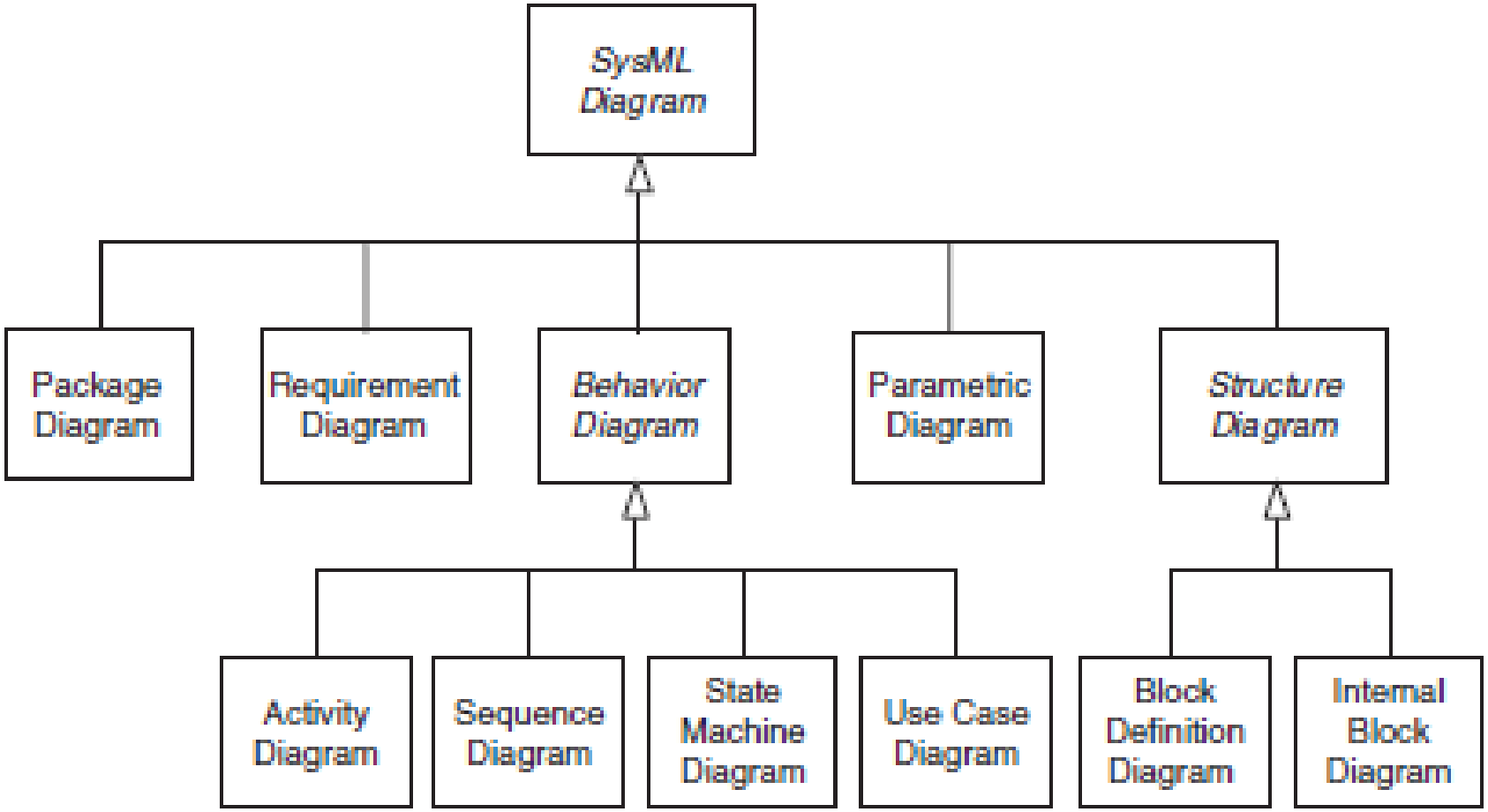


FIGURE 3.1

SysML diagram taxonomy.

Each diagram kind is summarized here, along with its relationship to UML diagrams:

- **Package diagram** – presents the organization of a model in terms of packages that contain model elements (same as UML package diagram).
- **Requirement diagram** – presents text-based requirements and their relationships to other requirements, design elements, and test cases to support requirements traceability (not in UML).
- **Activity diagram** – presents flow-based behavior indicating the order in which actions execute based on the availability of their inputs, outputs, and control, and how the actions transform the inputs to outputs (modification of UML activity diagram).
- **Sequence diagram** – presents behavior in terms of a sequence of messages exchanged between systems or parts of systems (same as UML sequence diagram).
- **State machine diagram** – presents behavior of an entity in terms of its transitions between states triggered by events (same as UML state machine diagram).
- **Use case diagram** – presents functionality in terms of how a system is used by external entities (i.e., actors) to accomplish a set of goals (same as UML use case diagram).
- **Block definition diagram** – presents structural elements, called blocks, and their composition and classification (modification of UML class diagram).
- **Internal block diagram** – presents interconnection and interfaces between the parts of a block (modification of UML composite structure diagram).
- **Parametric diagram** – presents constraints on property values, such as F = m * a, used to support engineering analysis (not in UML).

SysML Diagram taxonomy

- Every SysML diagram must have a **diagram frame** that encloses the diagram content. The diagram frame corresponds to a model element that provides the context for the diagram content.
- The **diagram header** is a rectangle with its lower right corner cut off. It includes the following information:
  - Diagram kind - an abbreviation indicating the kind of diagram.
  - Model element kind - the kind of model element to which the diagram frame corresponds.
  - Model element name - the name of the model element to which the diagram frame corresponds.
  - Diagram name - the name of the diagram, which is often used to indicate the diagram purpose.
  - Diagram usage - a keyword indicating a specialized use of a diagram.

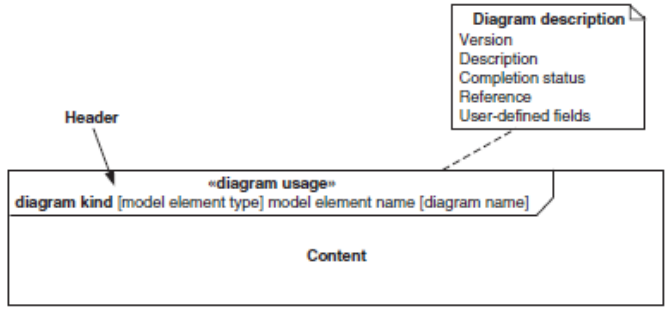


FIGURE 5.5
A diagram frame.

SysML structural diagrams

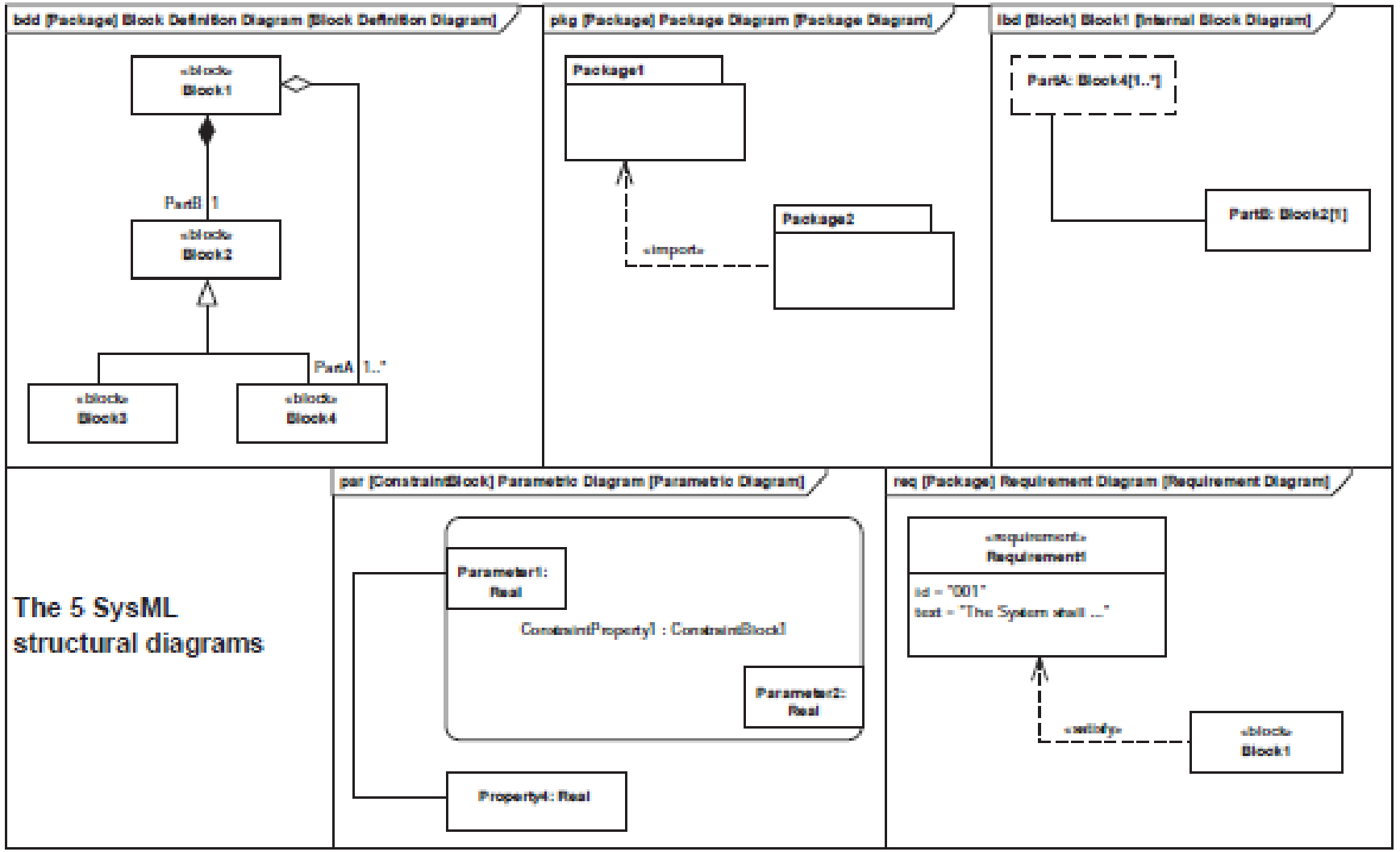


Figure 4.7 SysML structural diagrams

SysML behavioural diagrams

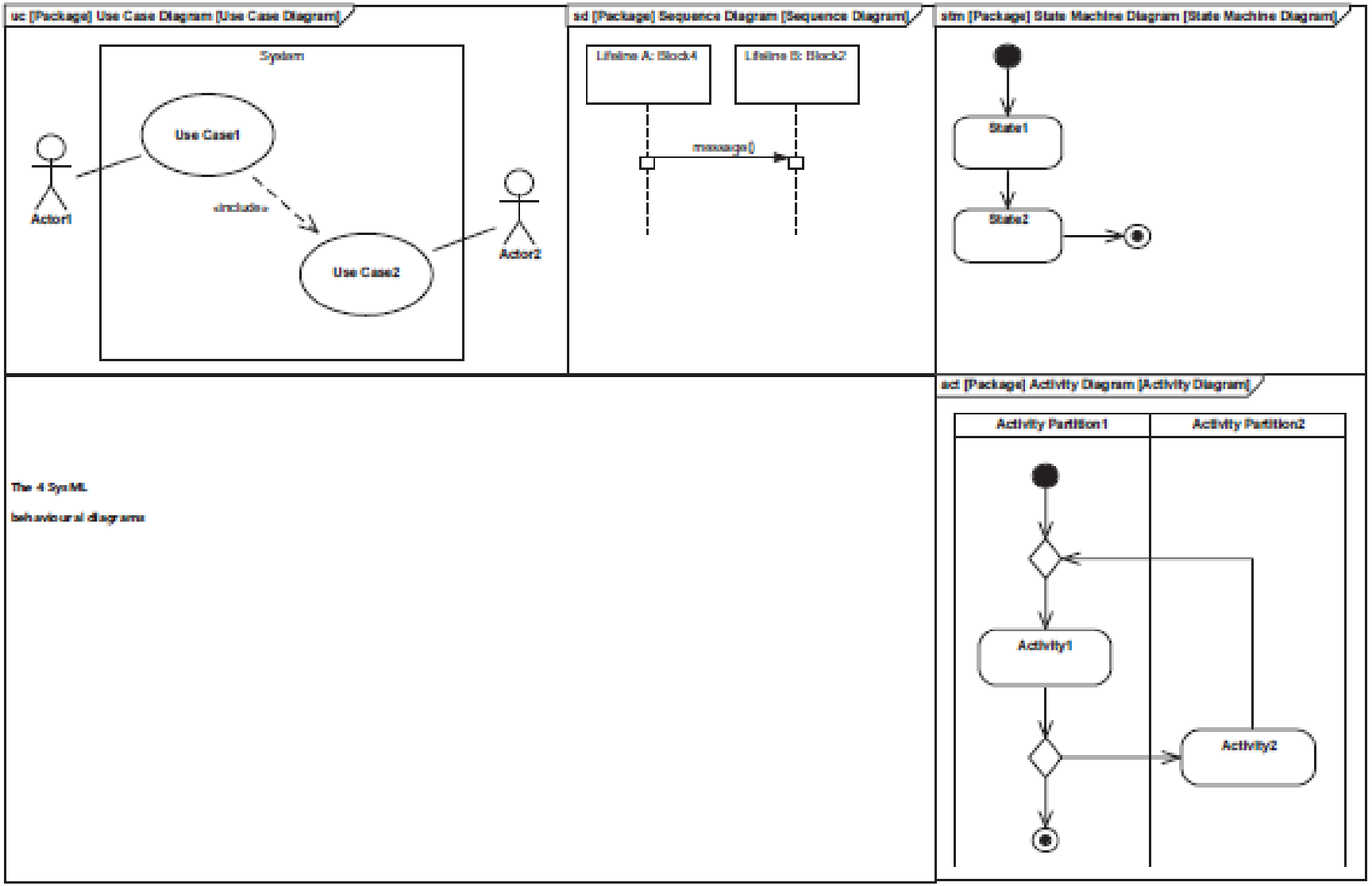


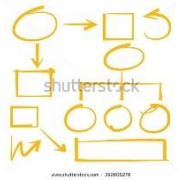
Figure 4.8 SysML behavioural diagrams

OPM

The start:
How do we explain ideas to each other?

Grab a pen and piece of paper, or a chalk and blackboard

Scribble shapes with names next to them

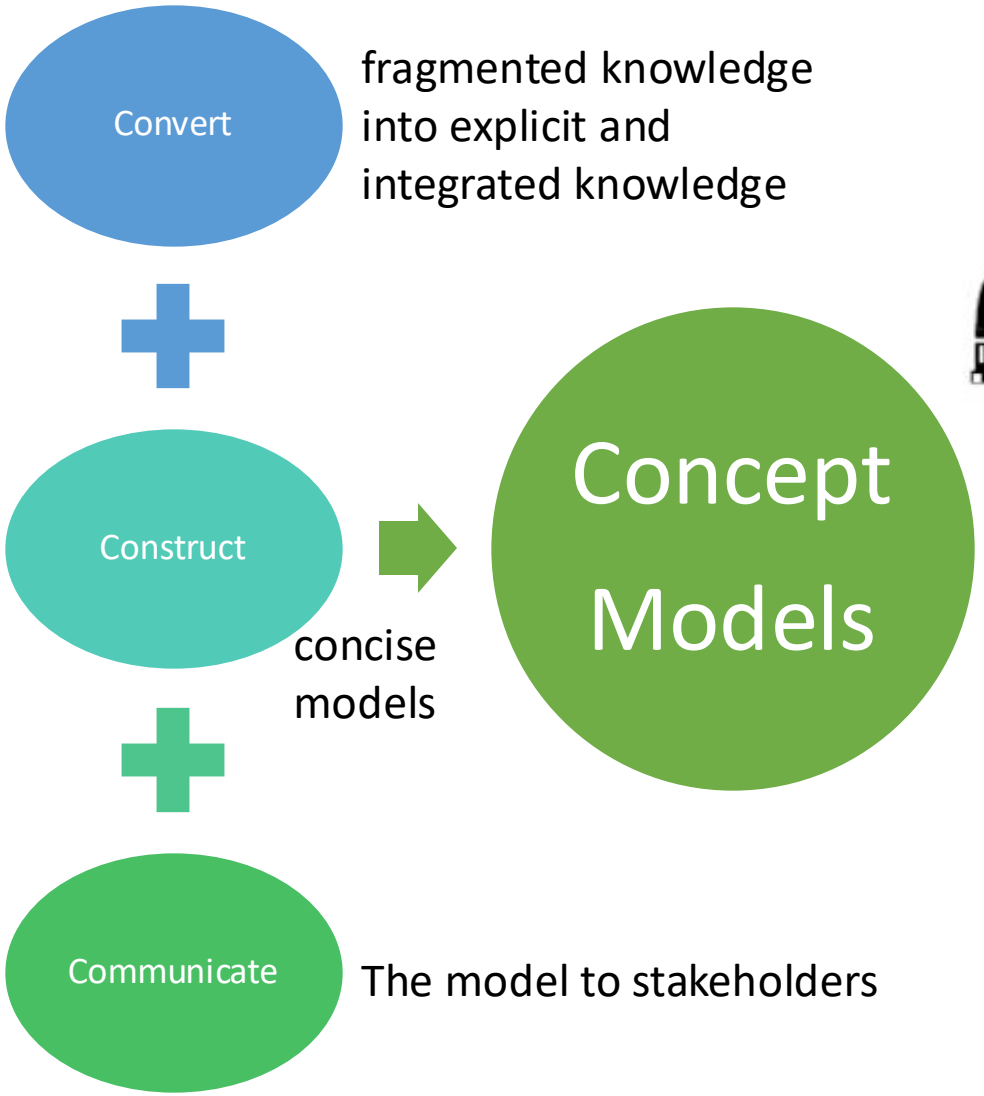


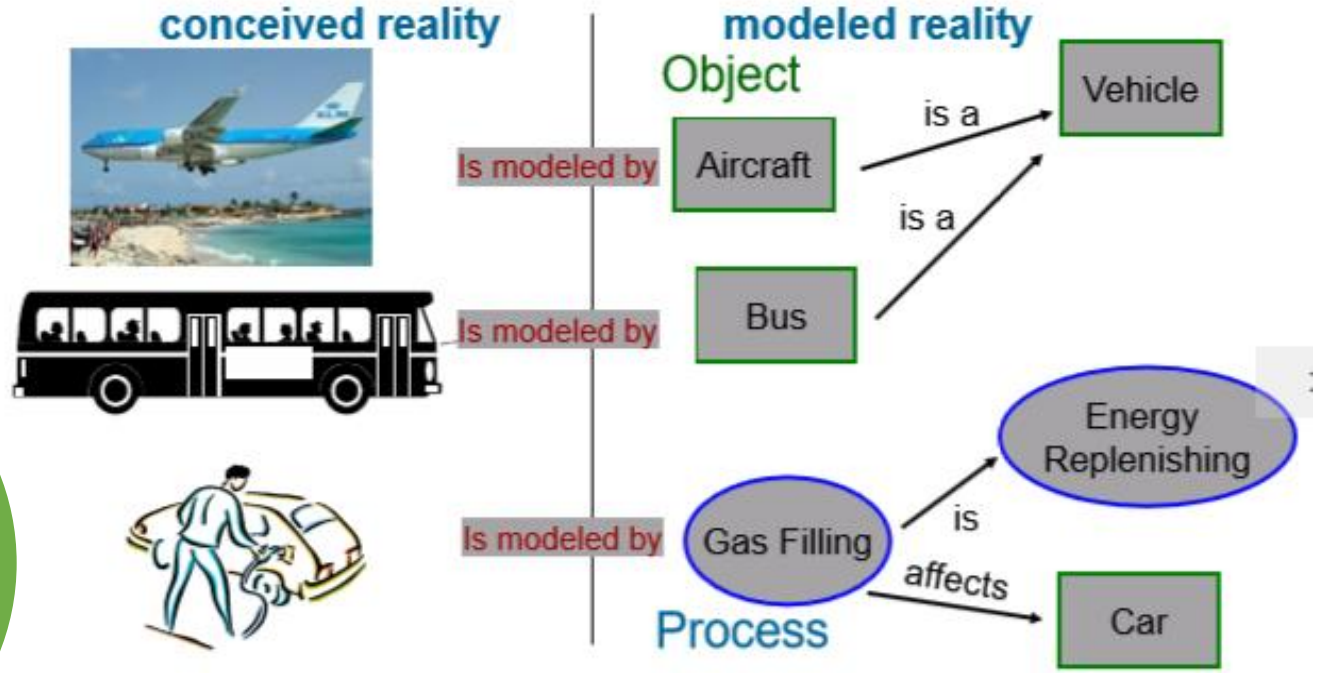
While talking, run lines with or without arrows among the shapes

Follow the reaction of the audience to see if idea is understood

Answer questions, continue scribbling…

Conceptual Modelling





- **simple** yet **expressive**, and
- **intuitive** yet **formal**

These system “first” ideias → Conceptual Modelling

- A systematic, formalized process of describing, specifying, designing or explaining ideas, systems, products or processes through a model

- Applicable to both:
  - **Science** – Studying what is known and what is missing to satisfy human thirst for knowledge, and
  - **Engineering** – Designing systems to benefit humans, based on sound scientific principles

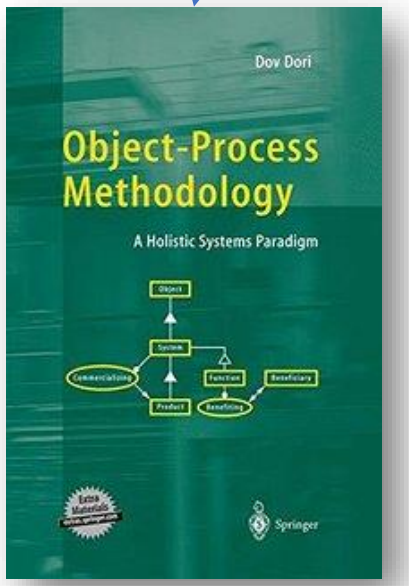
- Science can be thought of as reverse engineering of nature

OPM

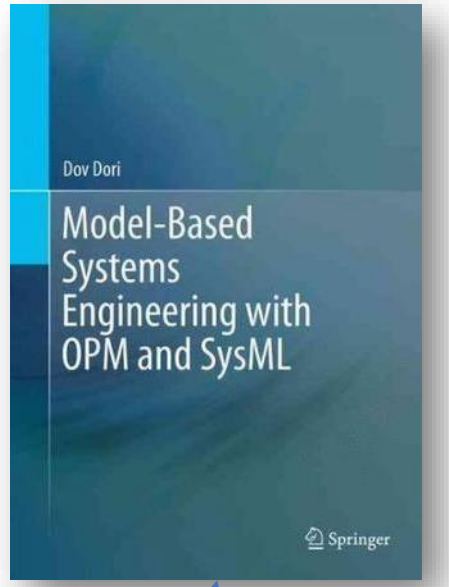
Prof. Dov Dori

Created in 2002

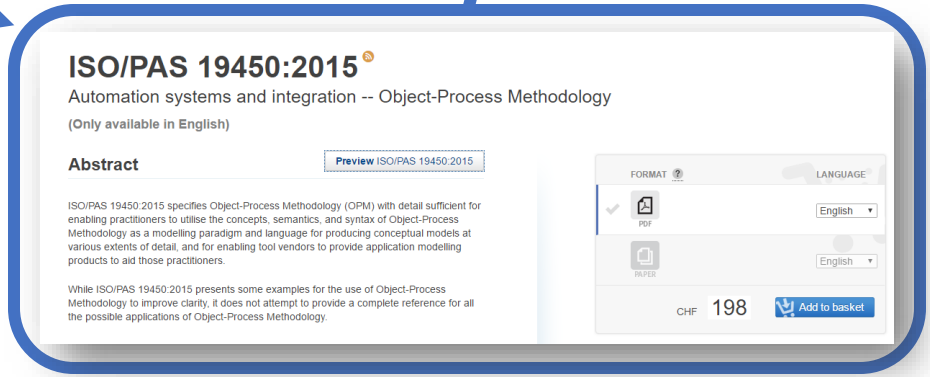
- **A** Single Diagram – Maps Behaviour and Structure
- **2 Building Blocks and 10 basic relations**
- Designed to “Systemic View” and “Concept Modelling”
- Simulation Ready

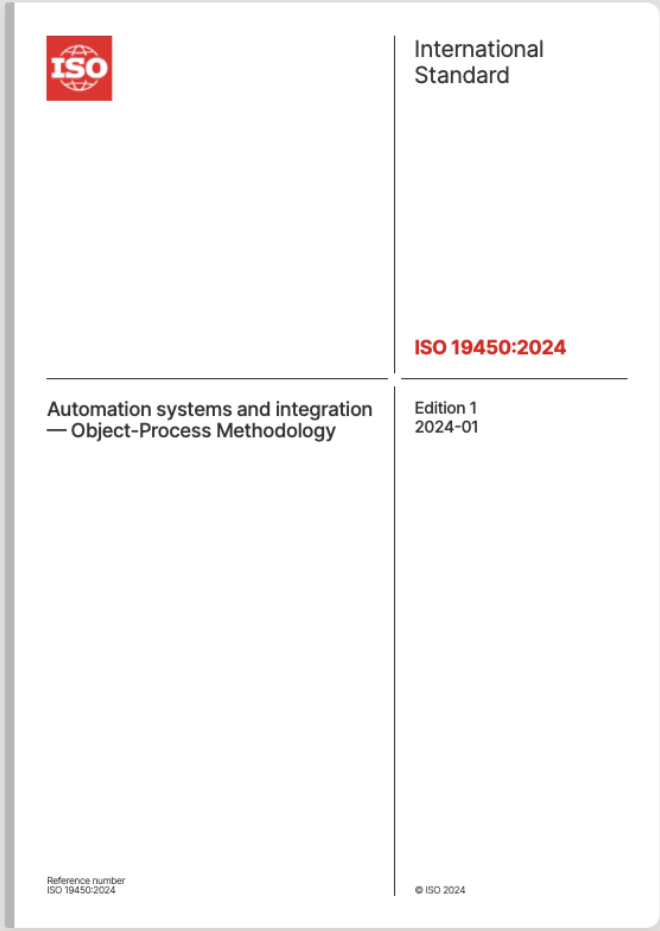


improving and showing it applicability



- ~130 Pages standard
- Published in late 2015
- Intended to “*Automation Systems and Integration*”
- Has the “**power**” of a ISO seal.



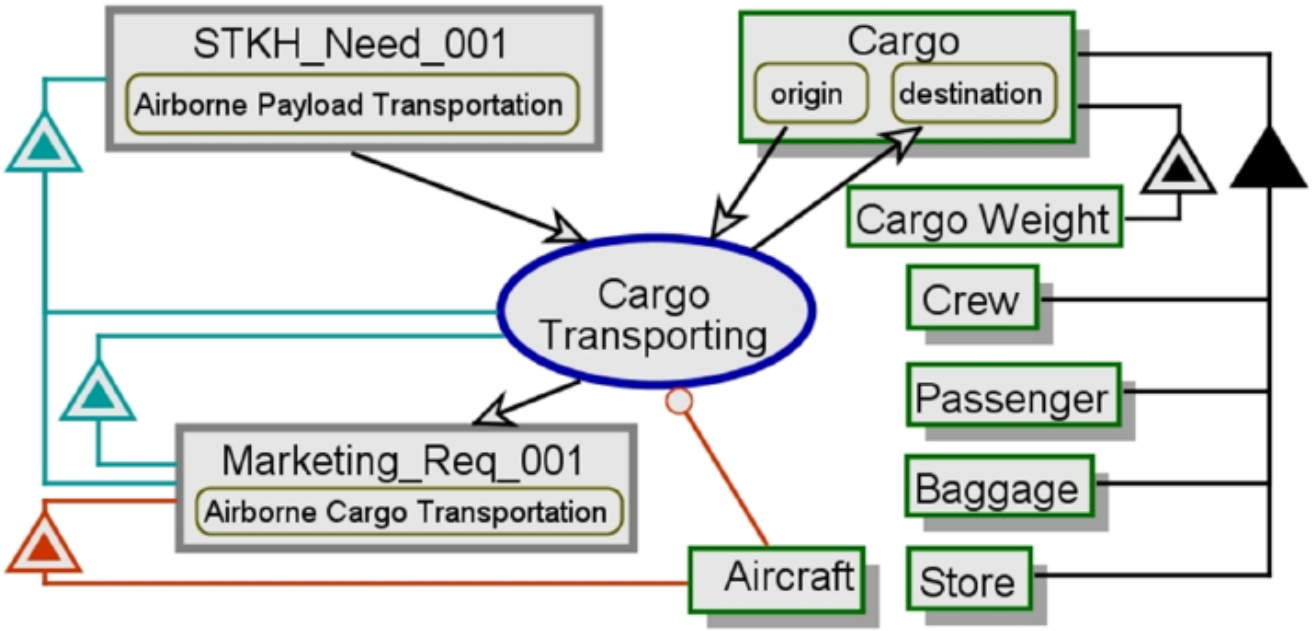


Read sample

ISO 19450:2024

Automation systems and integration — Object-Process Methodology

Published (Edition 1, 2024)



STKH_Need_001 is Airborne Payload Transportation.
Cargo Transporting consumes STKH_Need_001.
Cargo Transporting yields Marketing_Req_001.
Marketing_Req_001 is Airborne Cargo Transportation.
Marketing_Req_001 exhibits Aircraft.
Cargo Transporting requires Aircraft.
Aircraft is physical.
Cargo Transporting changes Cargo from origin to destination.

Function Defining
Requirements Identifying
Requirements Allocating

STKH_Need_001 exhibits Marketing_Req_001, as well as Cargo Transporting.
Cargo Transporting exhibits Marketing_Req_001.

Traceability

Cargo is physical.
Cargo can be origin or destination.
Cargo exhibits Cargo Weight.
Cargo consists of Passenger, Baggage, Store, and Crew.
Passenger is physical.
Baggage is physical.
Store is physical.
Crew is physical.

Configuration management

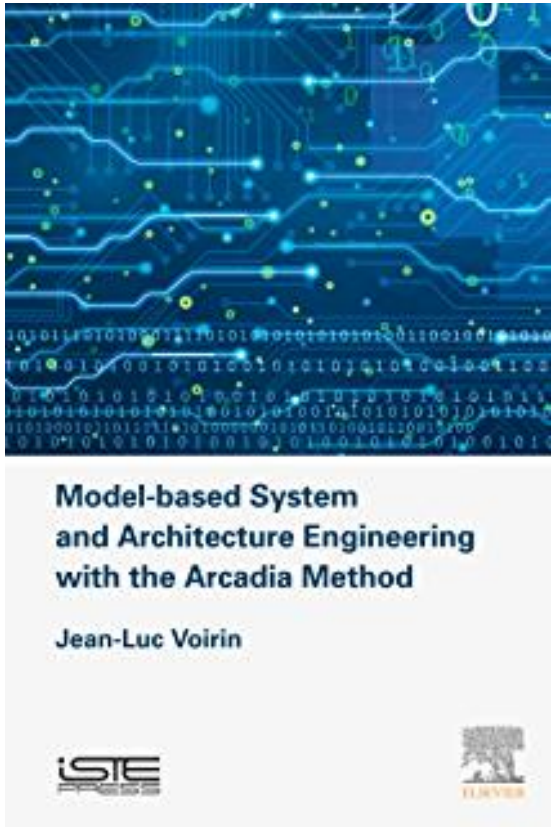
Model-Based Systems Engineering for Aircraft Design with Dynamic Landing Constraints Using Object-Process Methodology , AIAA Scitech 2019 Forum

ARCADIA

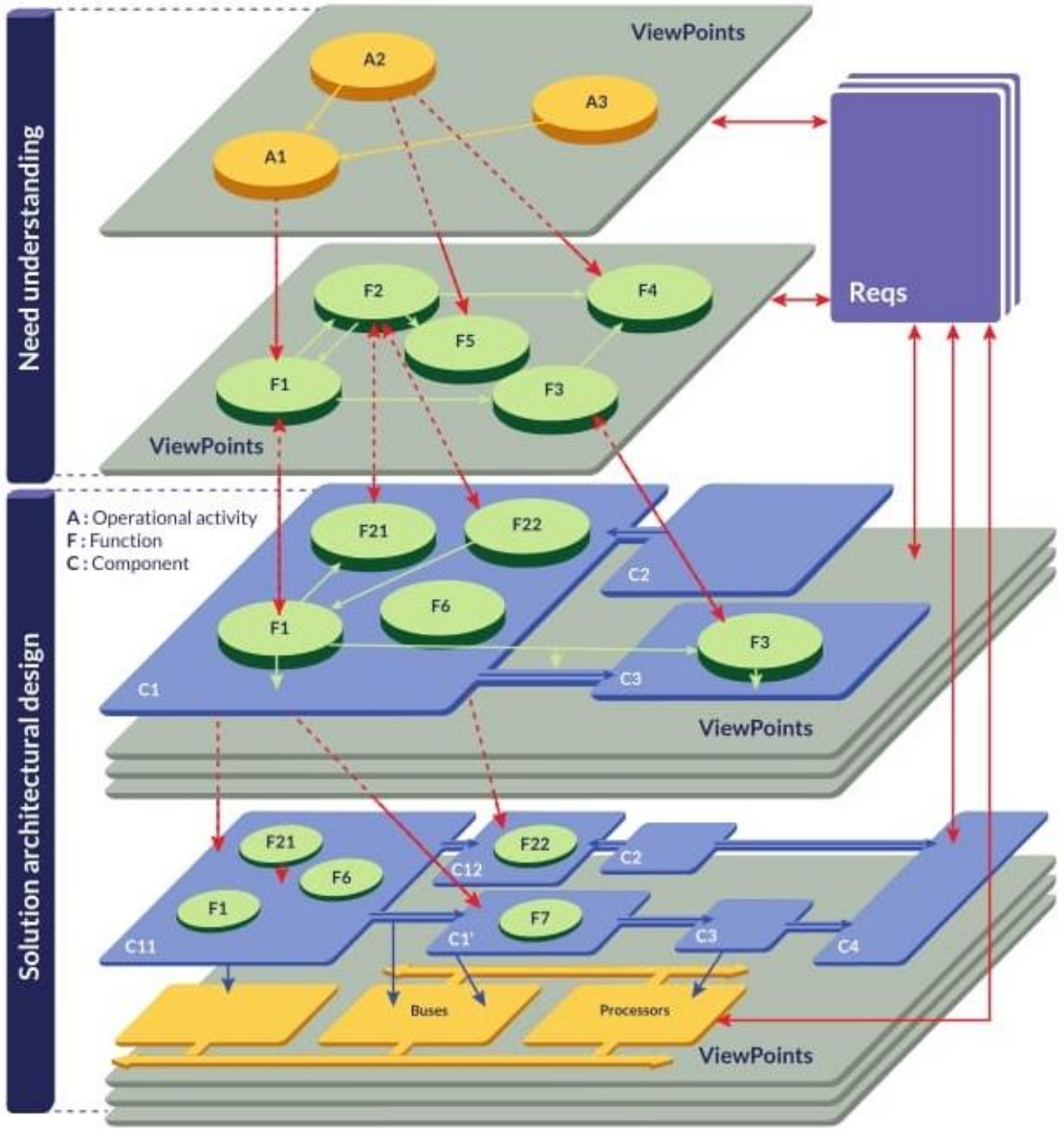
Founding principles

- all of the engineering stakeholders share the **same methodology**, the **same information**, **the same description of the need** and the product in the form of a shared model;
- each **specialized type of engineering** (for example security, performance, cost and mass) **is formalized as a "viewpoint"** in relation to the requirements from which the proposed architecture is then verified;
- the rules for the **anticipated verification of the architecture** are established in order to verify the architecture as soon as possible;
- **co-engineering between the different levels** of engineering is supported by the joint elaboration of models, and the models of the different levels and specialties are deducted/validated/linked one to the other.

XP Z67-140 - ARCADIA



[Norme XP Z67-140 (afnor.org)]

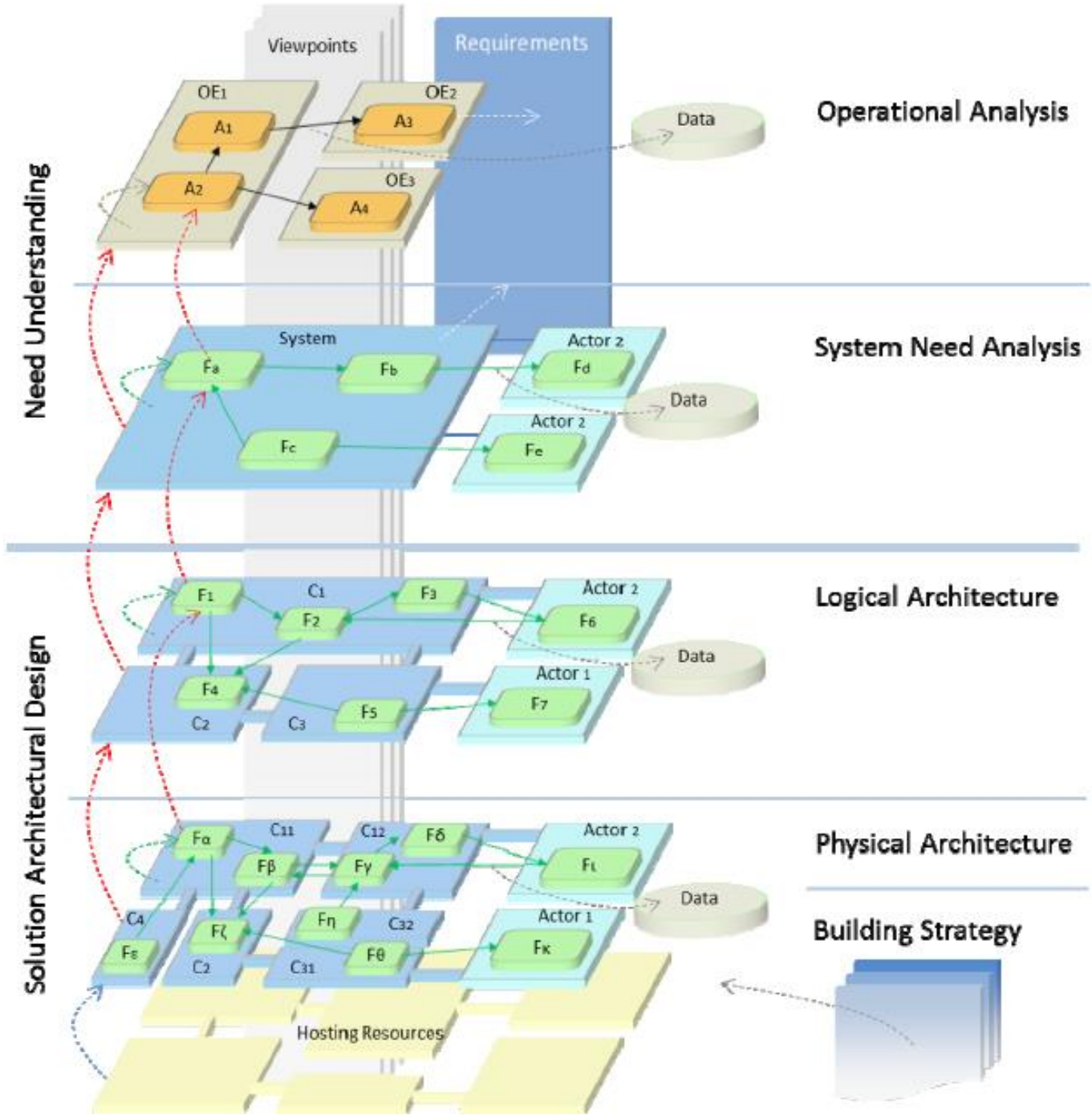


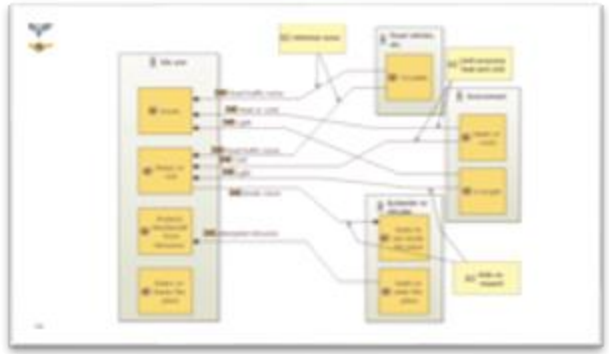
Operational Analysis
What the users of the system need to accomplish

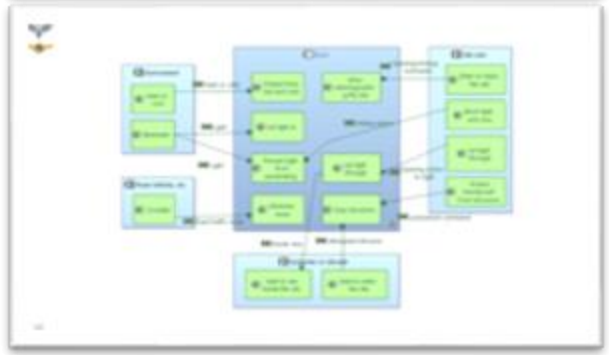
Functional & Non Functional Need
What the system has to accomplish for the users

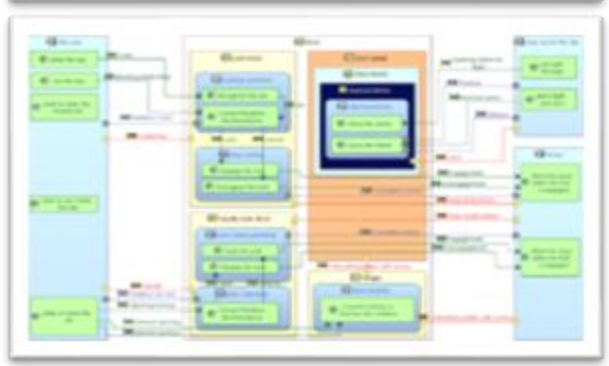
Logical Architecture
How the system will work to fulfill expectations

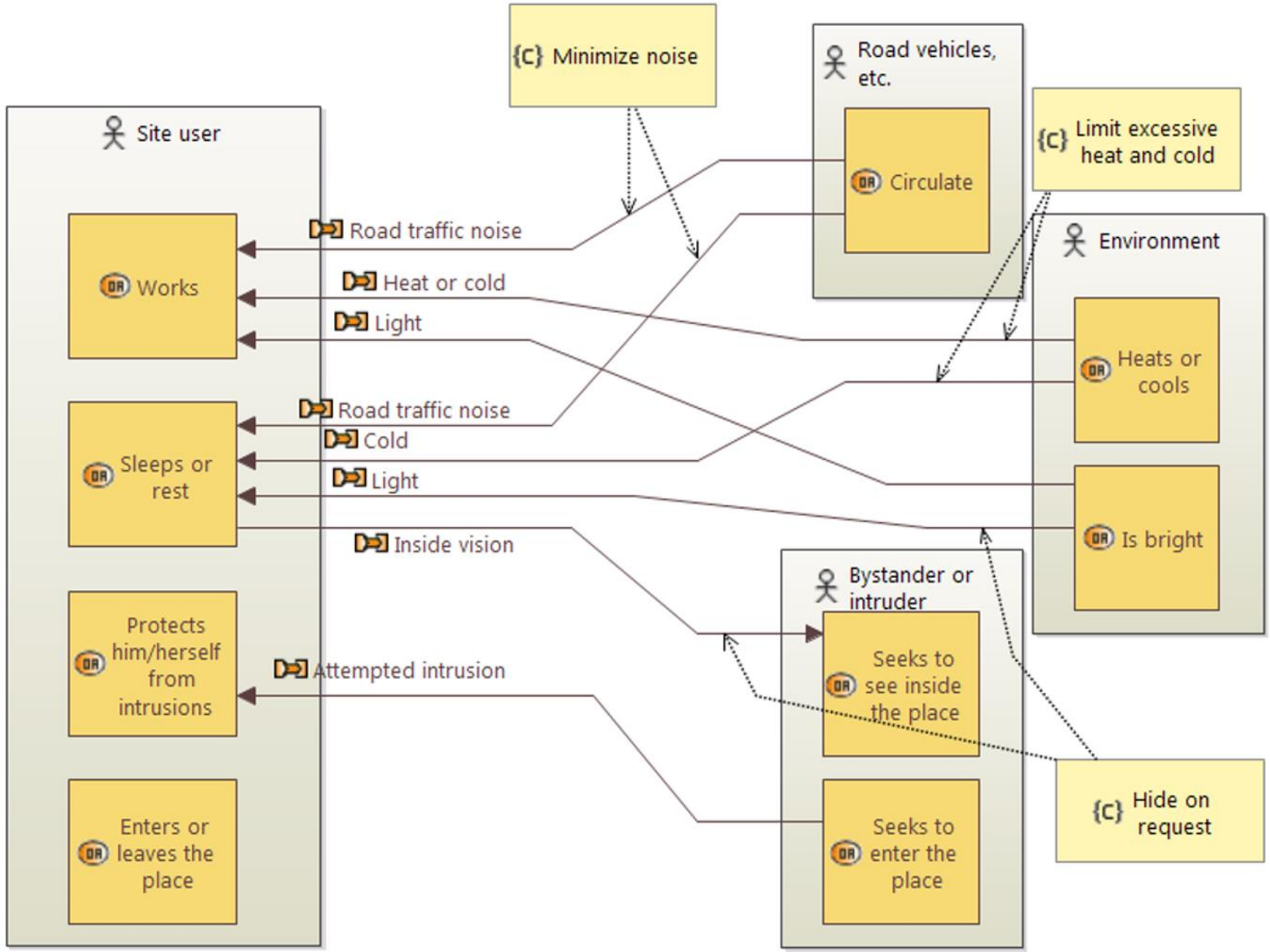
Physical Architecture
How the system will be developed and built

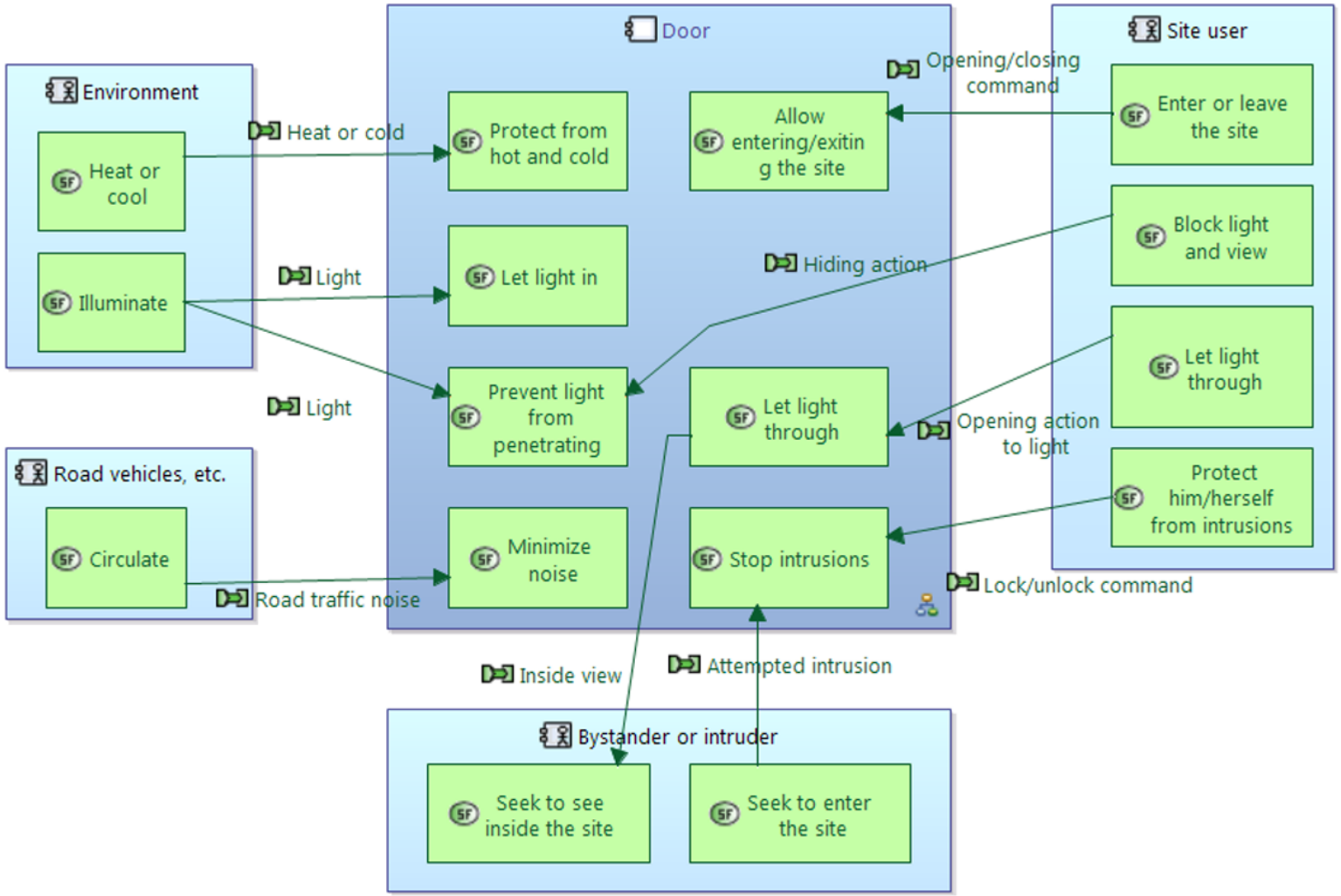

Viewpoints
Requirements
Need Understanding
Solution Architectural Design
OE1
OE2
OE3
A1
A2
A3
A4
Data
Operational Analysis
System
Actor 2
Actor 2
Actor 1
Fa
Fb
Fc
Fd
Fe
Data
System Need Analysis
F1
C1
F2
F3
F4
C2
C3
F5
F6
F7
Logical Architecture
Fα
C11
Fβ
C12
Fγ
Fδ
Fε
C4
Fζ
C2
Fη
C31
C32
Fθ
Fι
Fκ
Physical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

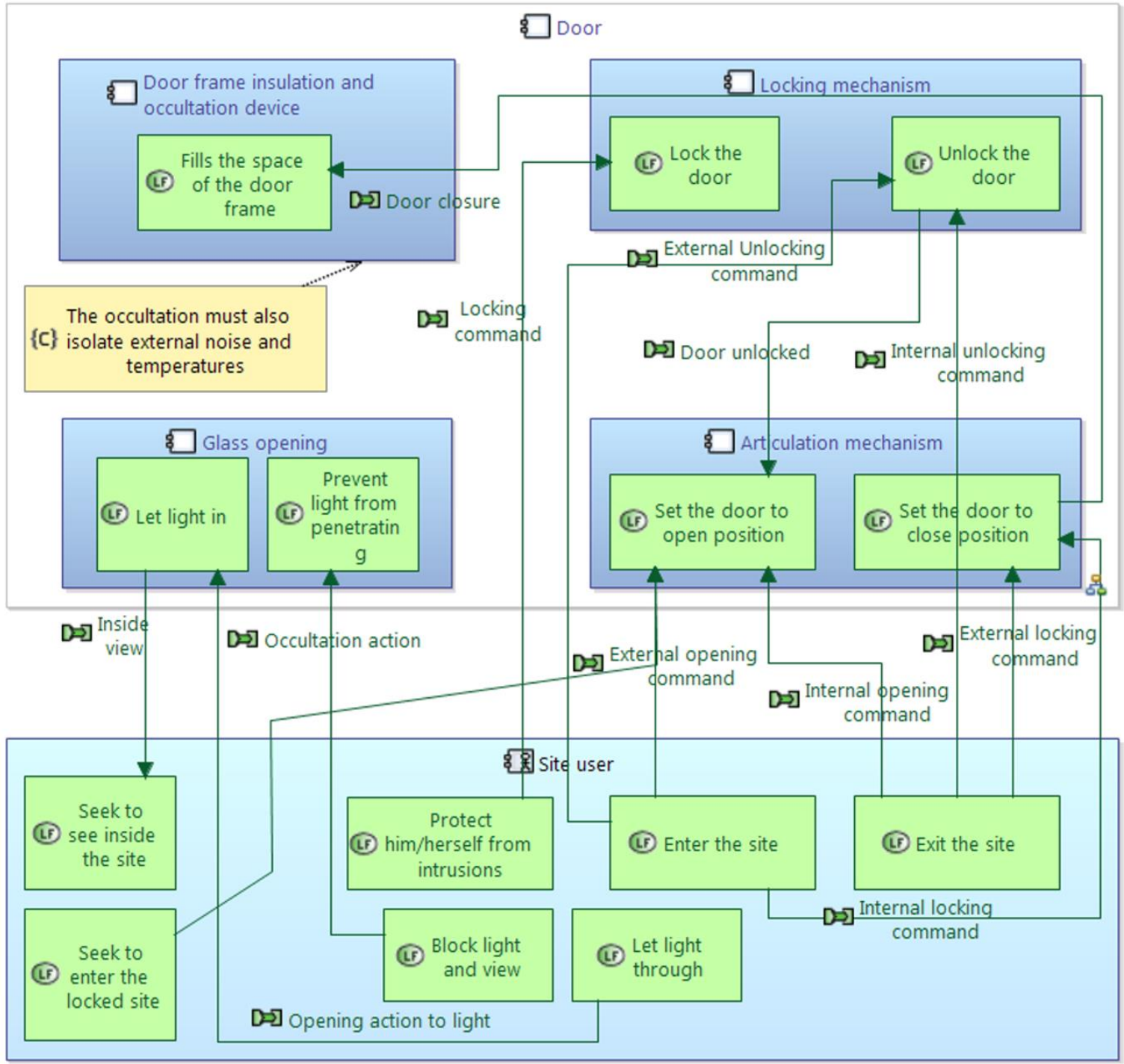


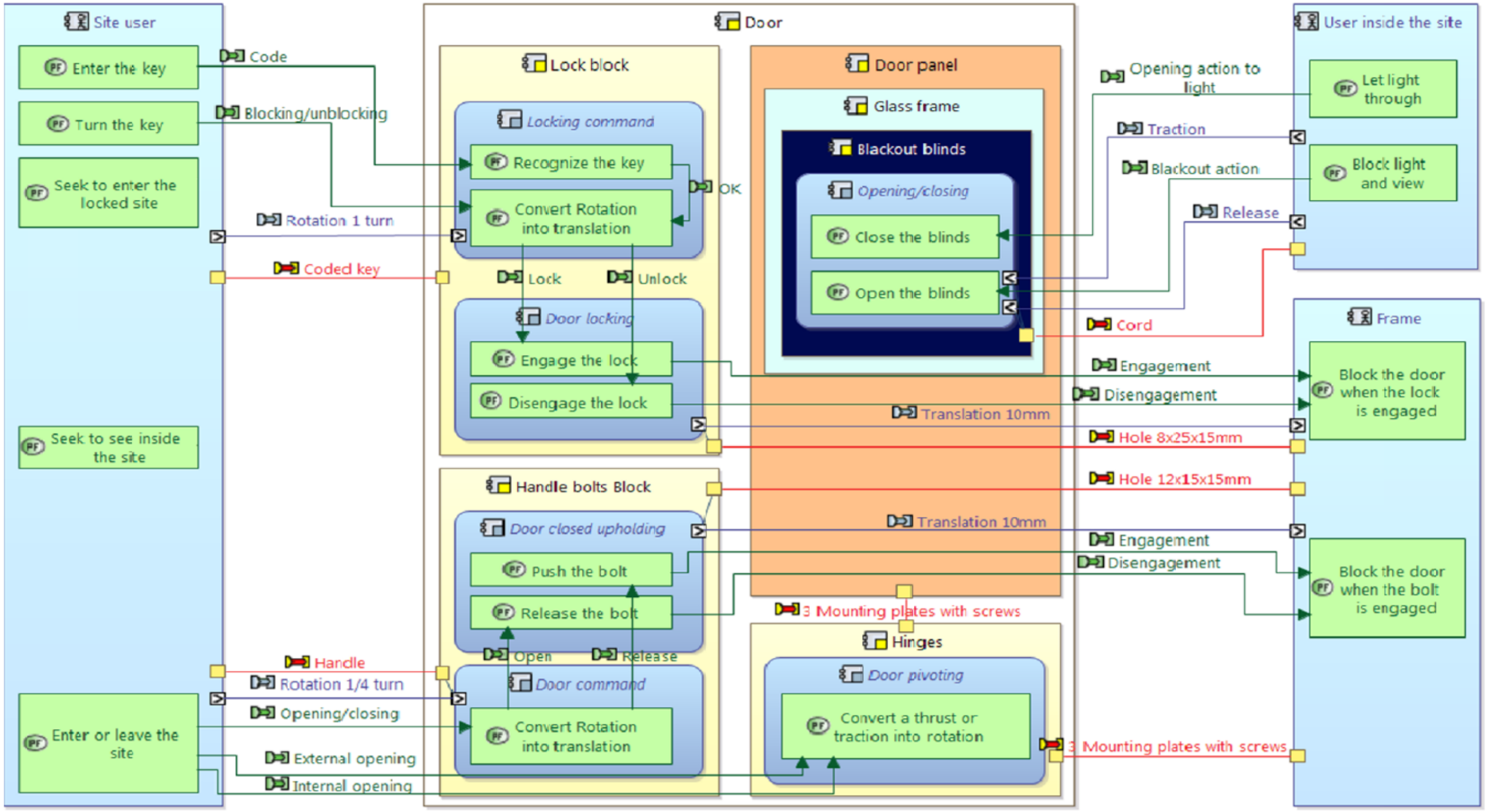





{C} Minimize noise
Road vehicles, etc.
Site user
Circulate
{C} Limit excessive heat and cold
Environment
Road traffic noise
Works
Heat or cold
Light
Heats or cools
Road traffic noise
Cold
Sleeps or rest
Light
Is bright
Inside vision
Bystander or intruder
Protects him/herself from intrusions
Attempted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


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


Door
Door frame insulation and occultation device
Fills the space of the door frame
Door closure
Locking mechanism
Lock the door
Unlock the door
External Unlocking command
The occultation must also isolate external noise and temperatures
Locking command
Door unlocked
Internal unlocking command
Glass opening
Let light in
Prevent light from penetratin g
Articulation mechanism
Set the door to open position
Set the door to close position
Inside view
Occultation action
External opening command
External locking command
Internal opening command
Site user
Seek to see inside the site
Protect him/herself from intrusions
Enter the site
Exit the site
Internal locking command
Seek to enter the locked site
Block light and view
Let light through
Opening action to light


Site user
Enter the key
Turn the key
Seek to enter the locked site
Seek to see inside the site
Enter or leave the site
Code
Blocking/unblocking
Rotation 1 turn
Coded key
Handle
Rotation 1/4 turn
Opening/closing
External opening
Internal opening
Door
Lock block
Locking command
Recognize the key
Convert Rotation into translation
OK
Lock
Unlock
Door locking
Engage the lock
Disengage the lock
Handle bolts Block
Door closed upholding
Push the bolt
Release the bolt
Open
Release
Door command
Convert Rotation into translation
Door panel
Glass frame
Blackout blinds
Opening/closing
Close the blinds
Open the blinds
Translation 10mm
Translation 10mm
3 Mounting plates with screws
Hinges
Door pivoting
Convert a thrust or traction into rotation
3 Mounting plates with screws
Opening action to light
Traction
Blackout action
Release
Cord
Engagement
Disengagement
Hole 8x25x15mm
Hole 12x15x15mm
Engagement
Disengagement
User inside the site
Let light through
Block light and view
Frame
Block the door when the lock is engaged
Block the door when the bolt is engaged

	SysML	Arcadia/Capella
Positioning	SysML is a standard and a general-purpose modeling language for modeling systems. SysML provides very rich and advanced expression means covering a very broad spectrum of applications, spanning from high-level architecture modeling to detailed design at the frontier of simulation.	Inspired by SysML concepts, the Arcadia/Capella solution focuses on the design of systems architectures. For the sake of an easier learning curve and because of the precise scope addressed by Arcadia/Capella, the expression means are sometimes reduced compared to SysML. The goal of Arcadia/Capella is to **have systems engineers embrace the cultural change of MBSE rather than having modeling "experts" owning the model** on behalf of systems engineers. Therefore, Arcadia/Capella are strongly driven by the current practices and concerns of system engineering practitioners.
Method	SysML is not associated to a particular method even though several engineering methods can be followed. As such, SysML only provides a vocabulary, but it does not specify when to use one concept or another, how to organize models, etc.	The Arcadia method enforces an **approach structured** on different engineering perspectives establishing a clear separation between system context and need modeling (operational need analysis and system need analysis) and solution modeling (logical and physical architectures), in accordance with the IEEE 1220 standard and covering parts of ISO/IEC/IEEE 15288.

	SysML	Arcadia/Capella
Language	Technically, the SysML language itself is defined as an extension of the Unified Modeling Language (UML). Both UML and SysML are general-purpose languages targeting wide spectrums of engineering domains and are relying on software-originated engineering paradigms using types, inheritance, etc.	The Arcadia concepts are mostly similar to the UML/SysML standard (about 75%) and the NATO Architecture Framework (NAF) standard (5%). Interoperability with SysML tools is possible using ad-hoc imports/exports. Because of the **focus on architectural design**, some of the SysML concepts have been simplified or specialized to better **match the concepts system engineering practitioners already use in their engineering documents** and assets. This is the case of the concepts related to functional analysis for instance.
Diagrams	SysML includes diagrams inherited from UML2 and adds new diagrams: • 4 diagrams are the same as UML2 diagrams (Sequence, State Machine, Use Case and Package); • 3 diagrams are extensions of UML2 diagrams (Activity, Block definition and Internal Block); • 2 diagrams are new diagram types (Requirement and Parametric).	Arcadia method is supported by various kinds of diagrams largely inspired by UML and SysML: • Architecture diagrams; • Dataflows diagrams; • Functional chains diagrams; • Sequence diagrams; • Tree diagrams; • Mode and States diagrams; • Classes and Interfaces diagrams.

Last Considerations

- SE is a broad engineering that focus on value delivering.
  - A lot of people think that is Project Management: Only breakdown, wbs, and so on..
  - SE is more a world view.
  - Capability to understand entities and relations and the emergence of it
- MBSE is “just” SE with models / language
  - SE will incorporate the (MB) as a unique thing... as the any engineering

AI-03

- Summary of the INCOSE’s reviewed methodology
- One page/slide per methodology

- Skip OPM/ARCADIA - we will cover them into the next sessions.