MBSE Introduction

[MBSE][LEC-001]

CONTEXT

Time

Formulation

Implementation

Pre-Phase A (Phase 0 to ECSS) – Concept Studies

Phase A – Concept and Technology Development

Phase B – Preliminary Design and Technology Completion

Phase C – Final Design and Fabrication

Phase D – System Assembly, Integration and Test, Launch

Phase E – Operations and Sustainment

Phase F – Closeout

Problem

Solution

Napkin Model

Operational Analysis

System Analysis

Logical Architecture

Physical Architecture

EPBS

Using MBSE through the other Phases

MDE

System Engineering

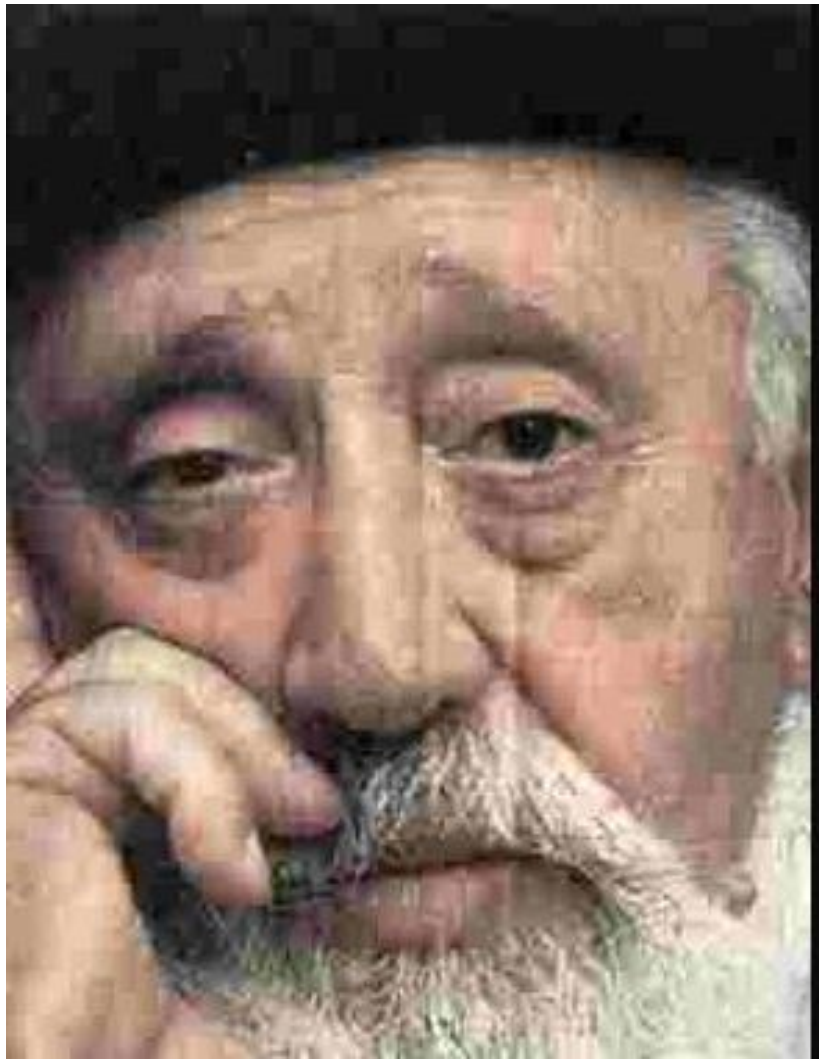
MBSE

OPM

ARCADIA

Related Analysis

Future Perspective



A goal of education is. to assist growth toward greater complexity and integration and to assist in the process of self-organization - to modify individuals capacity to modify themselves.

— Reuven Feuerstein —

AZ QUOTES

#escolasempartido

[even shorter] SE Review

What is a System?

A system is a

<set, combination, group, collection, configuration, arrangement, organization, assemblage, assembly, ensemble [10]>

of

<parts, components, elements, objects, subsystems, entities [6]>

<combined, integrated, organised, configured, arranged [5]>

in a way that

<creates, enables, motivates [3]>

<properties, functions, processes, capabilities, behaviors, dimensions [6]>

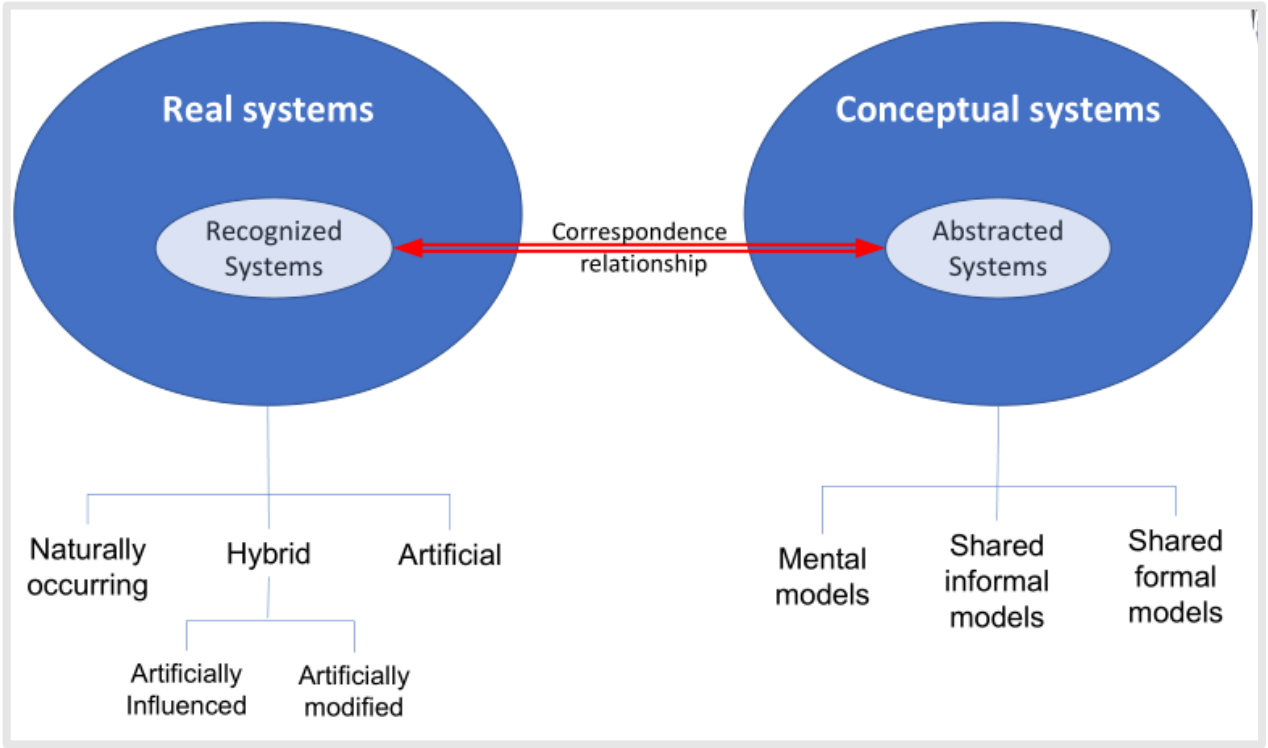
not

<possessed, exhibited, presented [3]>

by the

<separate, individual, single [3]>

<parts, components, elements, objects, subsystems, entities [6]>



Defining "System": a Comprehensive Approach. 27th Annual INCOSE International Symposium. 2017

What is Systems Engineering?! (newer def)

Systems Engineering is a transdisciplinary approach and means, based on systems principles and concepts, to enable the successful realization, use and retiral of engineered systems.

It focuses on

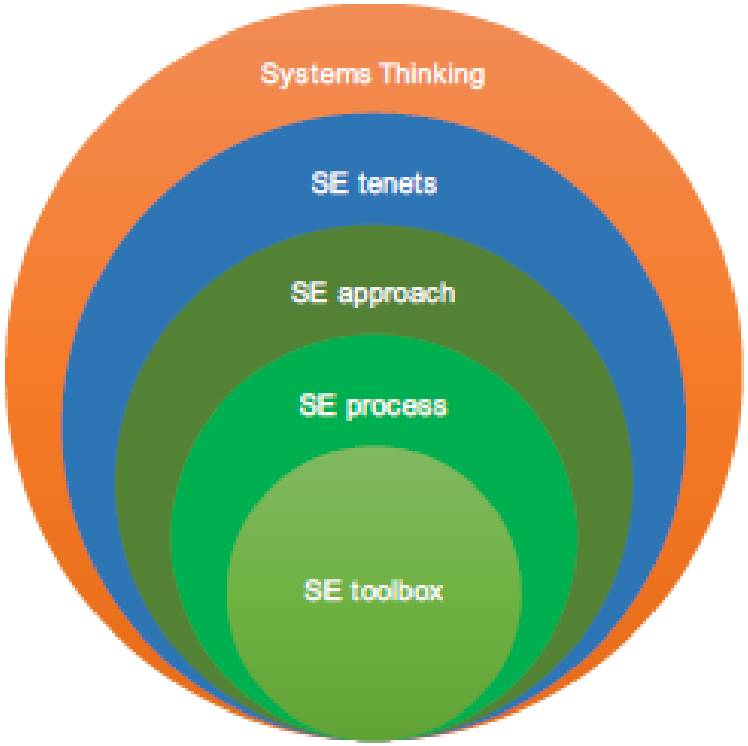
- establishing stakeholders' purpose and success criteria, and defining actual or anticipated customer needs and required functionality early in the development cycle,
- establishing an appropriate lifecycle model and process approach considering the levels of complexity, uncertainty and change
- documenting and modelling requirements and solution architecture for each phase of the endeavour
- proceeding with design synthesis and system validation
- while considering the complete problem and all necessary enabling systems and services.

Systems Engineering provides facilitation, guidance and leadership to integrate all the disciplines and specialty groups into a team effort forming an appropriately structured development process that proceeds from concept to production to operation, evolution and eventual disposal.

Systems Engineering considers both the business and the technical needs of all customers with the goal of providing a quality solution that meets the needs of users and other stakeholders and is fit for the intended purpose in real-world operation, and avoids or minimizes adverse unintended consequences.

Four Aspects of Systems Engineering

- 1. some very basic and widely applicable **SE tenets** (principles and beliefs);
- 2. a general **SE approach** to complex and complicated problems;
- 3. the "**SE process**", which we see evolving from the current SE process described in the INCOSE SE Handbook into a family of SE process models, targeted towards different system types; and
- 4. a **SE toolbox** of techniques and methods that are widely applicable across the spectrum.



12 Systems Engineering Tenents

1. Understand what success means
2. Consider the whole problem, the whole solution and the full lifecycle
3. Understand and manage interdependencies
4. Adapt the parts to serve the purpose of the whole
5. Recognize that Systems Engineering occurs at multiple levels
6. Base decisions on evidence and reasoned judgement
7. Recognize uncertainty while managing change, risk opportunities and expectations
8. Handle structure and behavior as two complementary aspects of any system
9. Understand and use appropriate feedback
10. Understand and manage value
11. Be both systemic and systematic
12. Respect the people

Who is the major player?!

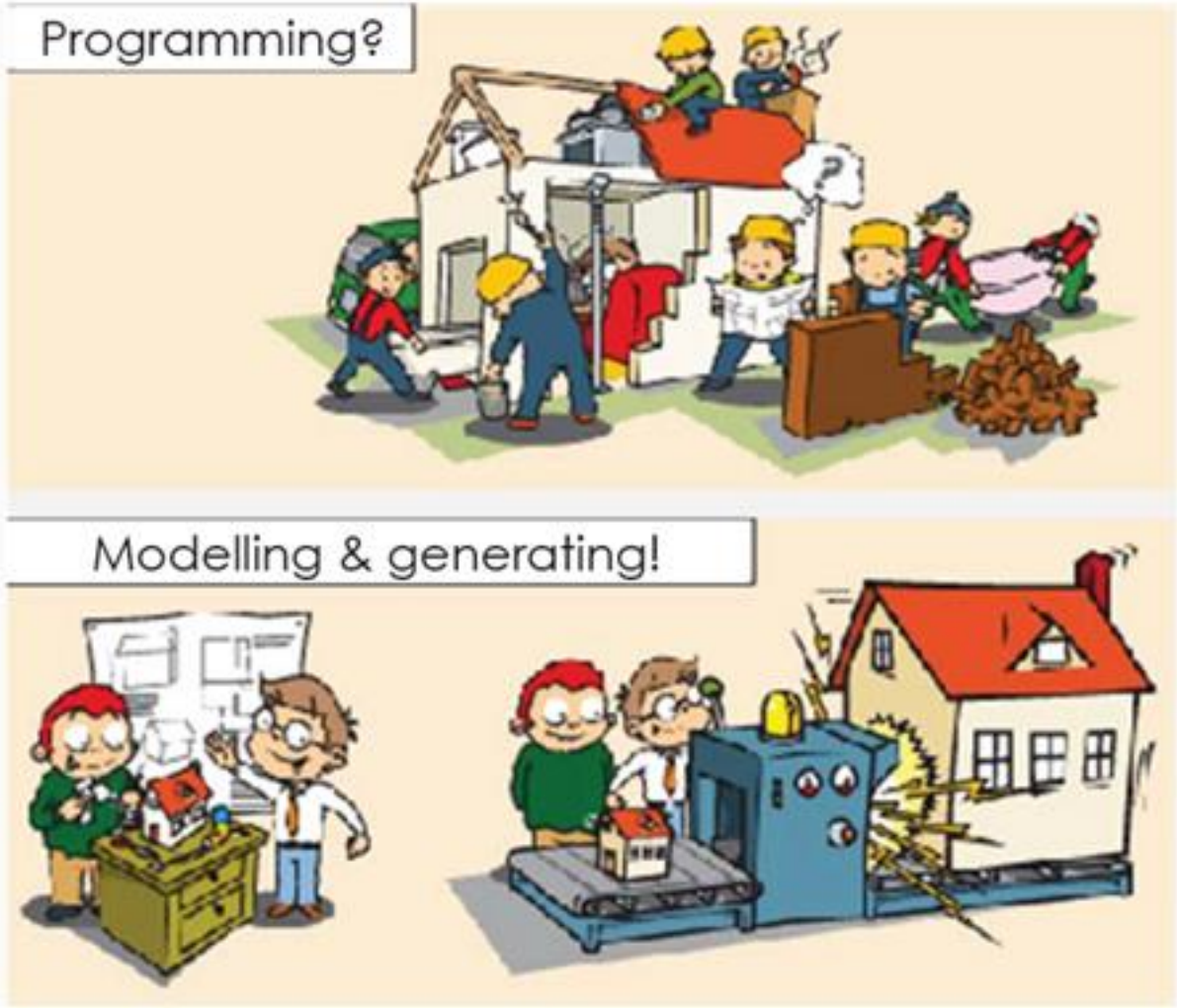
SmallSats → constellations

Drones

60s	80s	00s	Future
Space	Aero	Auto	Bio

Model Driven Engineering

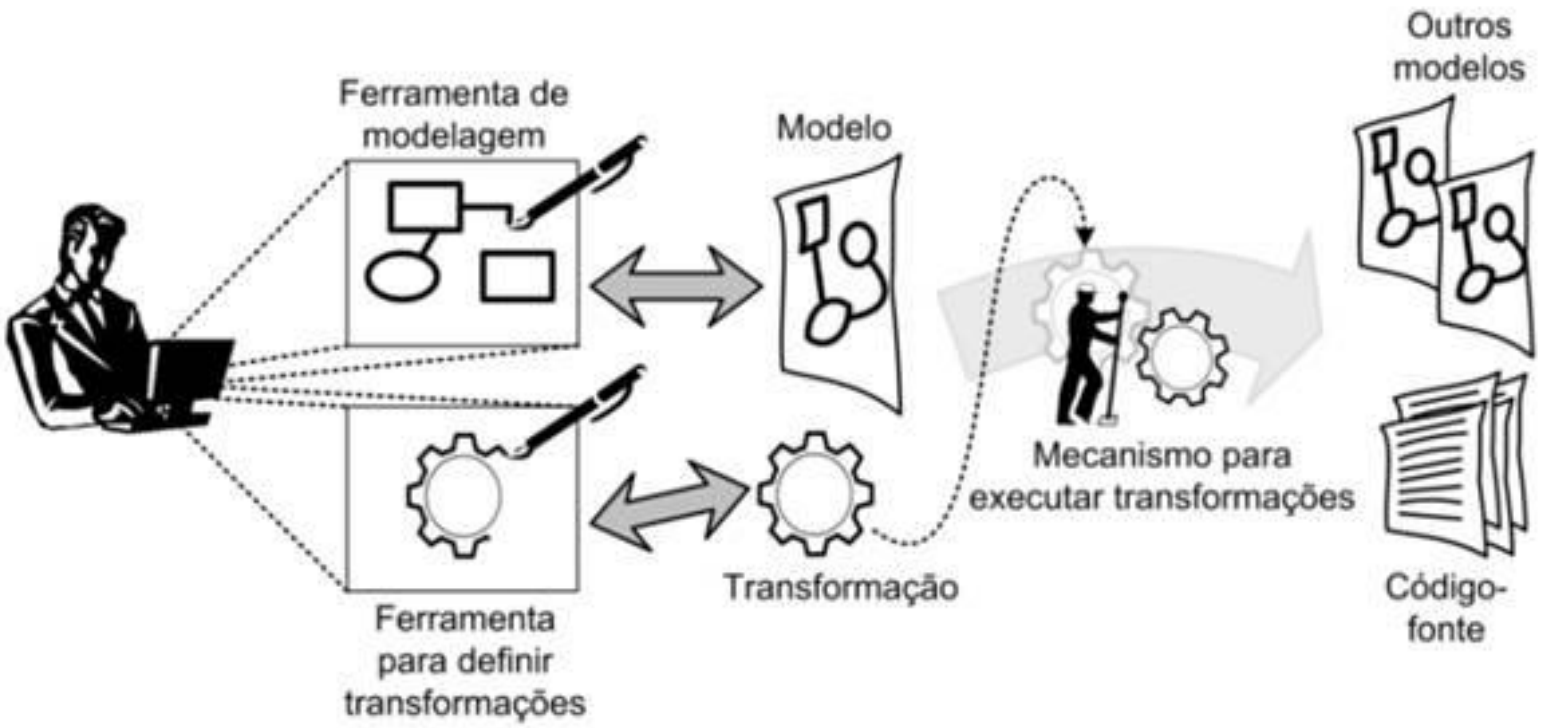
Computation: Models and Simulations



Engineering?

Modelling and auto-build?

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
- ... → infinito

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 directly **generate executable systems**”
- Based: “uses models rather than documents as the **data source** for all activities throughout the product life cycle” “models are used to drive all aspects of the product lifecycle and that data is created once and reused by all downstream data consumers.”
- Centric: “is a focus entirely on **development of information** of and about the system – the model”
- Oriented: “ is a paradigm that does not differentiate between modelling and coding. This is referred to as **model-code duality**.”

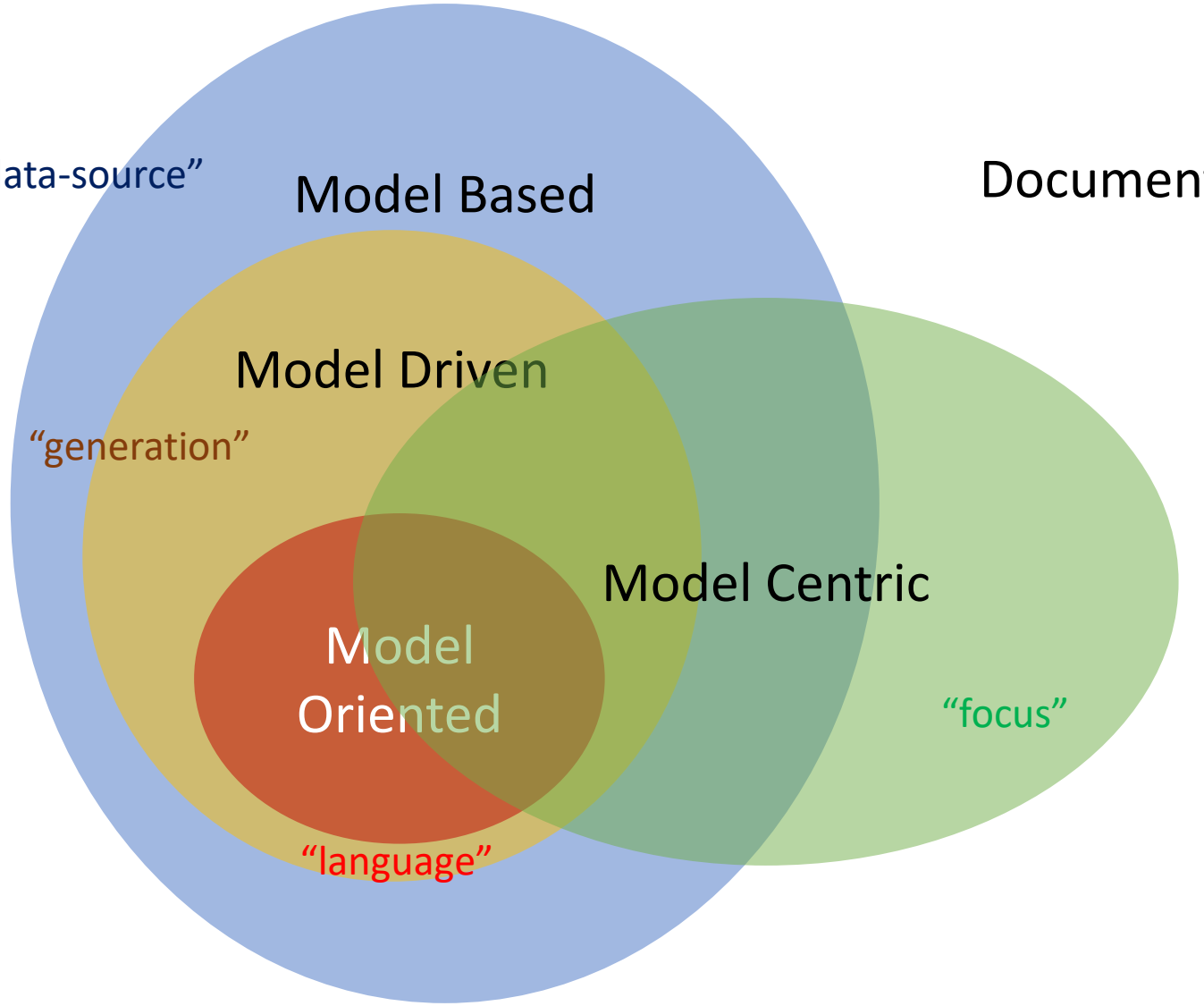
http://onlinelibrary.wiley.com/doi/10.1002/spe.1155/abstract#spe1155-note-0001

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

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

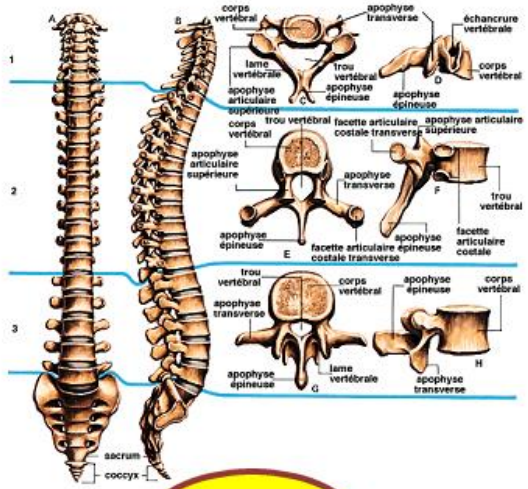
http://cruise.eecs.uottawa.ca/umple/

Model (driven x based x centric x oriented)

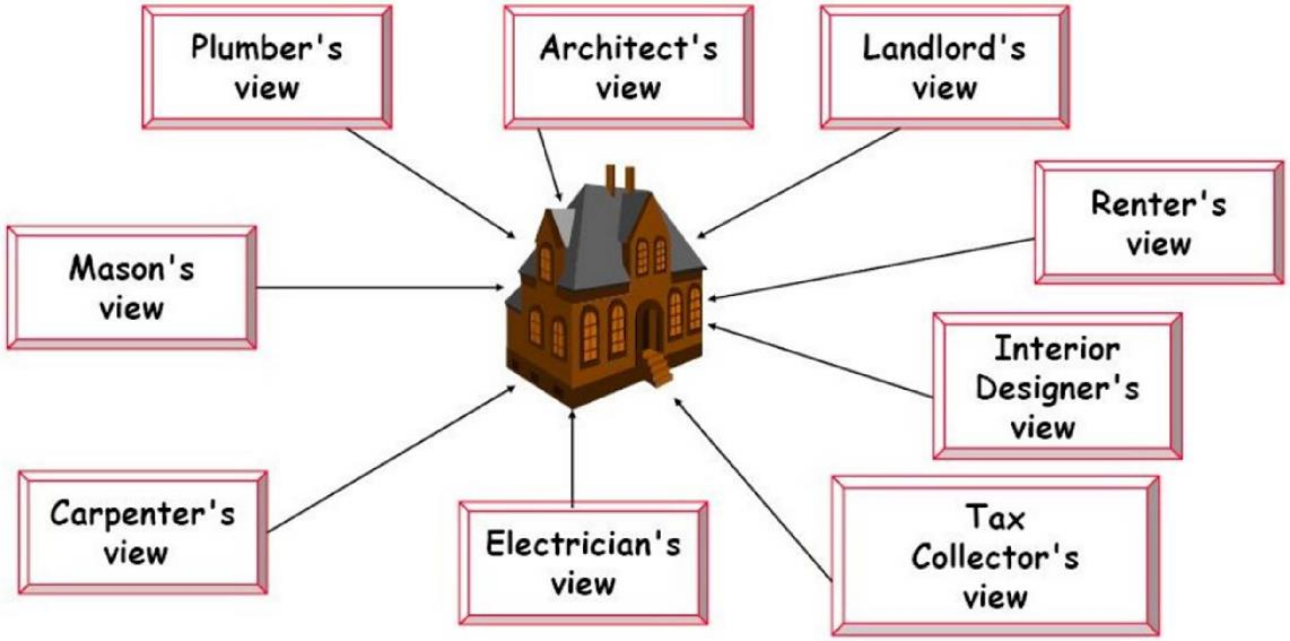


Models

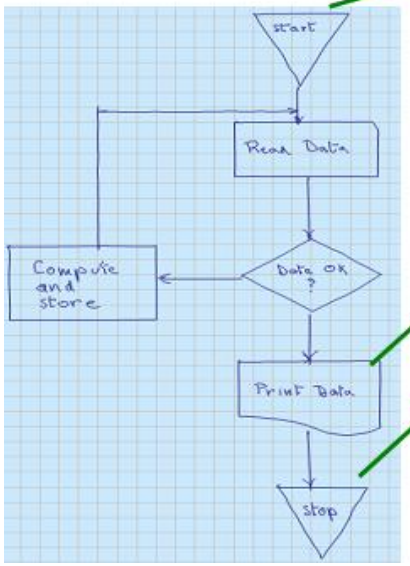
- A model is a **simplified** image of a system.
- "*modullus*", "*modus*" (measure)
  - **What**? (**mapping**)
  - **How**? (**reduction**)
  - **Whom, when, to what**? (**pragmatic**)
- Quite old idea

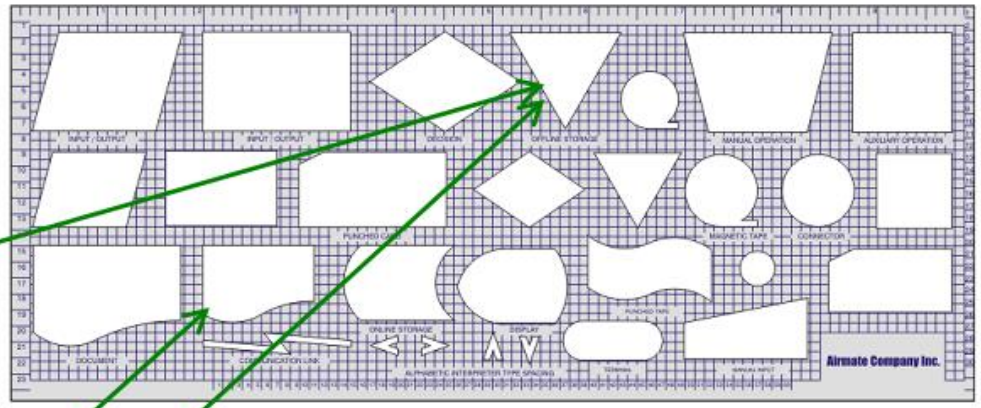


J. Bézivin, Model Engineering for Software Modernization, 2004.

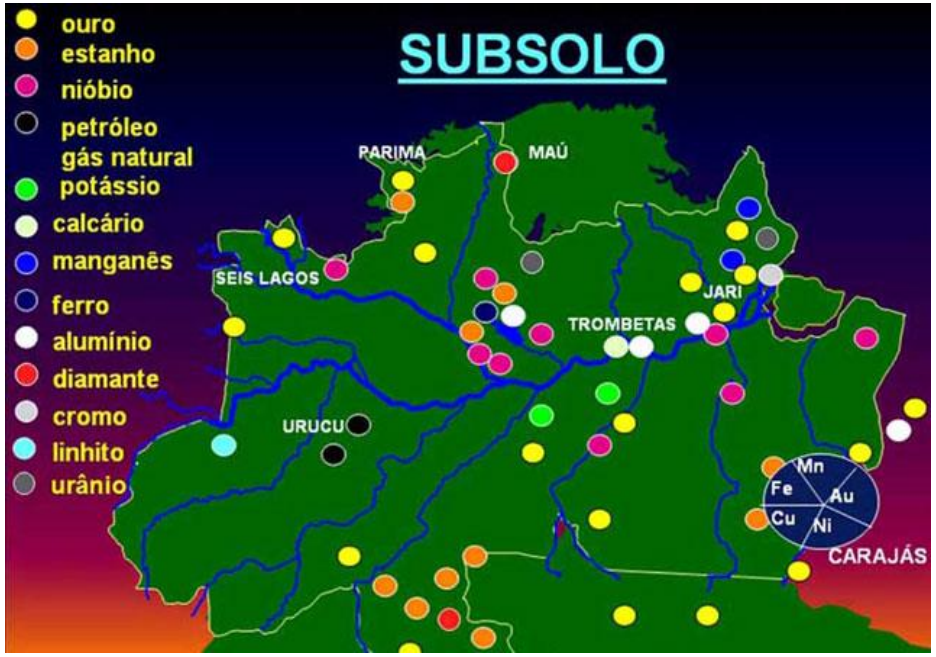


Meta-models

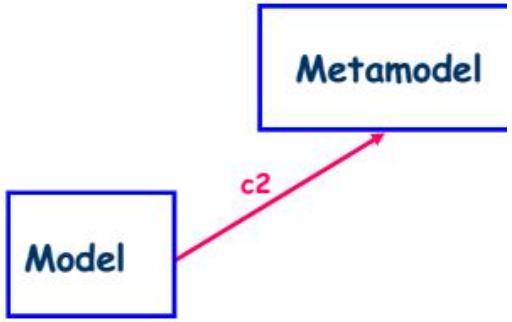




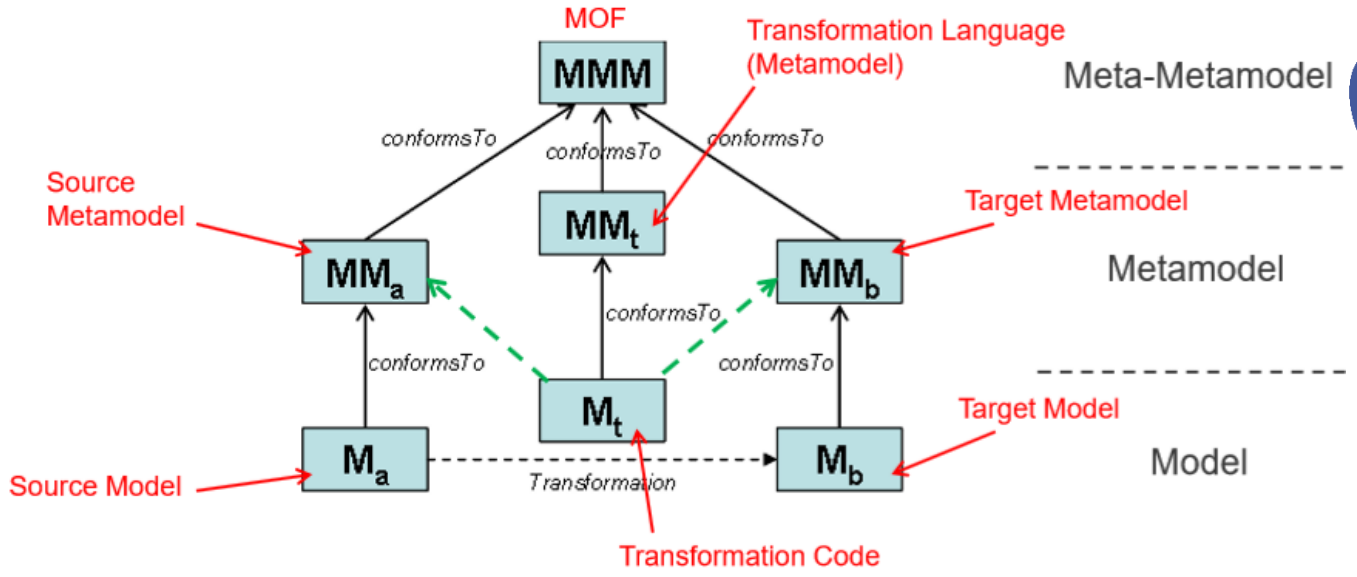
μ

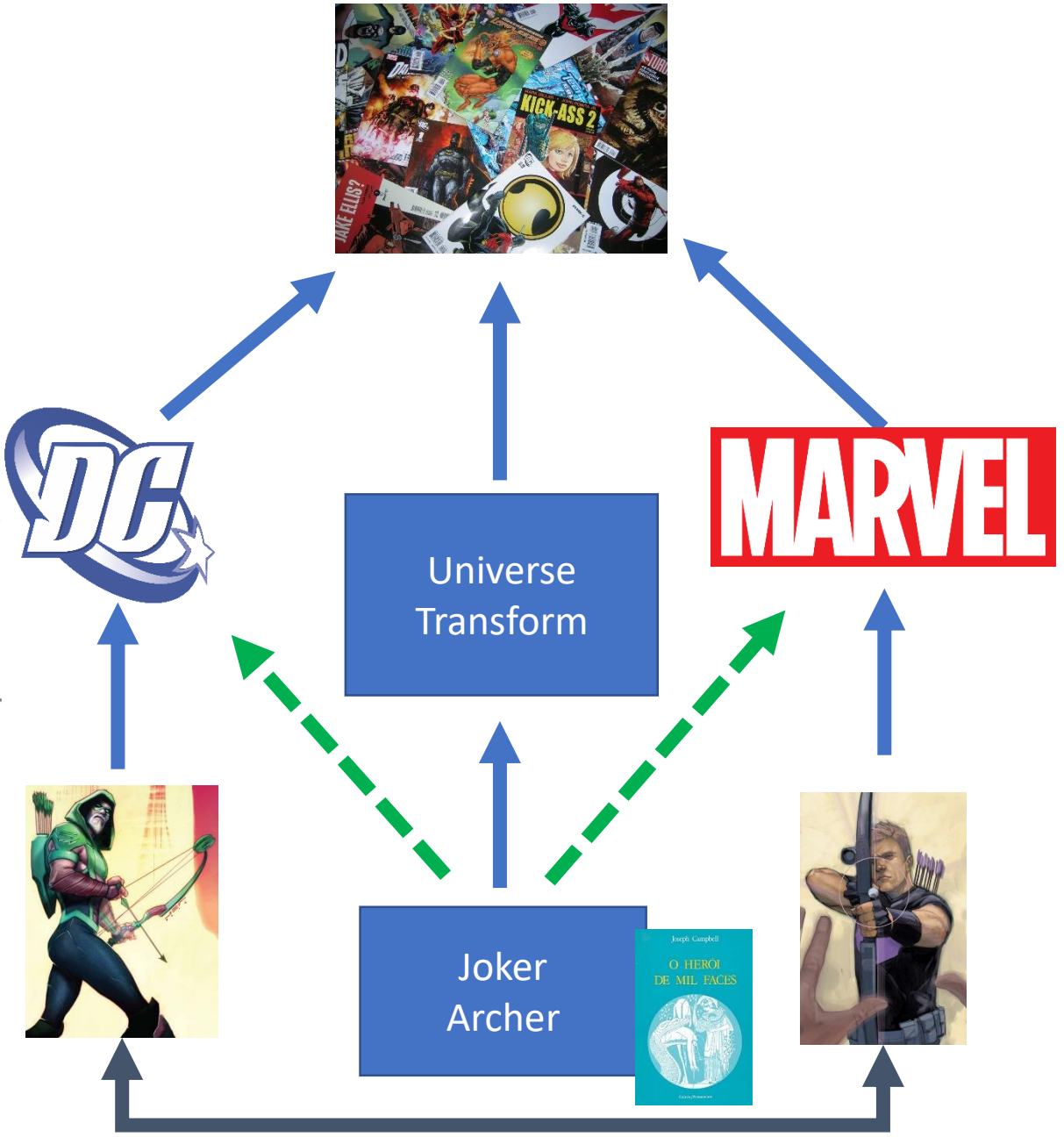


A metamodel is an explicit specification of a shared conceptualization

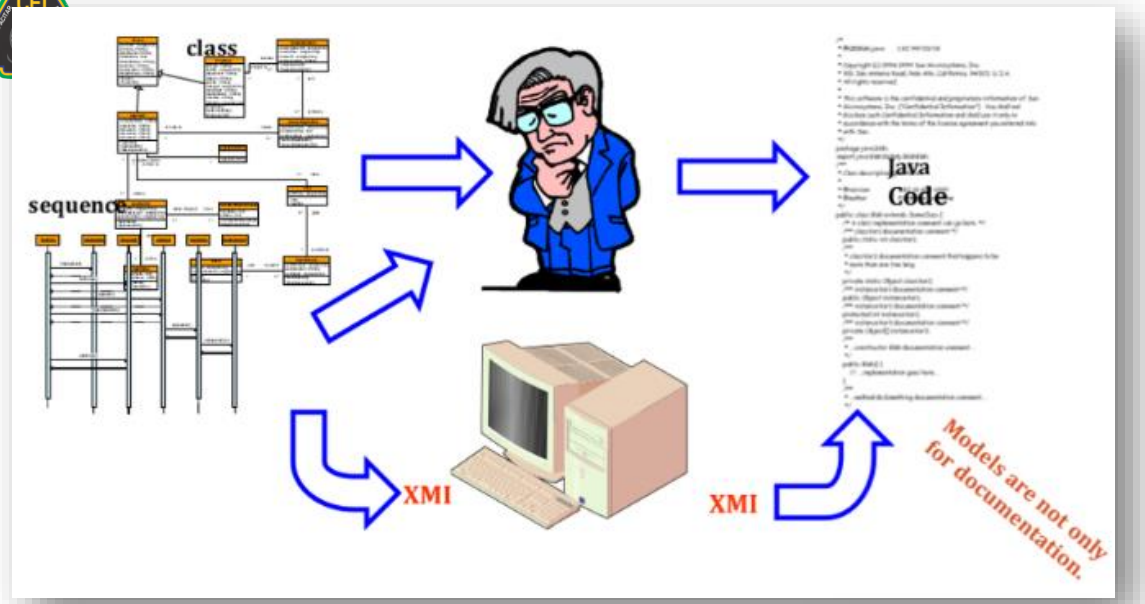


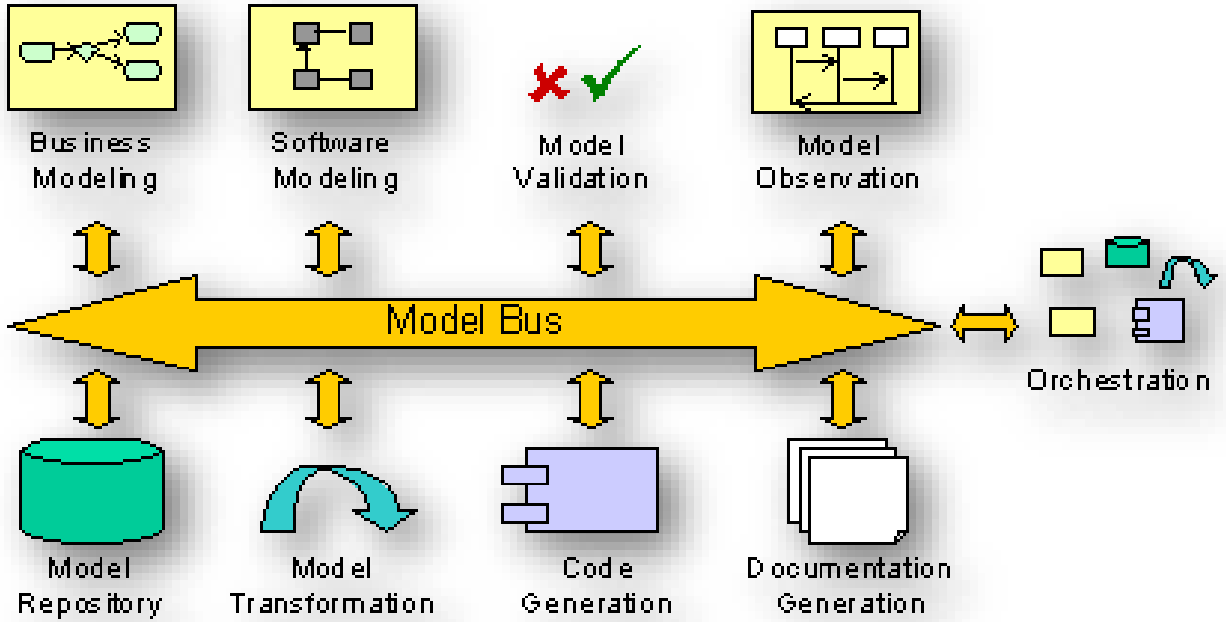
Model Transformation

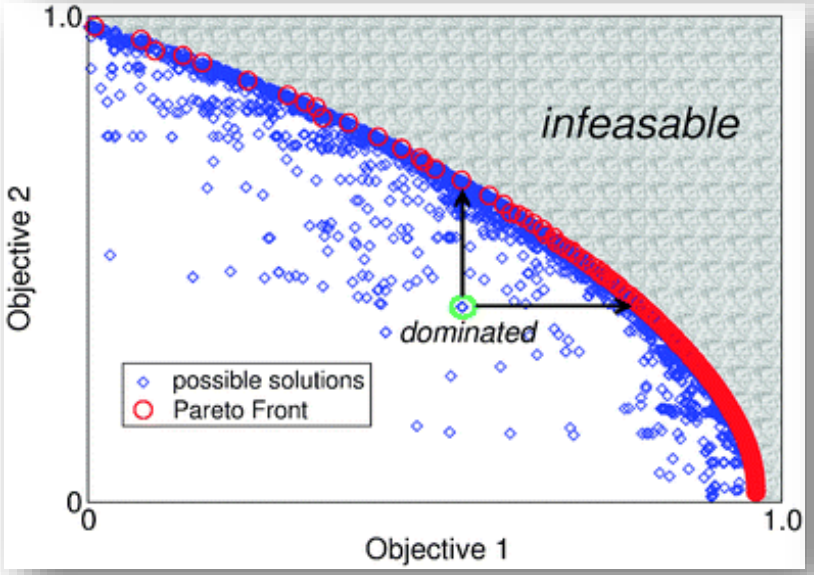


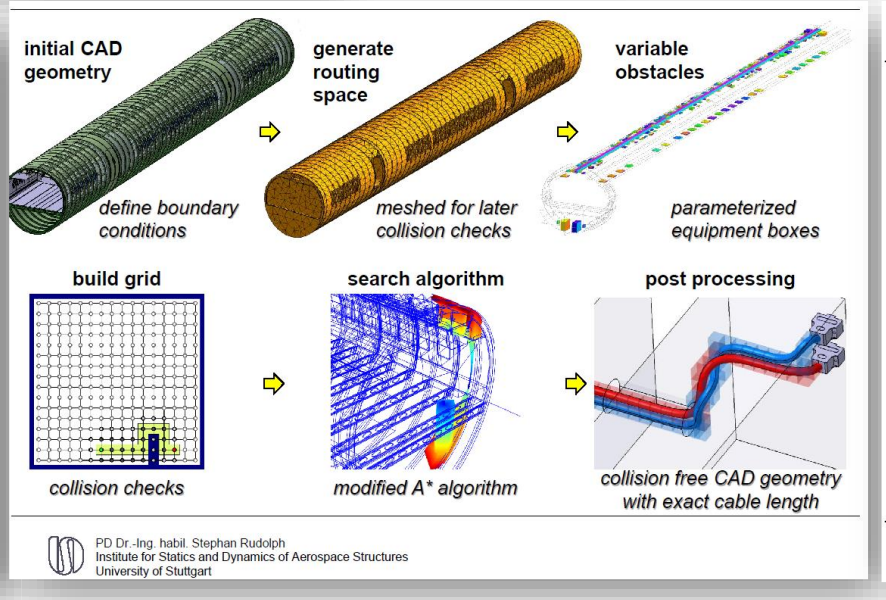


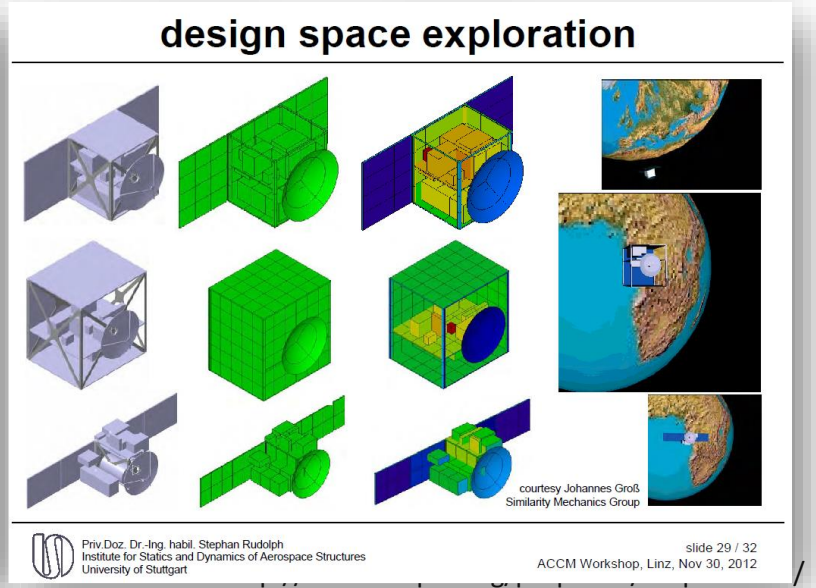
More benefits





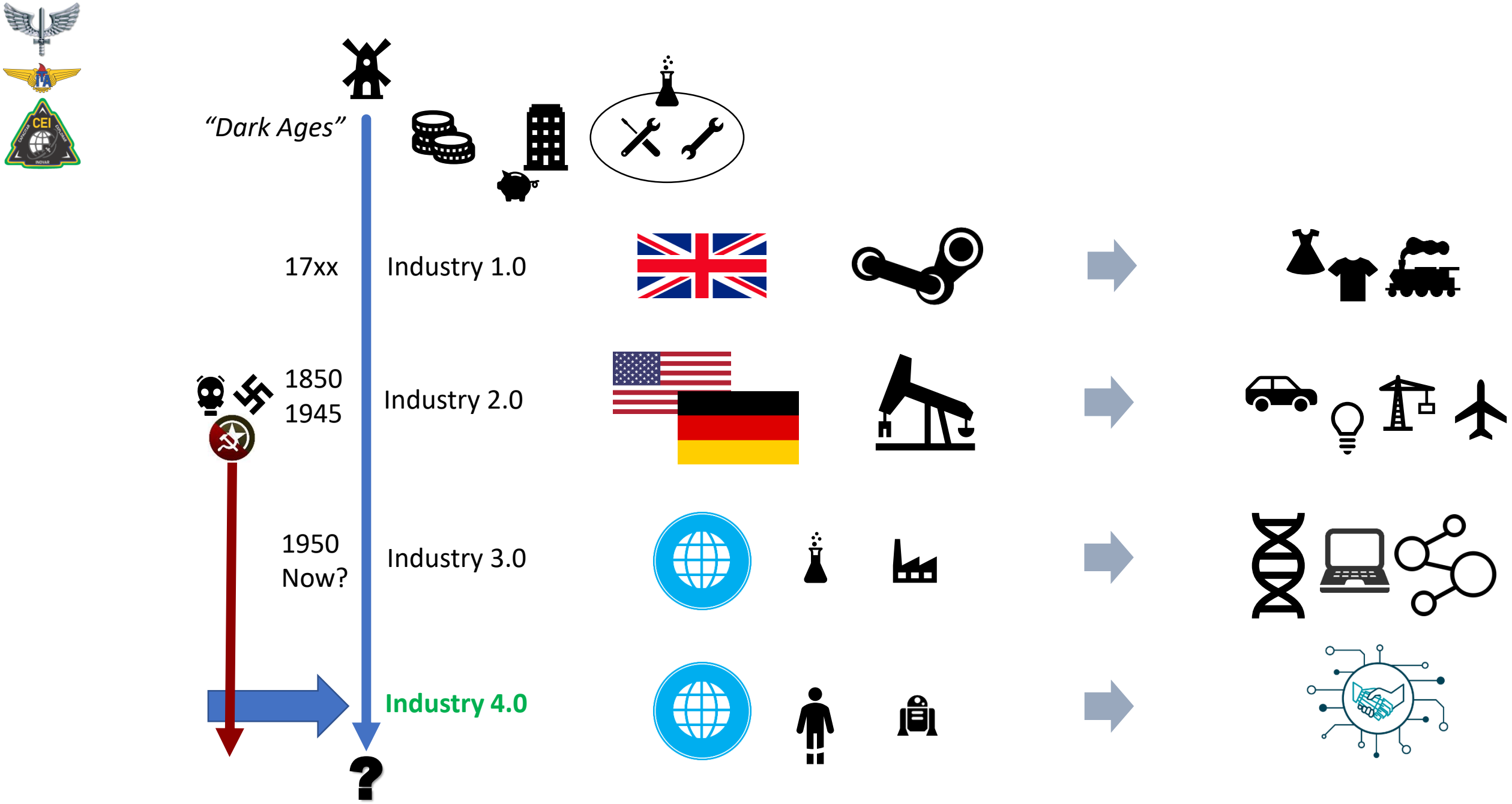




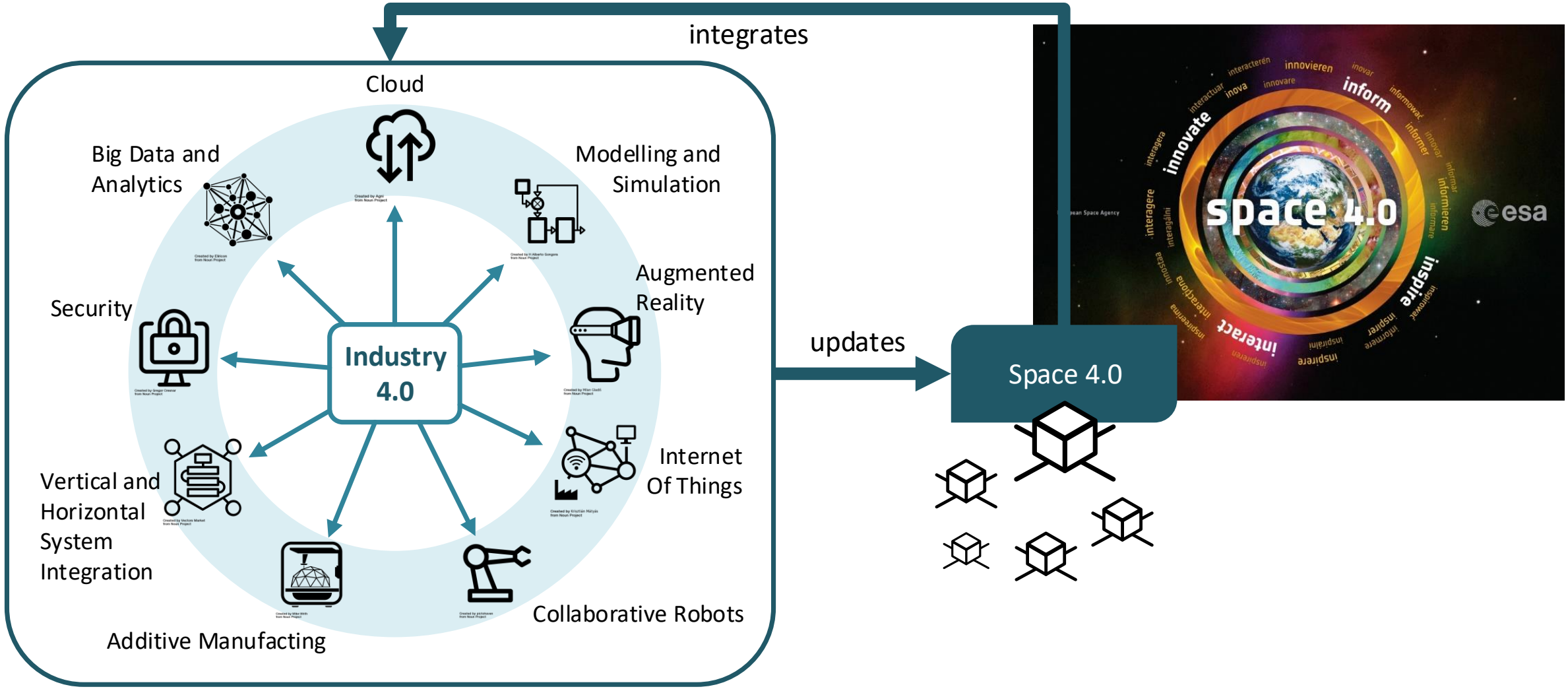


- THIS IS NOT RANDON...

- IT IS A CONSEQUENCE OF THE COMPUTATIONAL TECHNOLOGY BEING APPROPRIATED BY THE ENGINEERINGS


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

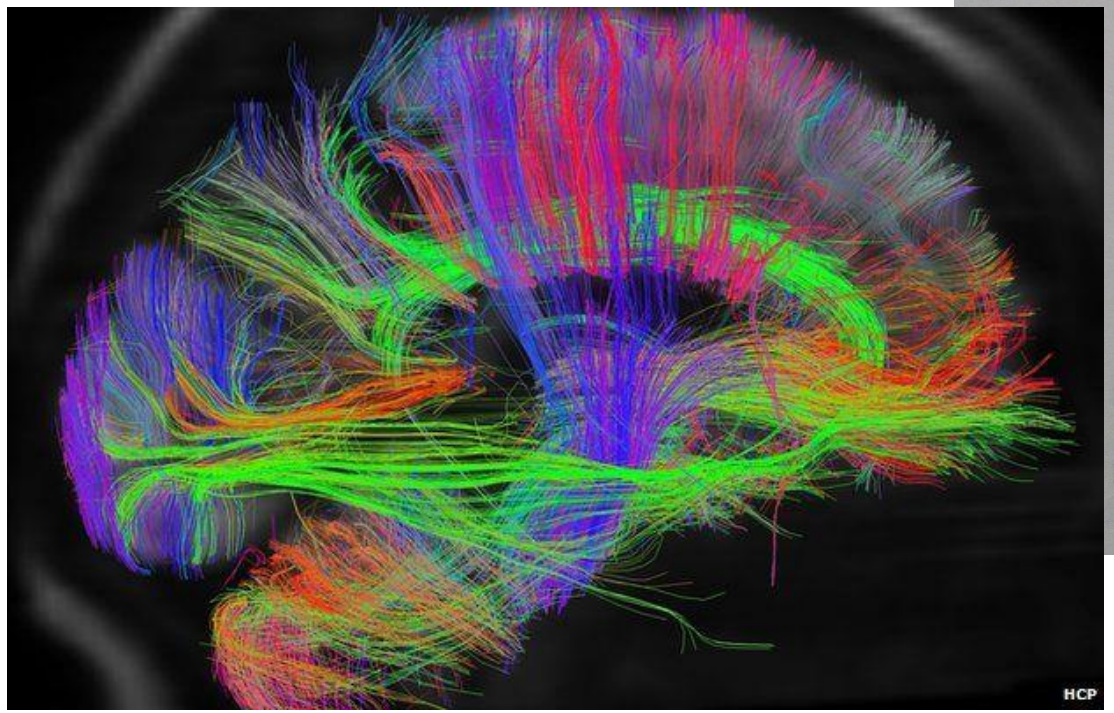
Future Motto: Creative Work Interconnection (INCOSE)



Language

Reducing to function, we are a machine that recognizes and connects model patterns…

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

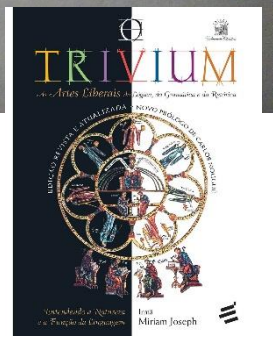


feita de tijolos, pedras ou de

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.

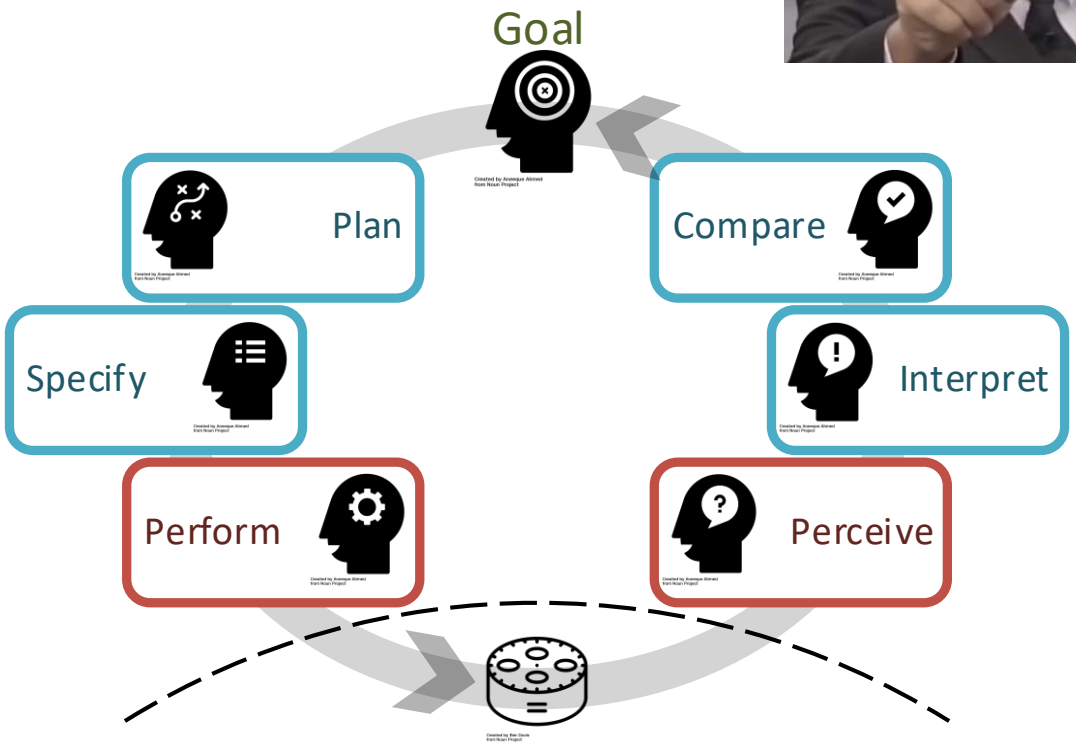
As regras para relacionar símbolos regem três operações gramaticais: substituir símbolos equivalentes, combinar símbolos e separar símbolos.

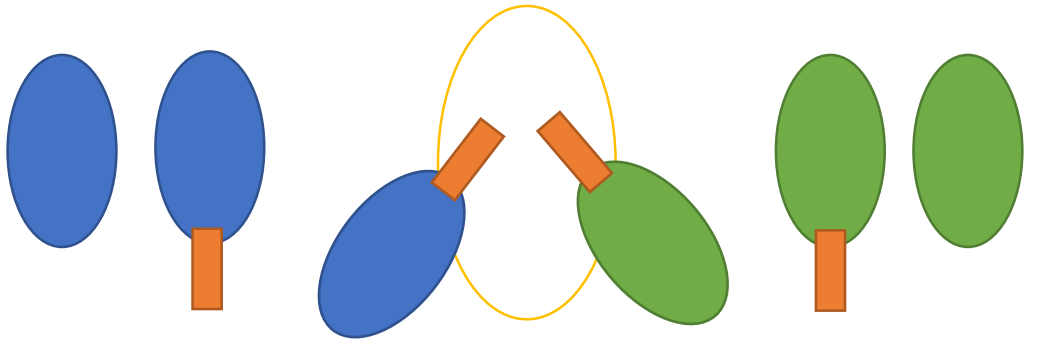


Cognition and Language

Cognition is "*the mental action or process of acquiring knowledge and understanding through thought, experience, and the senses*". It encompasses processes such as attention, the formation of knowledge, memory and working memory, judgment and evaluation, reasoning and "computation", problem solving and decision making, comprehension and production of language. Cognitive processes use existing knowledge and generate new knowledge.

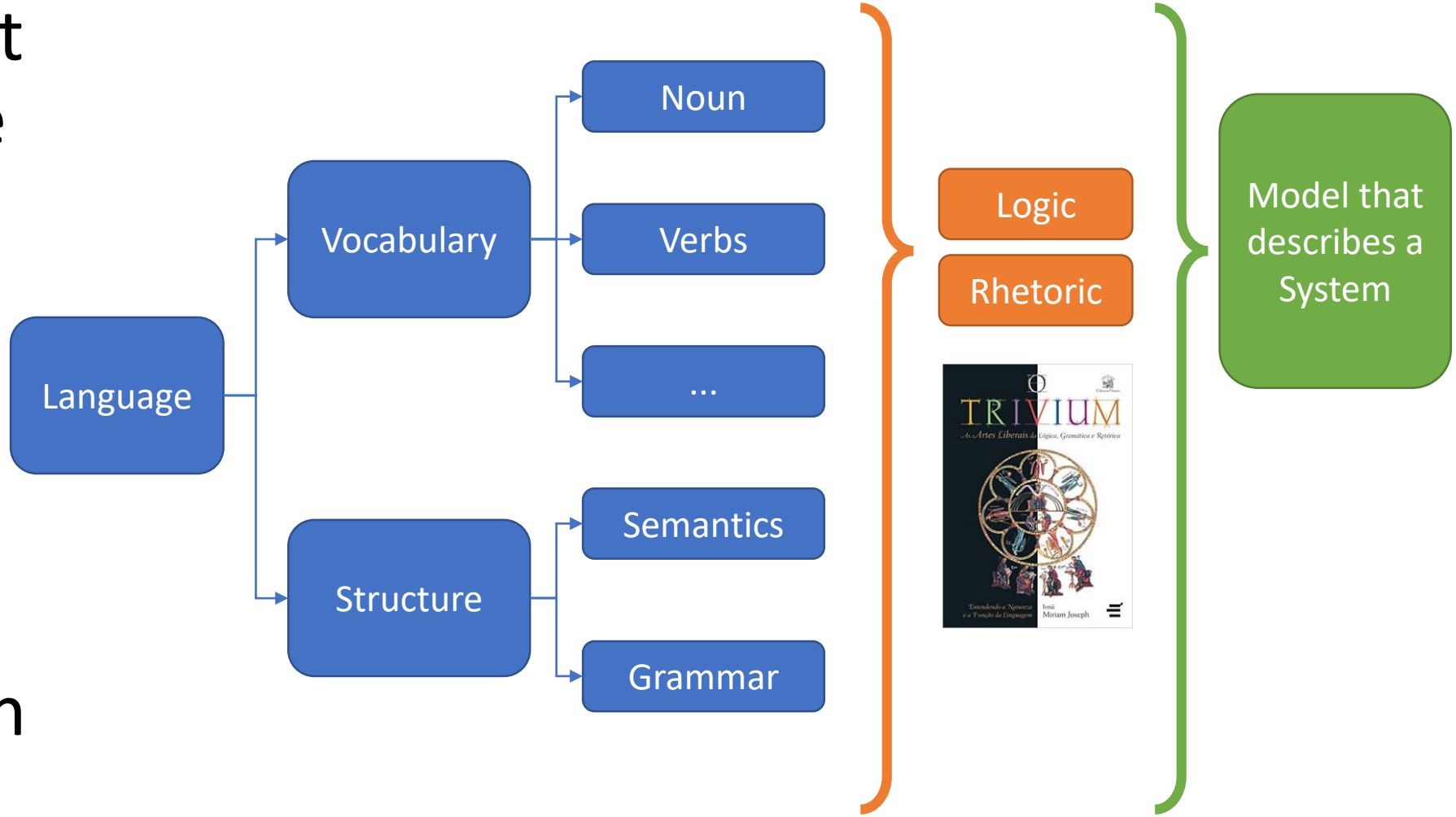
https://en.wikipedia.org/wiki/Cognition





Cognition and Language

is a system that consists of the development, acquisition, maintenance and use of complex systems of communication .



https://en.wikipedia.org/wiki/Language

Tools must map the knowledge in models within known languages. The user only handles already known “things”...

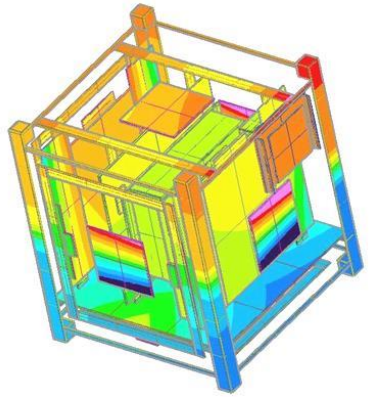
http://www.olavodecarvalho.org/apostilas/presenca.htm

6:15

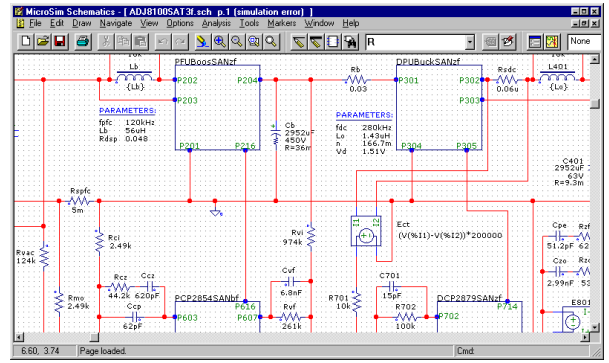
https://www.ted.com/talks/david_mccandless_the_beauty_of_data_visualization#

Every Language has its logical structure and semantics → Model

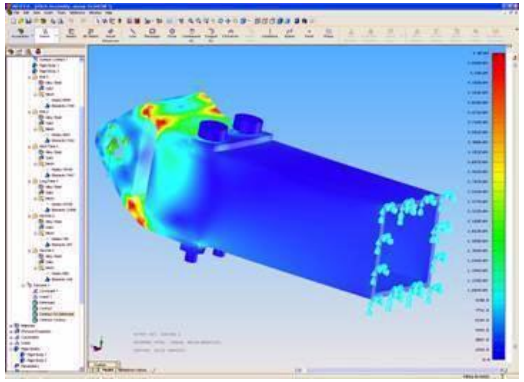
Thermal Eng.



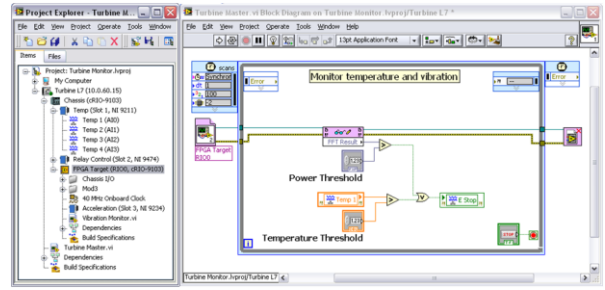
Electrical Eng.



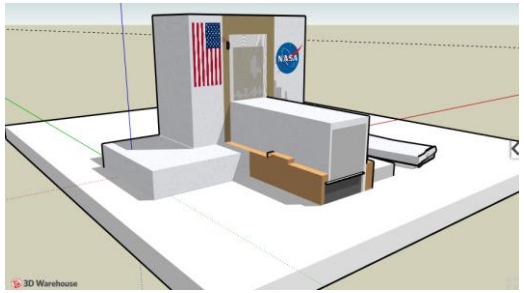
Mechanical Eng.



Control Eng.

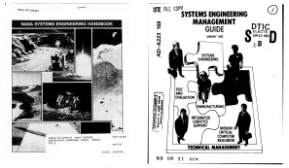


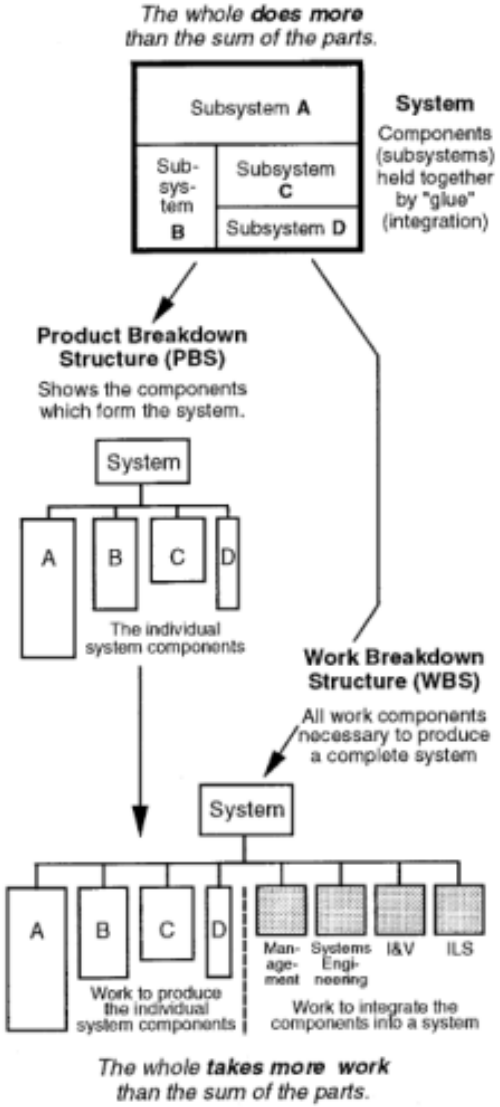
Infrastructure Eng.

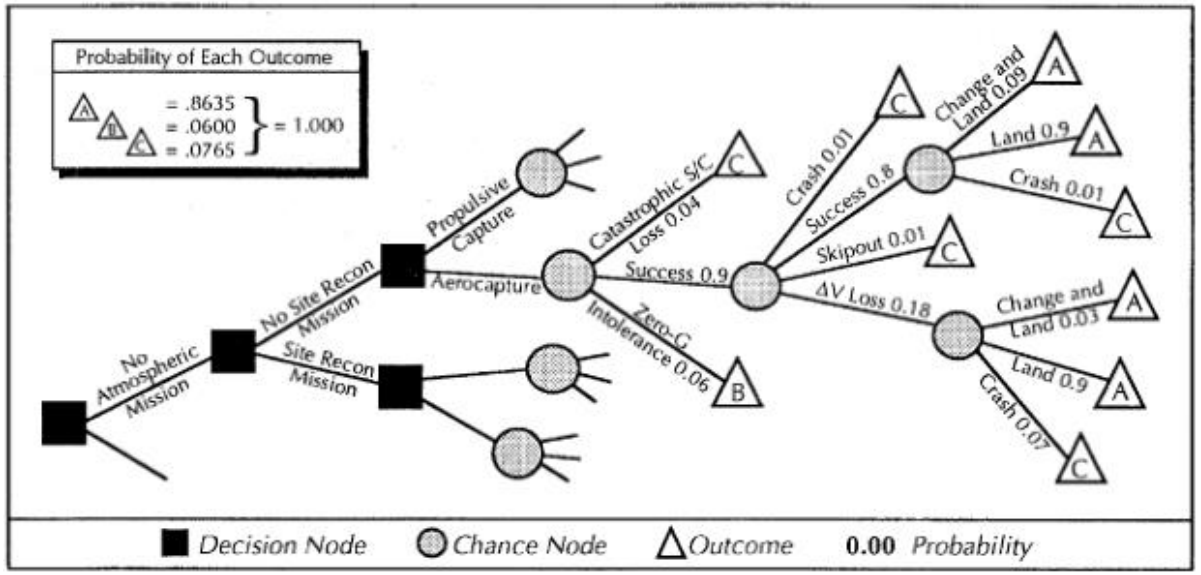


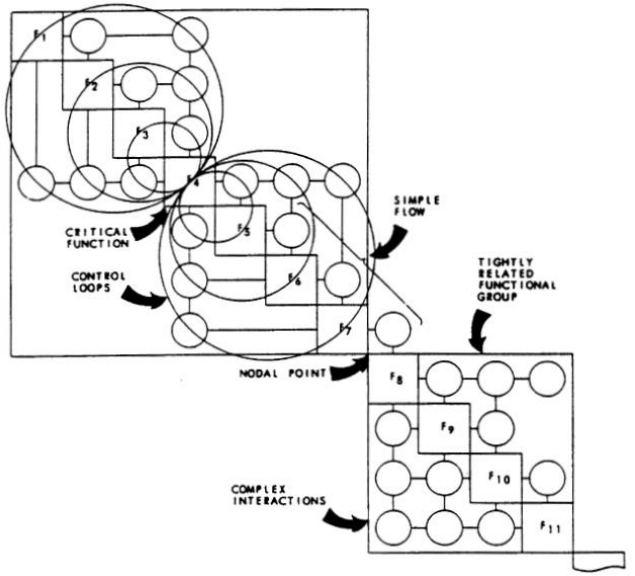
Systems Eng ???

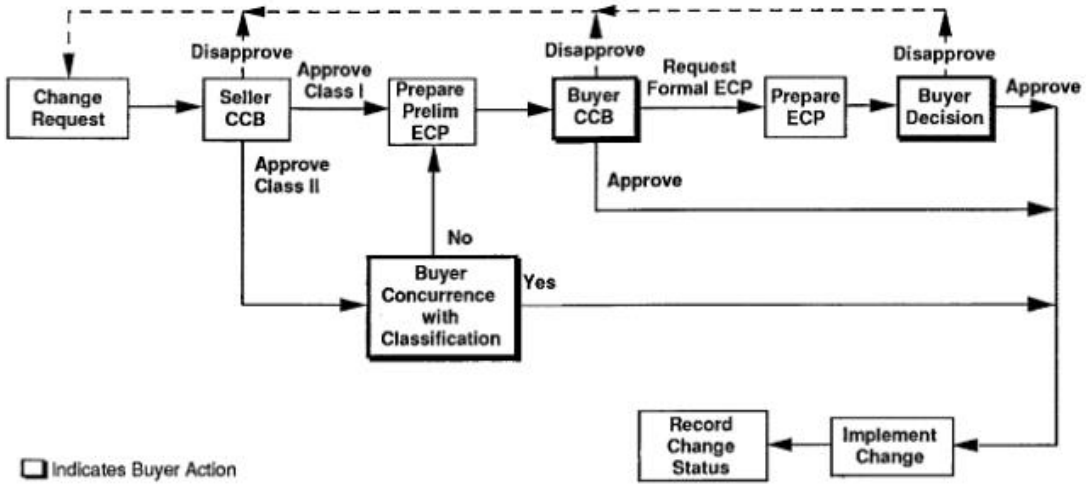
What is the SE Language?

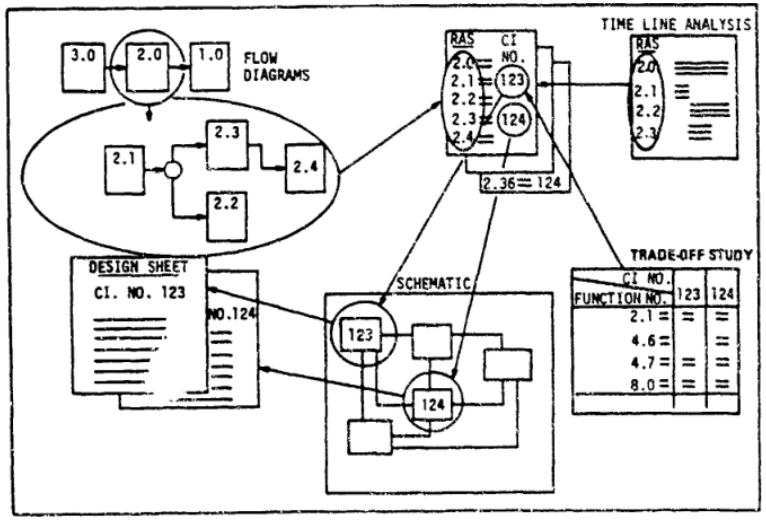




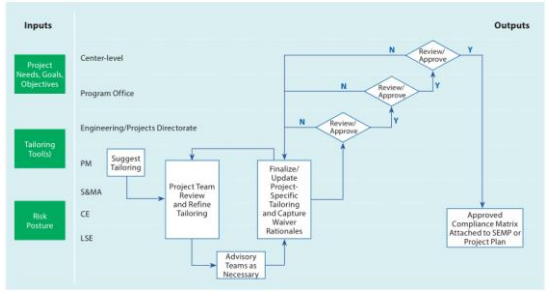


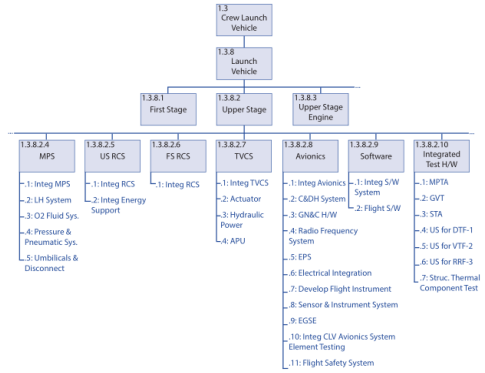


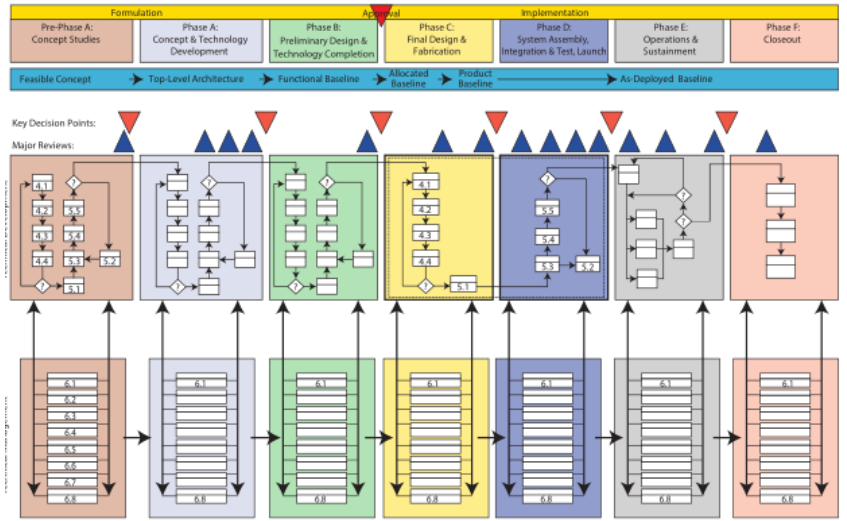


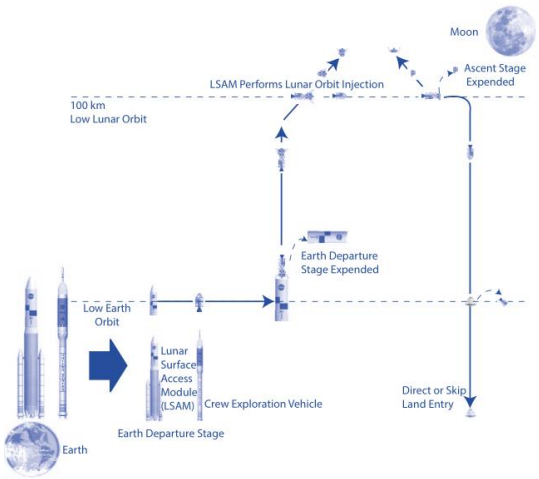


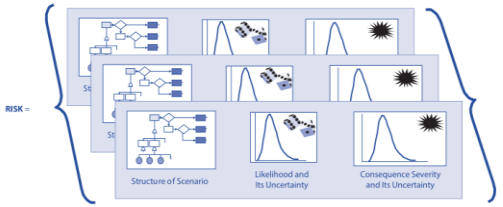
With time… Nicer diagrams











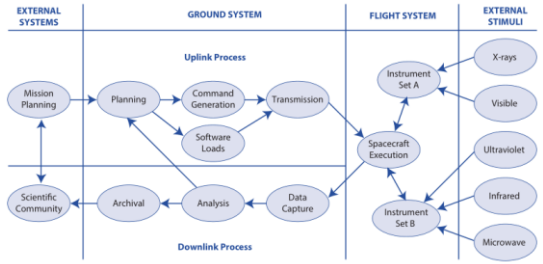


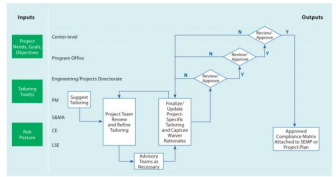
TABLE D-1 Requirements Verification Matrix

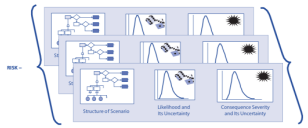
Require-ment No.	Document	Para-graph	Shall State-ment	Verifi-cation Success Criteria	Verifi-cation Method	Facility or Lab	Phase*	Accep-tance Require-ment?	Preflight Accep-tance?	Perform-ing Orga-nization	Results
Unique iden-tifier or each requirement	*Document number the requirement is contained within*	*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 per-form the ver-ification and validation.*	*Phase in which the verification and valida-tion will be performed.*	*Indicate whether this requirement is also ver-ified during initial accep-tance testing of each unit.*	*Indicate whether this requirement is also ver-ified during any pre-flight or recurring acceptance testing of each unit*	*Organization responsible for per-forming 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

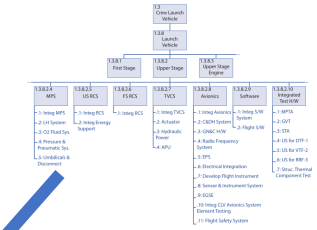
Red = Below TRL 3
Yellow = TRL 3, 4 & 5
Green = TRL 6 and above
White = Unknown
X = Exists

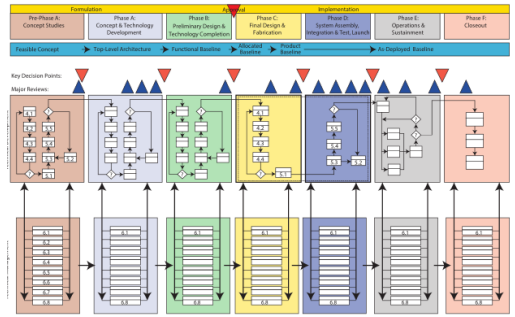
	Concept	Demonstration Units: Breadboard	Demonstration Units: Brassboard	Demonstration Units: Developmental Model	Demonstration Units: Prototype	Demonstration Units: Flight Qualified	Environment: Laboratory Environment	Environment: Relevant Environment	Environment: Space Environment	Environment: Space 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															

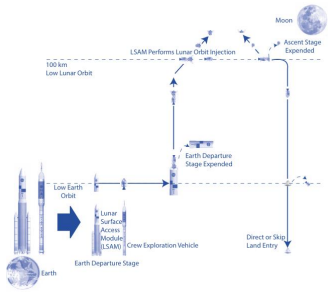
Model >>>>>>>>>>> Diagram

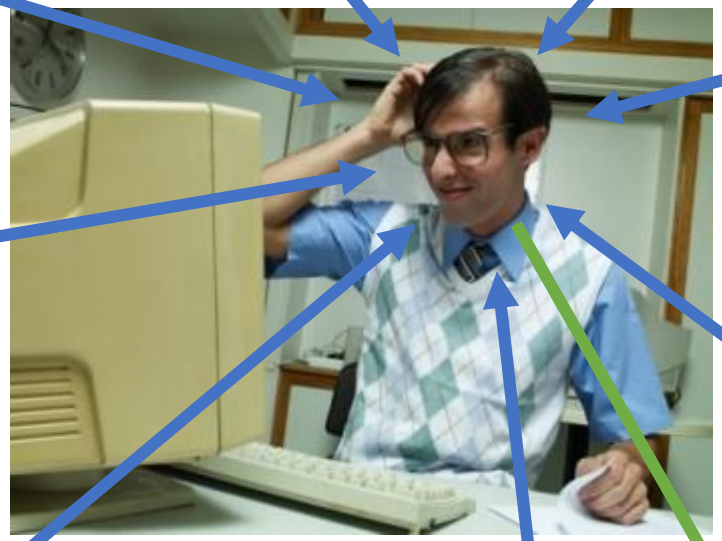


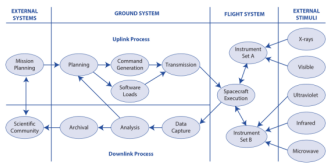


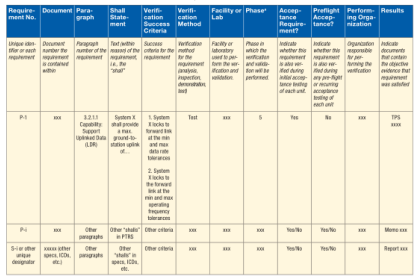


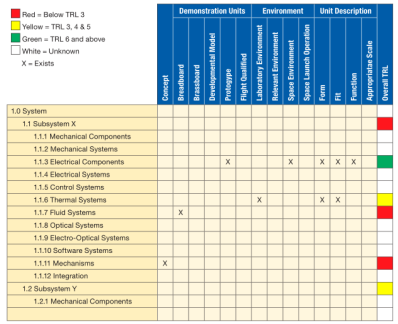












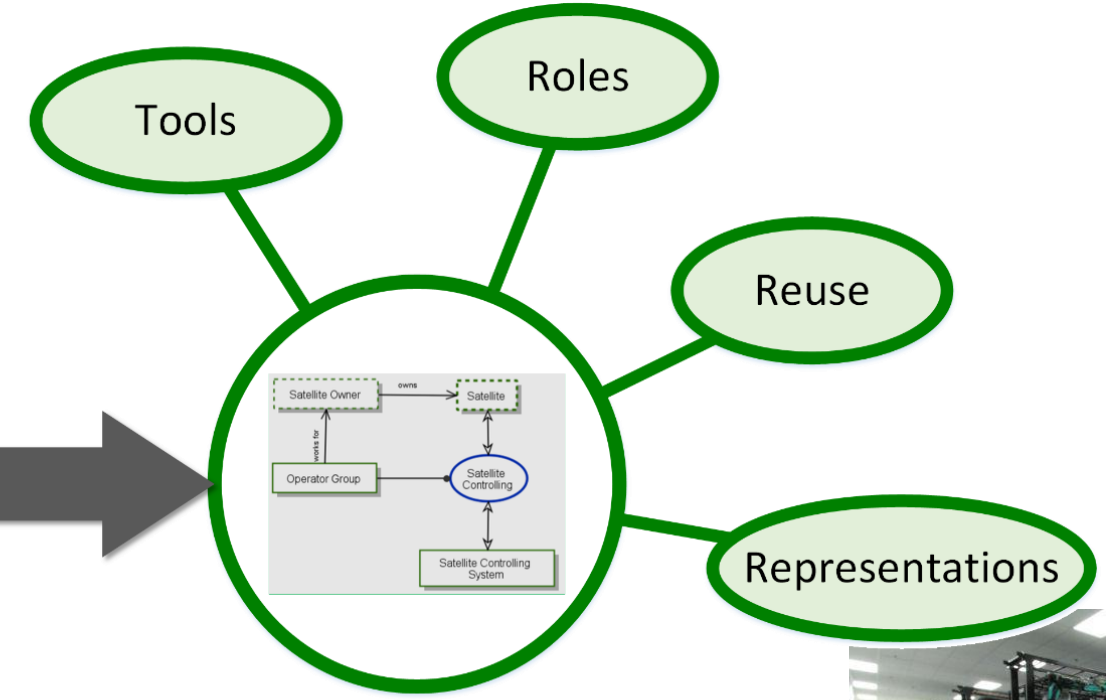
Integrator

Model “based” System Engineering

Paradigm Shift

Margaret Hamilton

Document Oriented



Model Oriented

MBSE

tenents, approach, process & toolbox

“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 is not universal, needs to 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 lists several methodologies

List of Methodologies and Methods

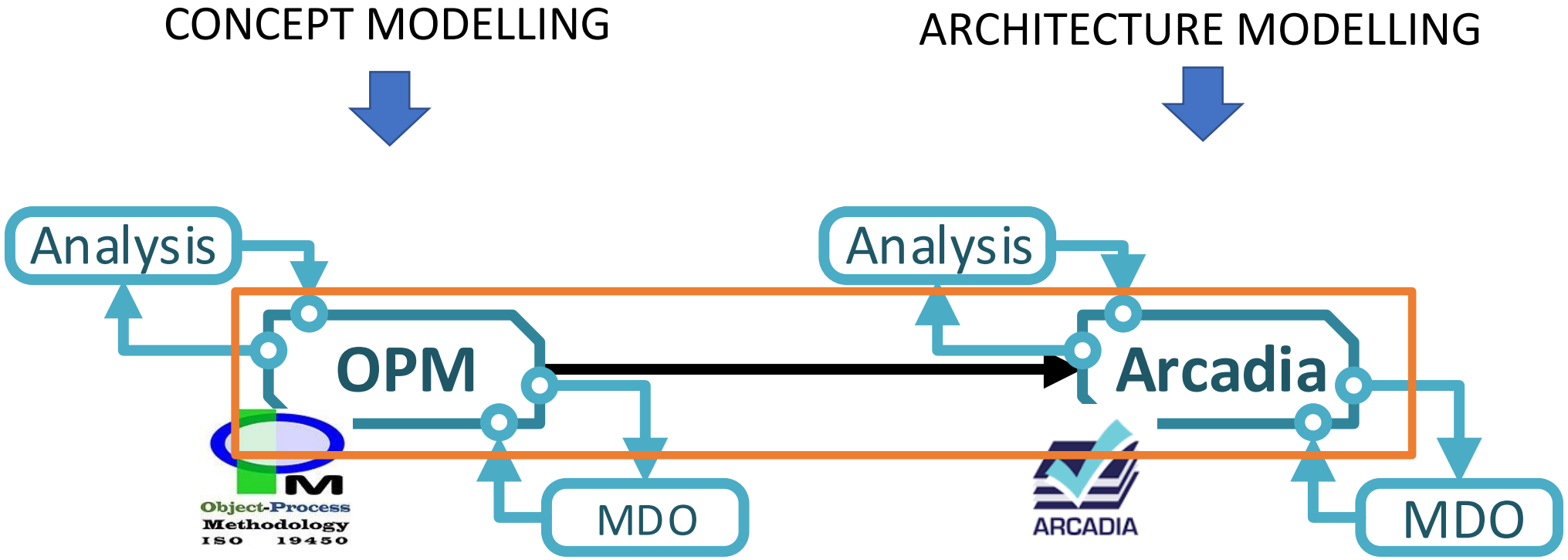
Methodologies Surveyed in INCOSE 2008 Report

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

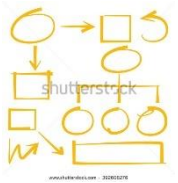
Our approach: OPM & Arcadia Hybrid



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…

14:34

The start:
These “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

The Start:

Convert — fragmented knowledge into explicit and integrated knowledge

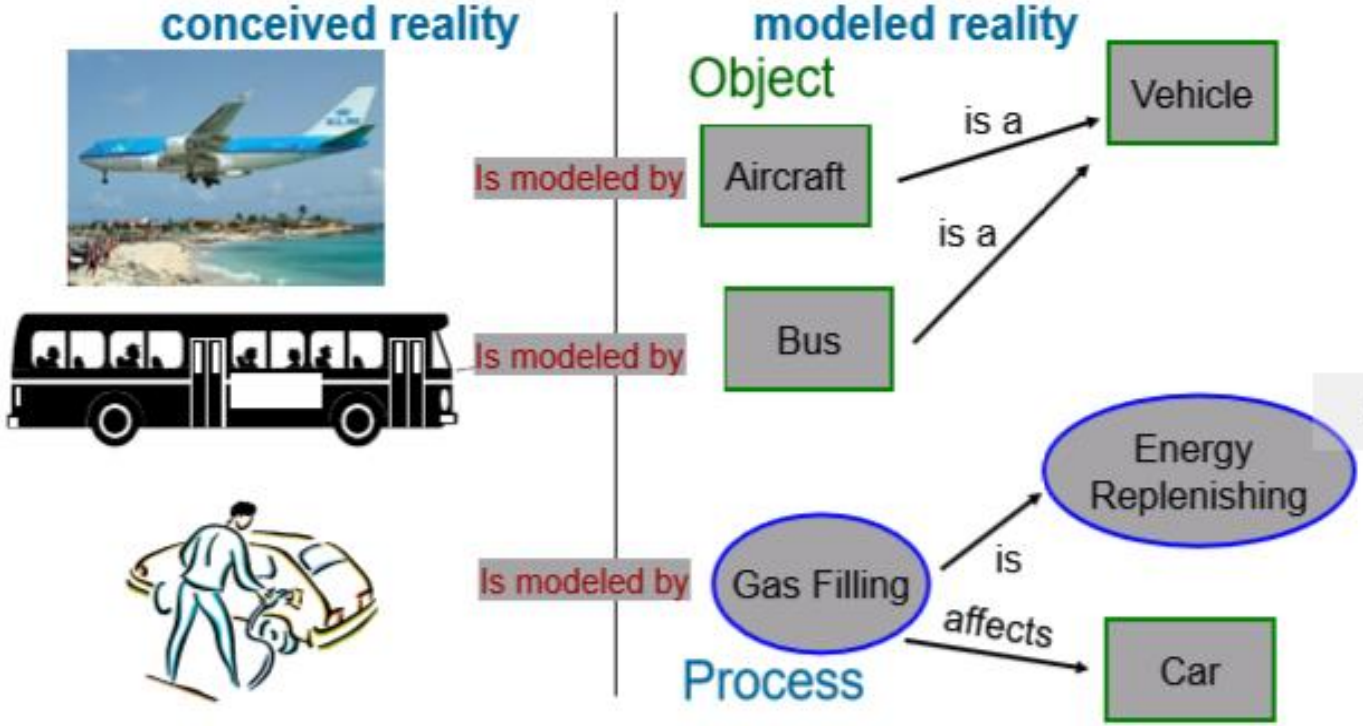
+

Construct — concise models

+

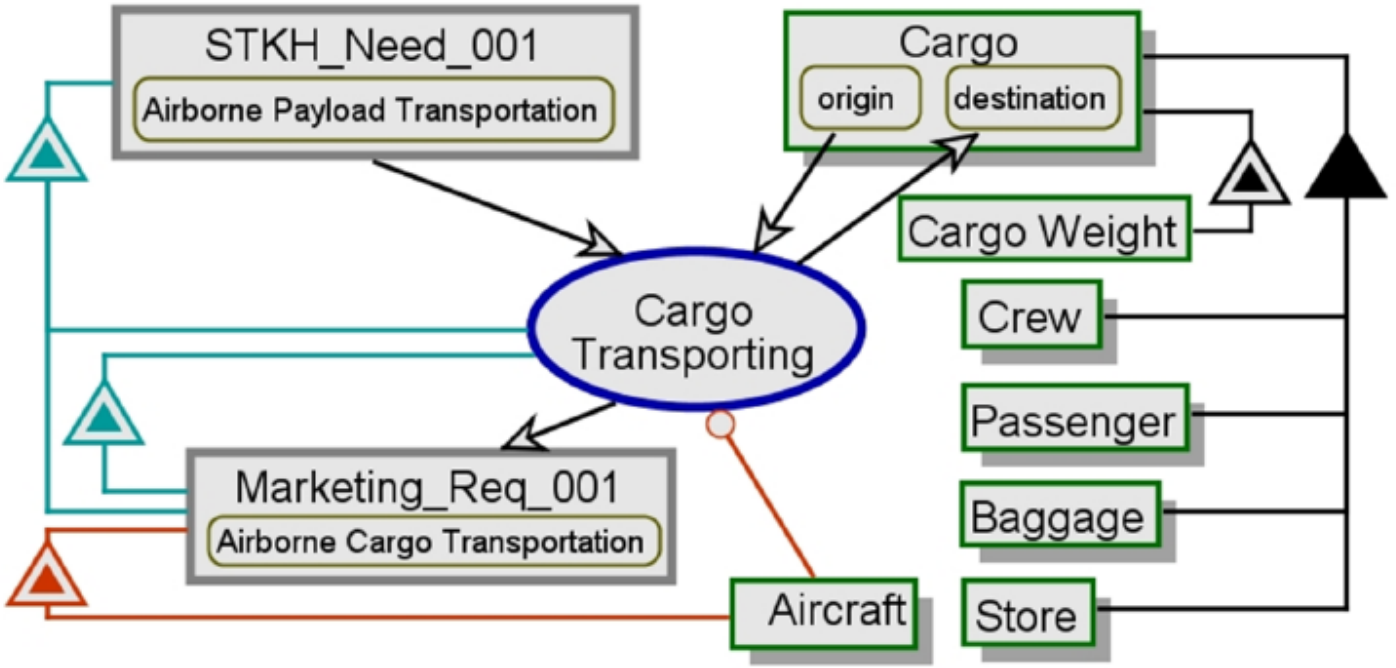
Communicate — The model to stakeholders

→ Concept Models



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

14:34



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

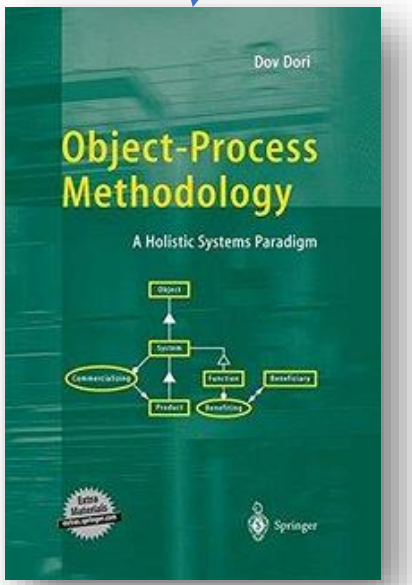
Model-Based Systems Engineering for Aircraft Design with Dynamic Landing Constraints Using Object-Process Methodology , AIAA Scitech 2019 Forum

OPM

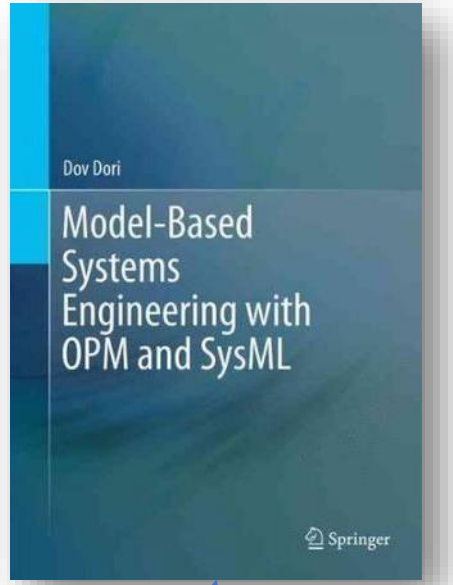
Prof. Dov Dori

Created in 2002

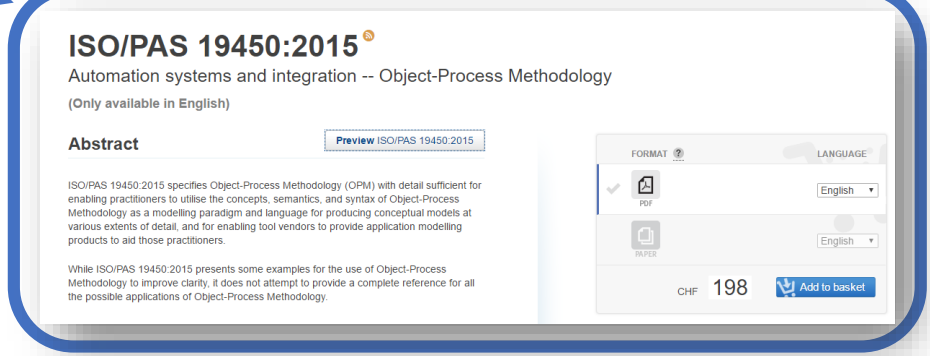
- **A** Single Diagram – Maps Behaviour and Structure
- **2 Building Blocks and 10 basic relations**
- Designed to “Systemic View” and “Concept Modelling”
- Simulation Ready



improving and showing it applicability

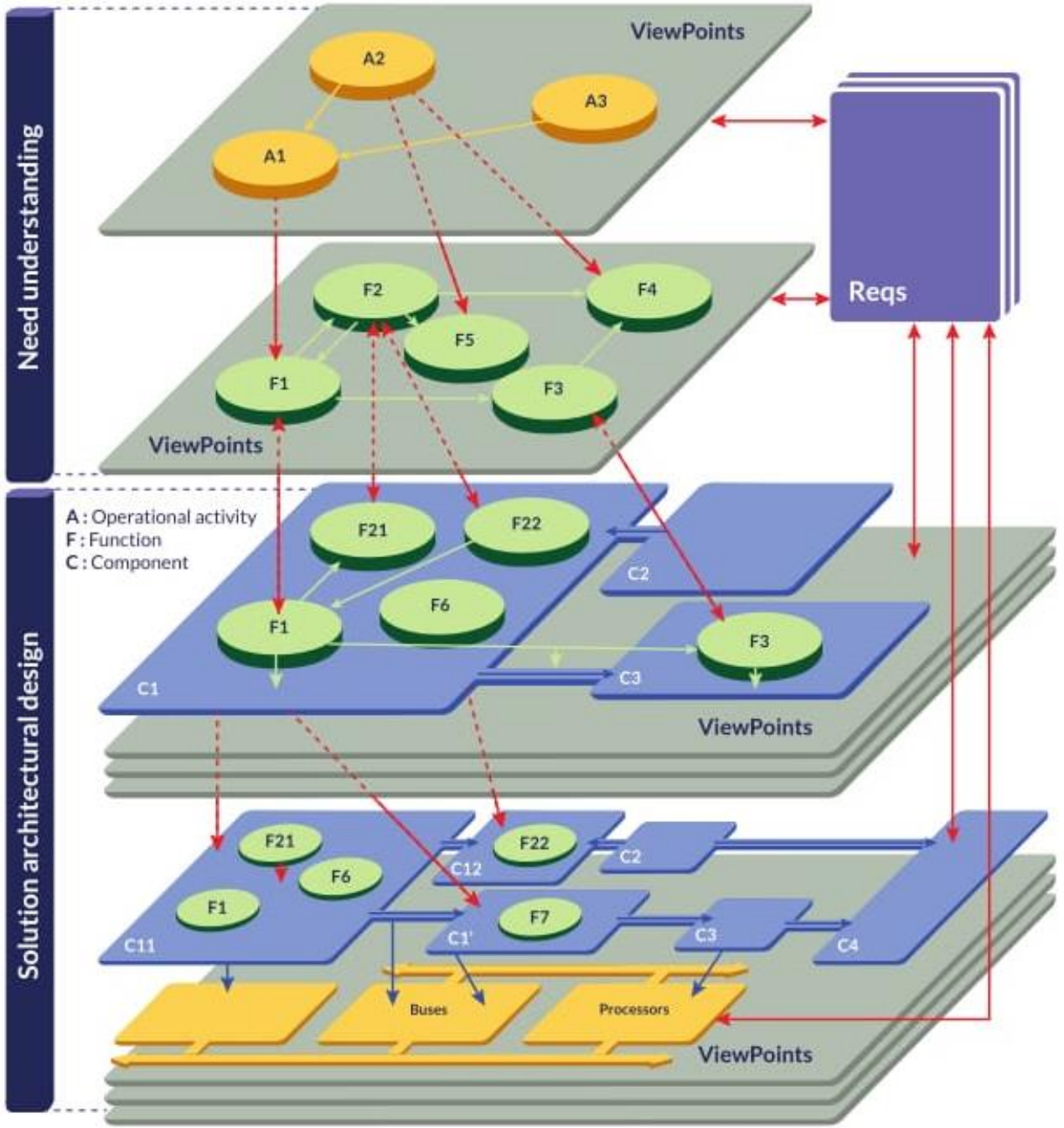


- ~130 Pages standard
- Published in late 2015
- Intended to “*Automation Systems and Integration*”
- Has the “**power**” of a ISO seal.



Evolving:
How can we construct the architecture?

- ✓Analyzing stakeholders' necessities
- ✓Analyzing the system functions that interact within the stakeholder’s necessities
- ✓Creating a logical decomposition
- ✓Creating a physical decomposition
- ✓Distributing to the supply chain

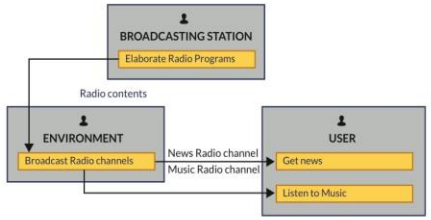
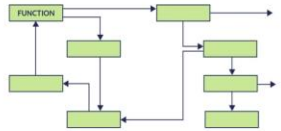
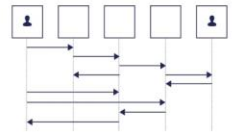
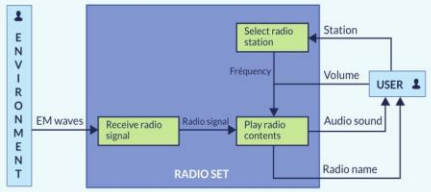
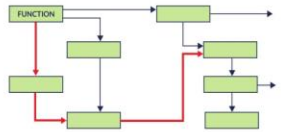
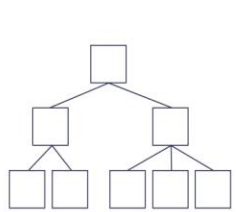
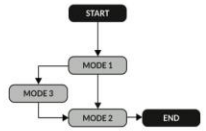
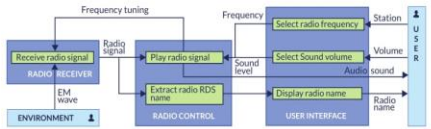
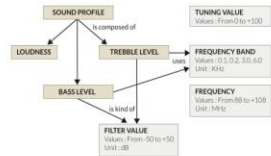
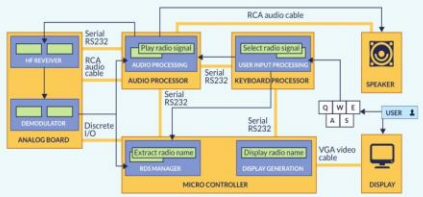
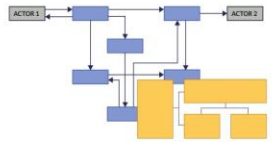
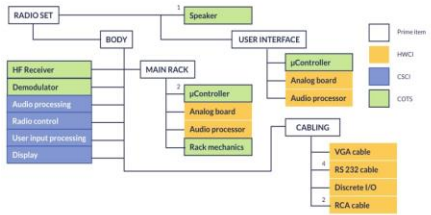
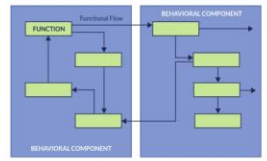


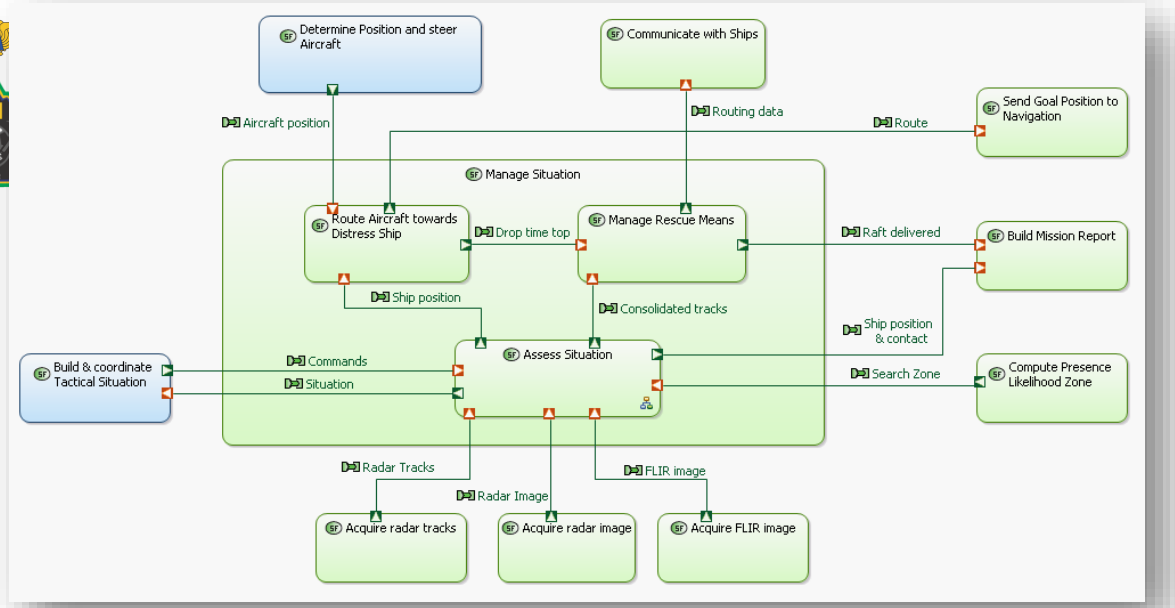
Operational Analysis
What the users of the system need to accomplish

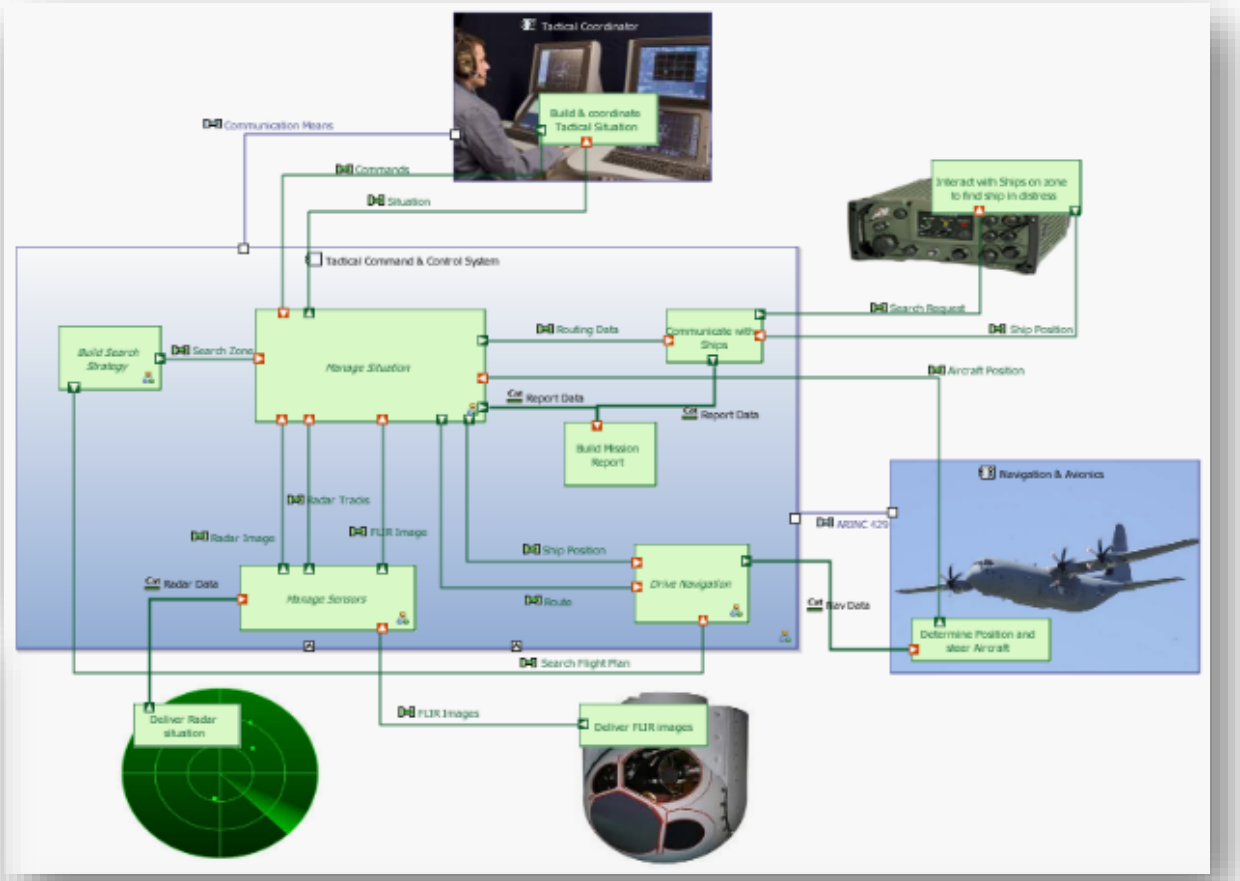
Functional & Non Functional Need
What the system has to accomplish for the users

Logical Architecture
How the system will work to fulfill expectations

Physical Architecture
How the system will be developed and built

METHOD STEPS	TASKS	SAMPLE MODEL	CONCEPTS	DESCRIPTION MEANS
NEED				
Customer Operational Need Analysis What the users of the system need to accomplish	✓ Define operational capabilities ✓ Perform an operational need analysis		- Operational capabilities - Actors, operational entities - Actor activities - Interactions between activities & actors - Information used in activities & interactions - Operational processes chaining activities - Scenarios for dynamic behaviour	 **Dataflow:** functions, op. activities interactions & exchanges **Scenarios:** actors, system, components interactions & exchanges 
System/ SW/HW Need Analysis What the system has to accomplish for the Users	✓ Perform a capability trade-off analysis ✓ Perform a functional and non-functional analysis ✓ Formalise and consolidate requirements		- Actors and system, capabilities - Functions of system & actors - Dataflow exchanges between functions - Functional chains traversing dataflow - Information used in functions & exchanges, data model - Scenarios for dynamic behaviour - Modes & states	 **Functional chains, operational processes** through functions & op. activities   **Modes & states** of actors, system, components **Breakdown** of functions & components
SOLUTIONS				
Logical Architecture Design How the system will work so as to fulfil expectations	✓ Define architecture drivers and viewpoints ✓ Build candidate architectural breakdowns in components ✓ Select best compromise architecture		*SAME CONCEPTS, PLUS :* - Components - Component ports and interfaces - Exchanges between components - Function allocation to components - Component interface justification by functional exchanges allocation	**Data model:** dataflow & scenario contents, definition & justification of interfaces 
Physical Architecture Design How the system will be developed & built	✓ Define architectural patterns ✓ Consider reuse of existing assets design a physical ✓ Design a physical reference architecture ✓ Validate and check it		*SAME CONCEPTS, PLUS :* - Behavioural components refining logical ones, and implementing functional behaviour - Implementation components supplying resources for behavioural components - Physical links between implementation components	 **Component wiring:** all kinds of components
Development Contracts What is expected from each designer/ sub-contractor	✓ Define a components IVVQ strategy ✓ Define & enforce a PBS and component integration contract		- Configuration items tree - Parts numbers, quantities - Development contract (expected behaviour, interfaces, scenarios, resource consumption, non-functional properties…)	**Allocation** of op.activities to actors, of functions to components, of behav.components to impl.components, of dataflows to interfaces, of elements to configuration items 


Determine Position and steer Aircraft
Communicate with Ships
Send Goal Position to Navigation
Aircraft position
Routing data
Route
Manage Situation
Route Aircraft towards Distress Ship
Drop time top
Manage Rescue Means
Raft delivered
Build Mission Report
Ship position
Consolidated tracks
Ship position & contact
Build & coordinate Tactical Situation
Commands
Situation
Assess Situation
Search Zone
Compute Presence Likelihood Zone
Radar Tracks
FLIR image
Radar Image
Acquire radar tracks
Acquire radar image
Acquire FLIR image


Tactical Coordinator
Build & coordinate Tactical Situation
Communication Means
Commands
Situation
Tactical Command & Control System
Interact with Ships on zone to find ship in distress
Search Request
Ship Position
Build Search Strategy
Search Zone
Manage Situation
Routing Data
Communicate with Ships
Aircraft Position
Report Data
Build Mission Report
Navigation & Avionics
Radar Tracks
Radar Image
FLIR Image
ARINC 429
Ship Position
Manage Sensors
Drive Navigation
Route
Nav Data
Determine Position and steer Aircraft
Search Flight Plan
Deliver Radar situation
FLIR Images
Deliver FLIR images

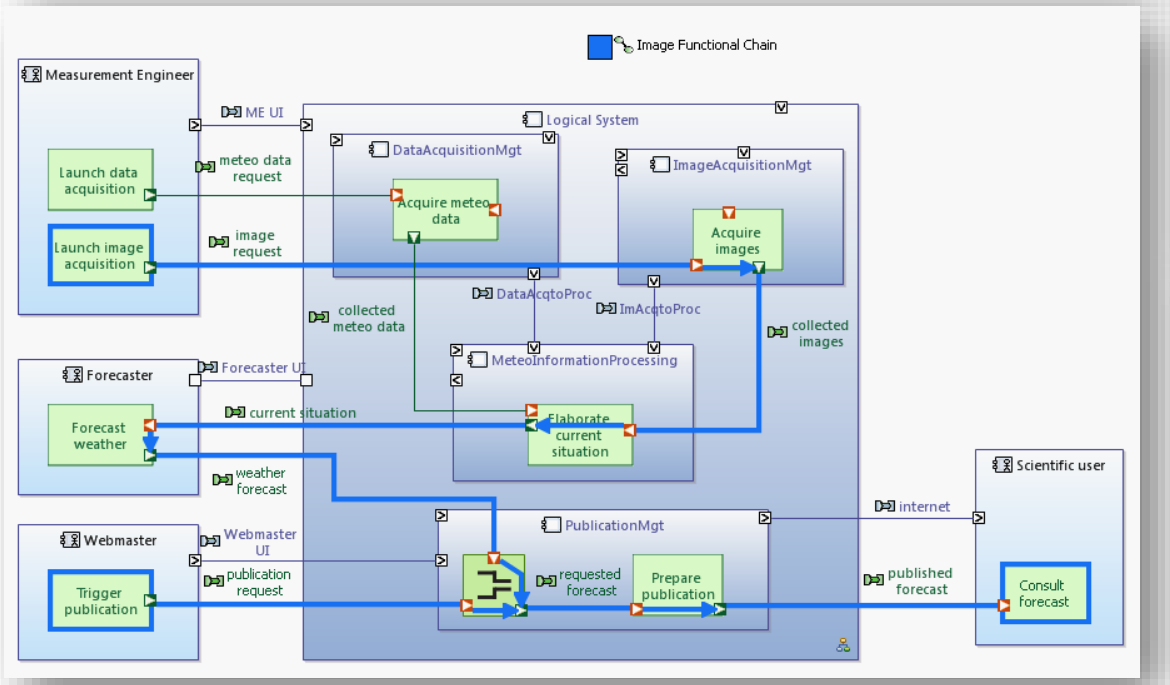
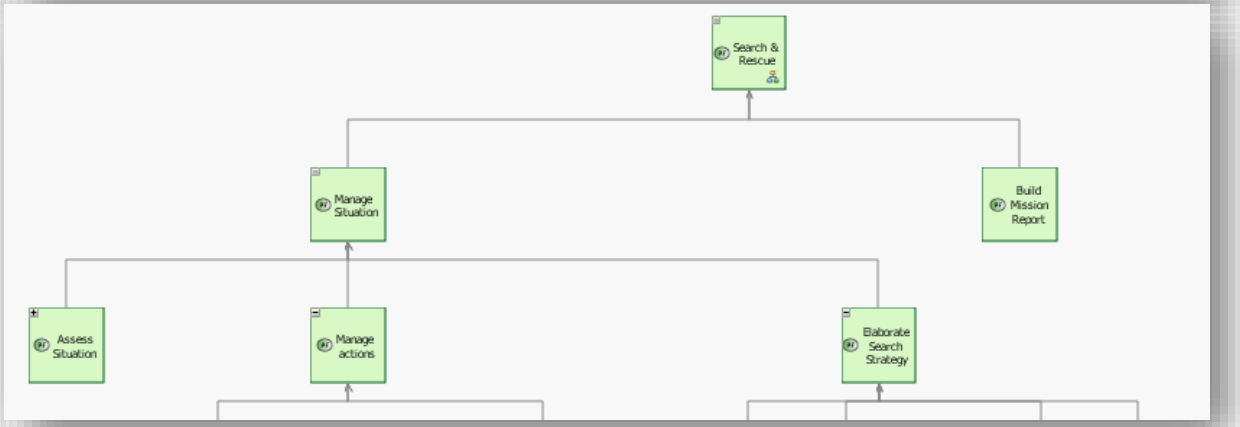
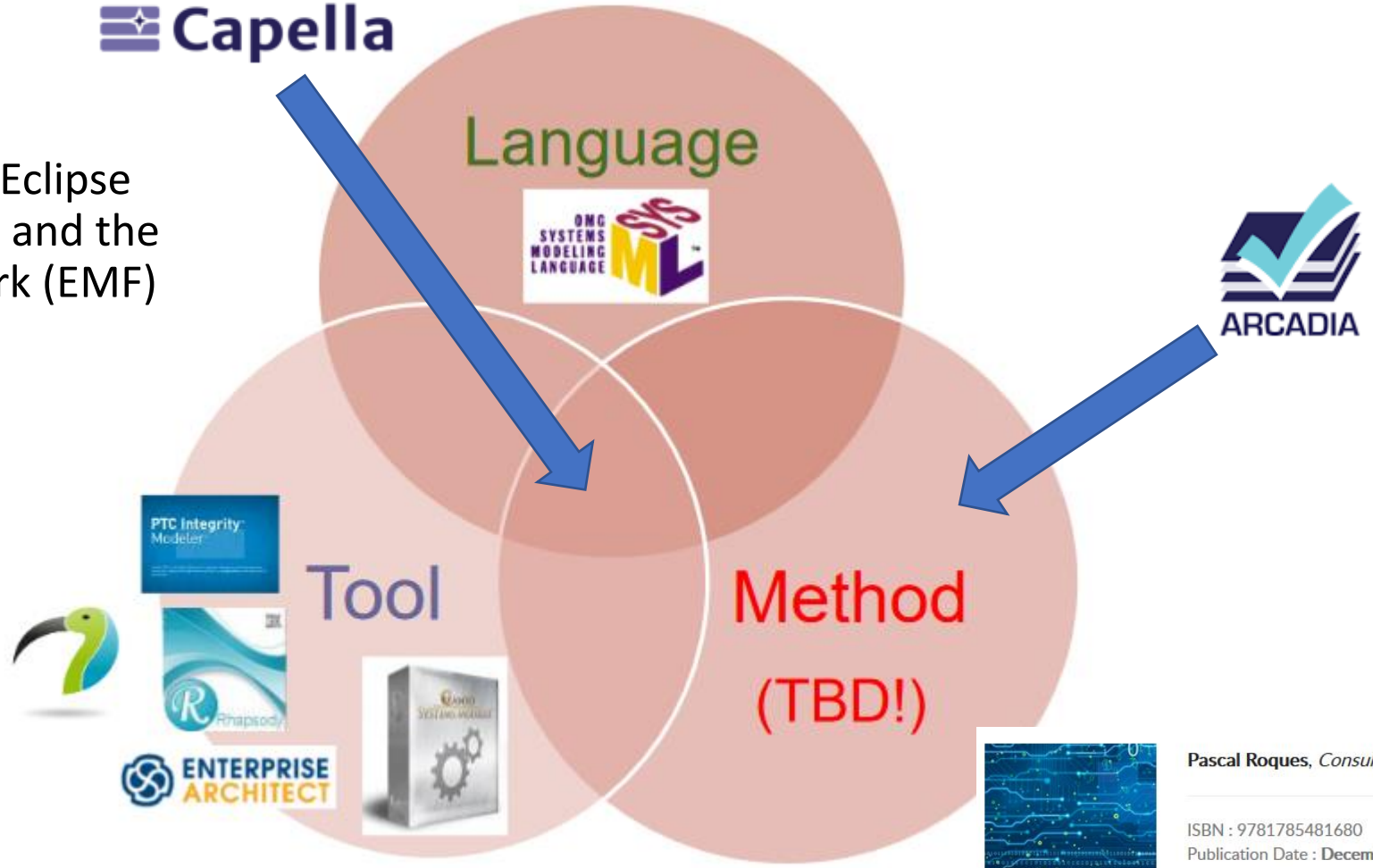


Image Functional Chain
Measurement Engineer
ME UI
Logical System
DataAcquisitionMgt
ImageAcquisitionMgt
Launch data acquisition
meteo data request
Acquire meteo data
Launch image acquisition
image request
Acquire images
DataAcqtoProc
ImAcqtoProc
collected meteo data
collected images
MeteoInformationProcessing
Forecaster
Forecaster UI
Forecast weather
current situation
Elaborate current situation
weather forecast
Webmaster
Webmaster UI
PublicationMgt
internet
Scientific user
Trigger publication
publication request
requested forecast
Prepare publication
published forecast
Consult forecast


Search & Rescue
Manage Situation
Build Mission Report
Assess Situation
Manage actions
Elaborate Search Strategy

Arcadia / Capella

- **Polarsys Group**
  - Thales Alenia
- Phases **0 - C**
- Capella is a layer above the Eclipse IDE, created using the Sirius and the Eclipse Modelling Framework (EMF)
- Diagrams:
  - operational architecture,
  - use cases,
  - dataflow,
  - architecture,
  - logical,
  - physical,
  - tree,
  - sequence,
  - state,
  - and classes



Pascal Roques, *Consultant*

ISBN : 9781785481680
Publication Date : **December 2017**
Hardcover **292 pp**
130.00 USD

https://www.polarsys.org/capella/

2:34

End of LEC-001 Class

Homework AGN-001: http://www.omgwiki.org/MBSE/doku.php?id=mbse:methodology

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~~

Each group must make a [tbd] minutes presentation/discussion about one methodology.

As this course will cover OPM and Arcadia, those are out of the list (☺)

Homework PRD-001: CASE RESEARCH

- Initial understanding statement report.
- Free report, reviewing the initial understanding of the system to be modelled.

- [tbd] minutes presentation to each group.

- Suggestion: context, reasons, needs, stakeholders, studies, motivations, etc…