July 3rd, 2025, Delft

Model-Based Systems Engineering Tutorial

ACM SIGSOFT School Software Engineering for Robotics

Carlos Hernandez Corbato

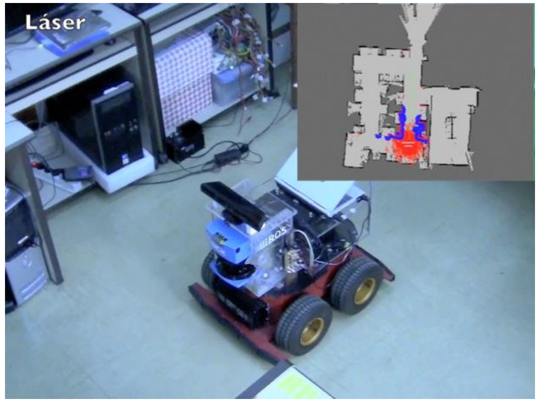
c.h.corbato@tudelft.nl

KAS Lab | R^2S | Cognitive Robotics

TU Delft Mechanical Engineering

Cognitive Robotics

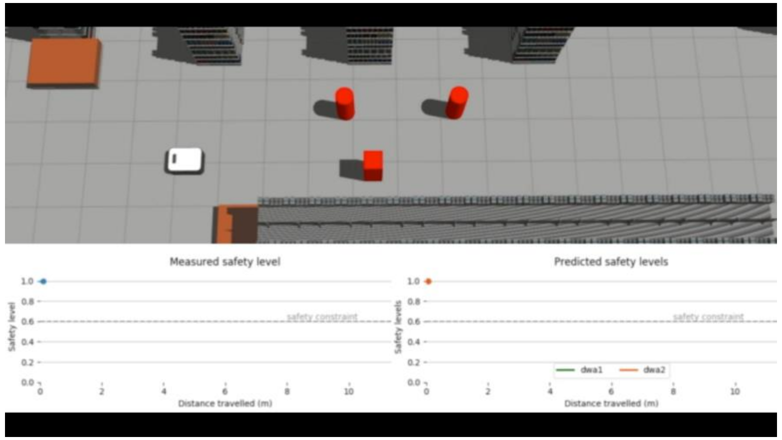
Personal story getting into MBSE: self-engineering robots



2013

2018

2019



2021

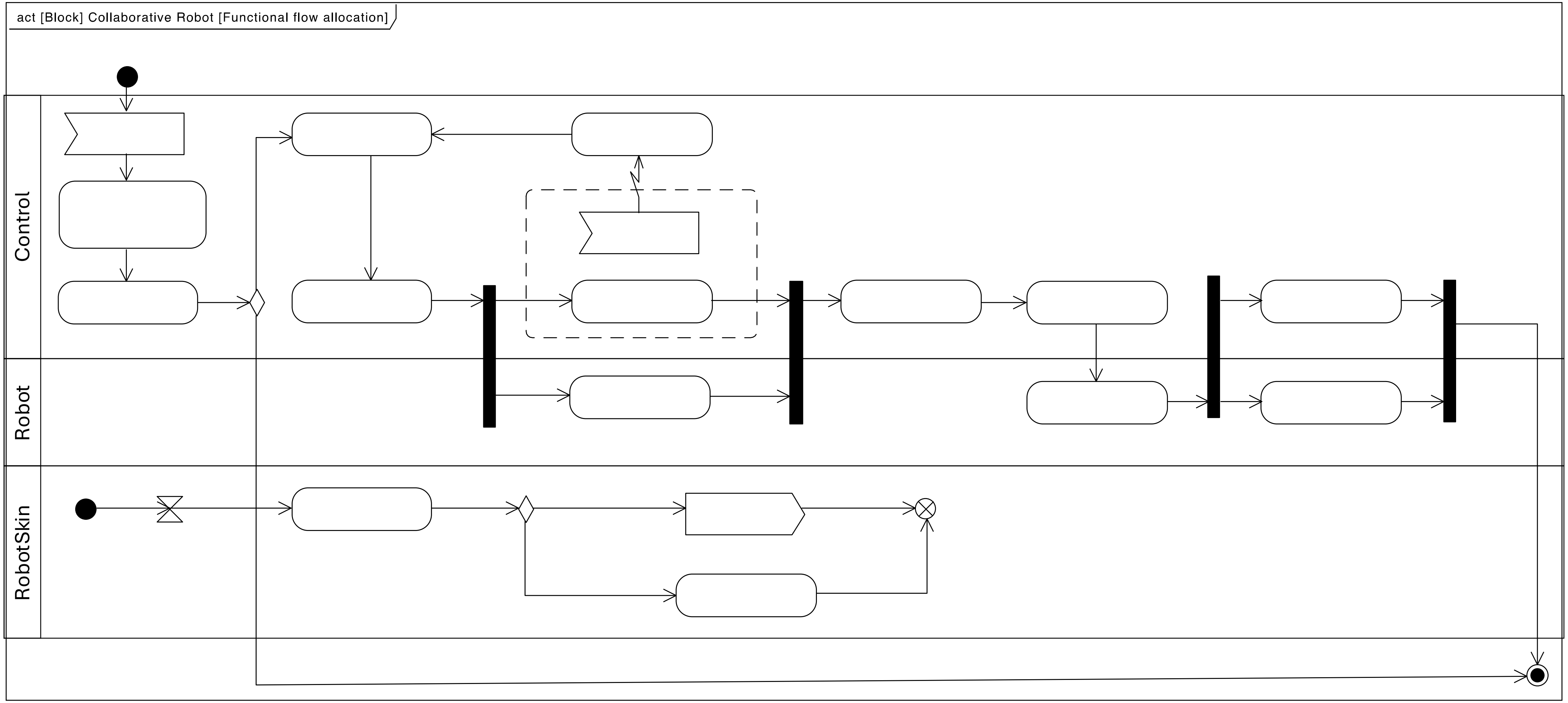
Personal story getting into **MBSE**: self- aware/adaptive/engineering robots

Problem	Approach	Solution
What **knowledge** do we capture?	Systemic, functional, architectural	**SE**
How do we **formalize** it?	Modelling	**MBSE**

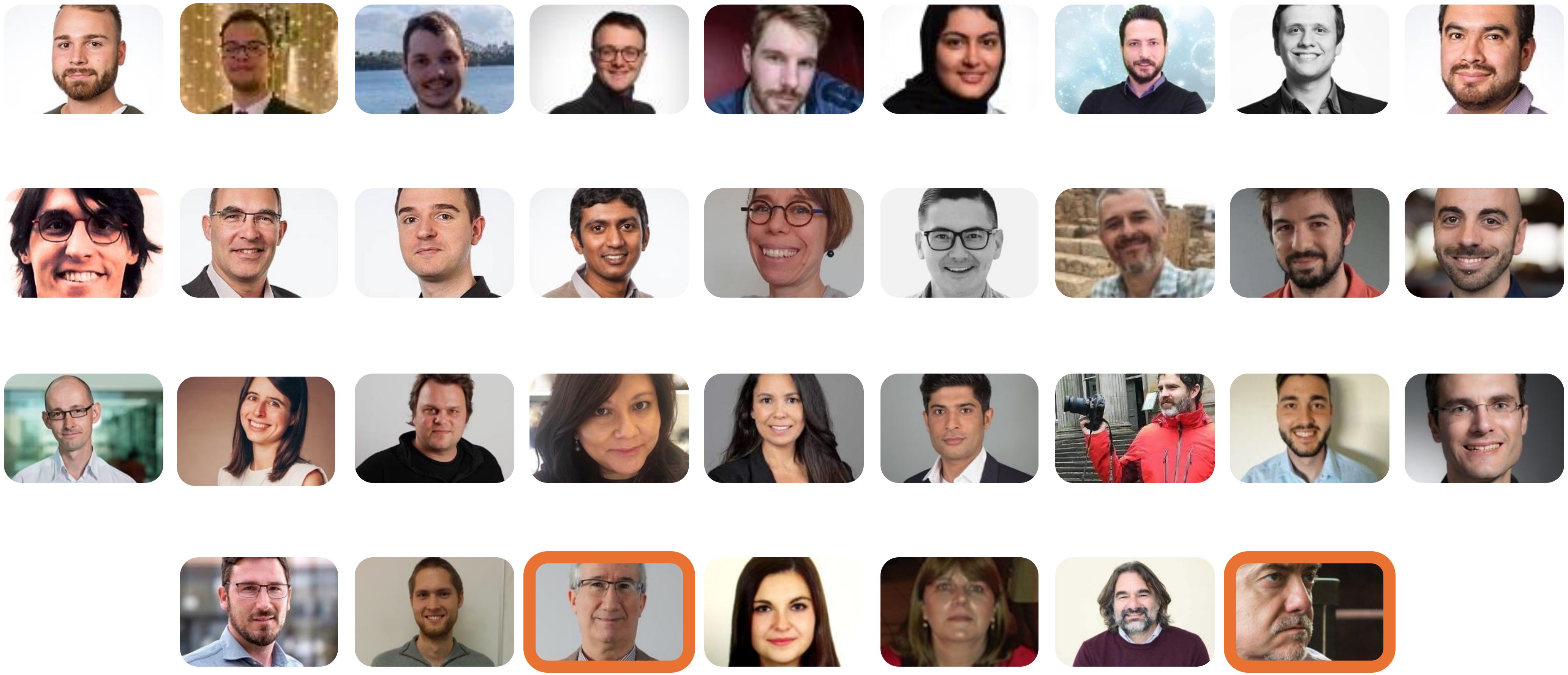
Example: collaborative robot

[Hernandez-2017]

Example: collaborative robot **modeling**



People are key!



Corrado Pezzaro, Gustavo Rezene, Eugenio Tamassia, Alex Gabriel, Forough Zamani, Burak Sisman, Darko Bozhinoski, Mario Garzon , Pierre Mercuriali, Martijn Wisse, Gijs van der Hoorn, Mukunda Bharatheesha, Geeske Langejans, Tim Djedilbaev, Alexander Peixe, Ilias Gerostathopoulos, Ivano Malavolta, Andrzej Wasowski, Juliane Päßler, Einar Johnsen, Lizeth Tapia, Nadia Hammoudeh Garcia, Hars Deshpande, Francisco Martin-Rico, Jonatan Gines, Ralph Lange, Arne Nordmann, Jon Tjerngren, **Jose Luis Fernandez**, Esther Aguado, Julita Bermejo, Manuel Rodrigues, **Ricardo Sanz** and many other colleagues and students not fitting in one slide.

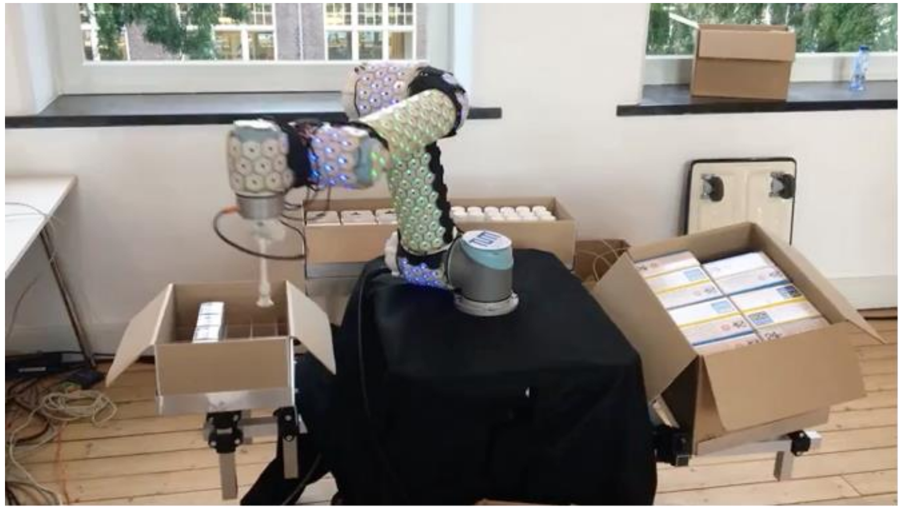
Outline

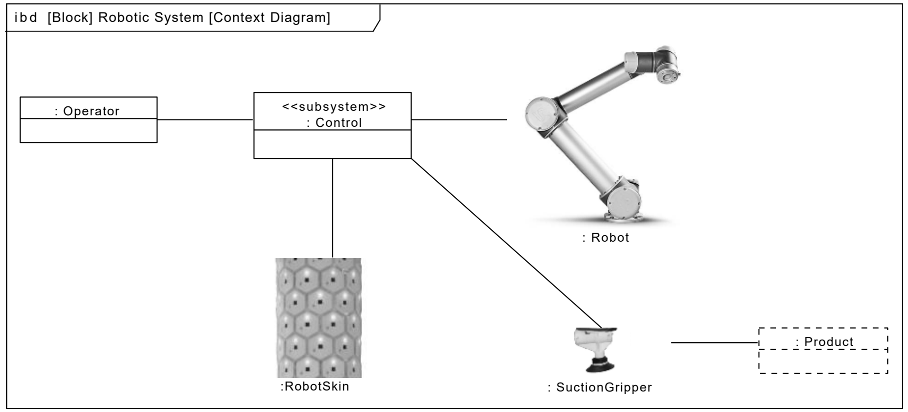
1. SE: setting a common ground
2. MBSE concepts
   - SysML language
   - ISE&PPOOA methodology
3. Group activity:
   Modelling functional architecture with ISE&PPOOA and SysML
4. Next
   - Use of the functional architecture
   - Beyond design-time MBSE: self-engineering

1. Systems Engineering

Our common ground, with examples *a la MBSE*

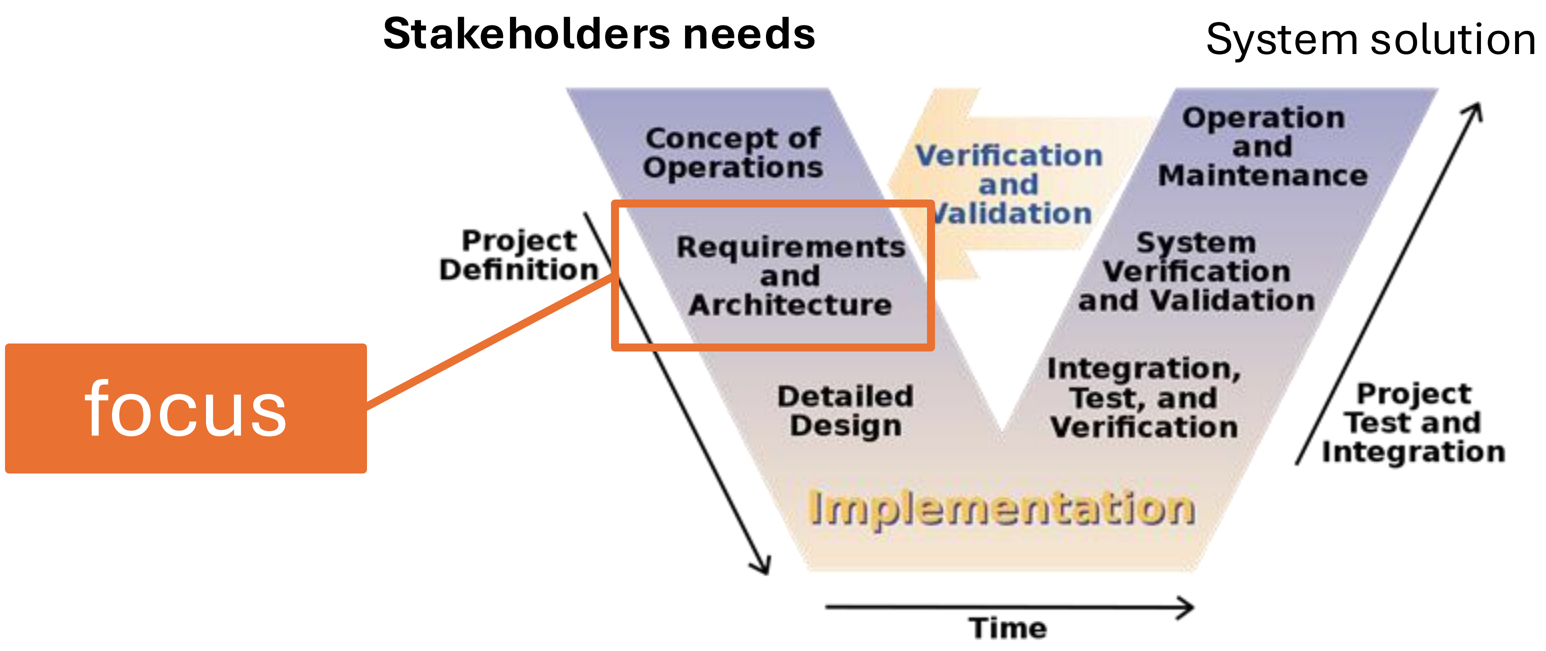
Robotics is about systems ...





- A robot has **parts** that may be either simple or composite.
- A robot **interacts** with the environment.
- A robot fulfils some stakeholder's **needs.**

System life-cycle



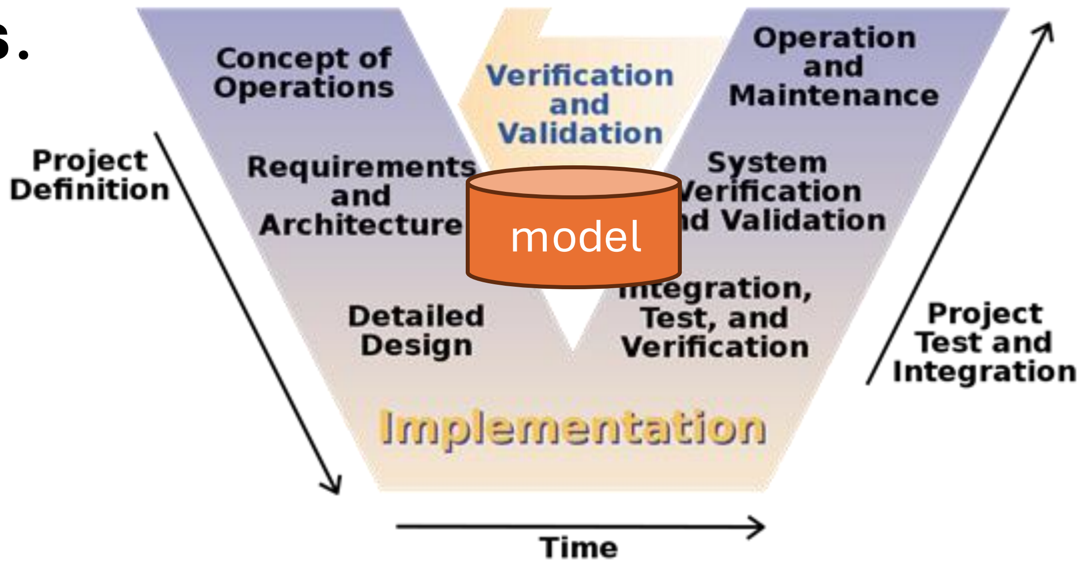
V-model borrowed from *Clarus Concept of Operations* Archived 2009-07-05 at the Wayback Machine, Publication No. FHWA-JPO-05-072, Federal Highway Administration (FHWA), 2005.

2. Model-Based Systems Engineering

Some concepts

Model-Based Systems Engineering

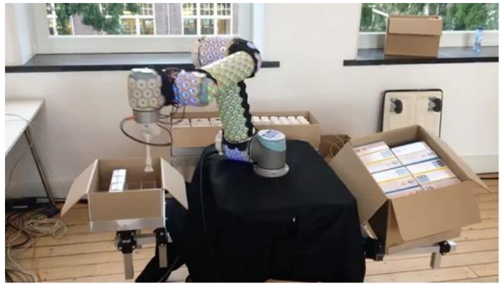
- Use an **integrated system model** for systems engineering, instead of traditional **documents**.
- “**Formal**” model (machine readable)

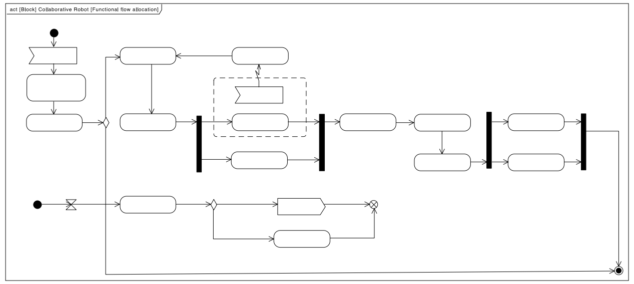


What is a model

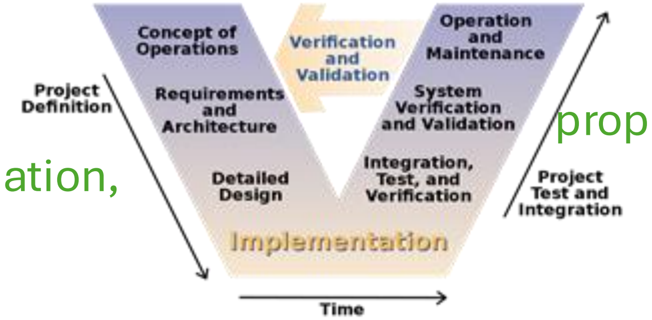
an **abstraction** of a system of interest that is developed for a **purpose**:

- addresses specific **stakeholder concerns/needs**
- have a clear **usefulness to engineering** the system [NASA Exp., 2016]





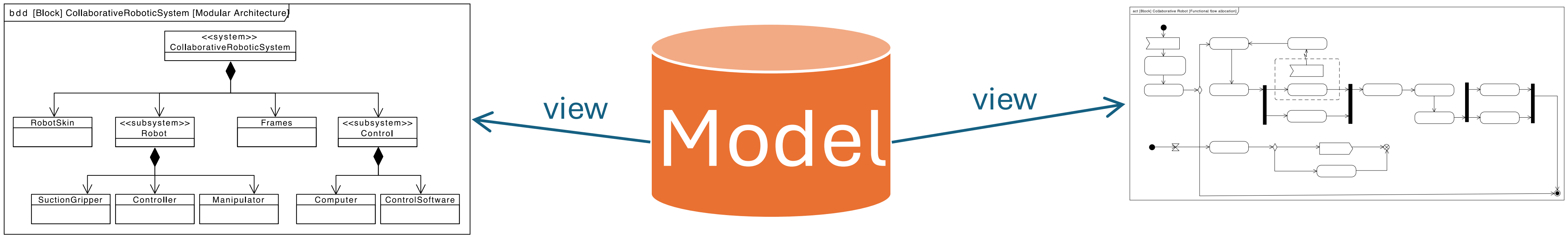
pick the products without harming humans



communication, analysis ...

simulation, property checking ...

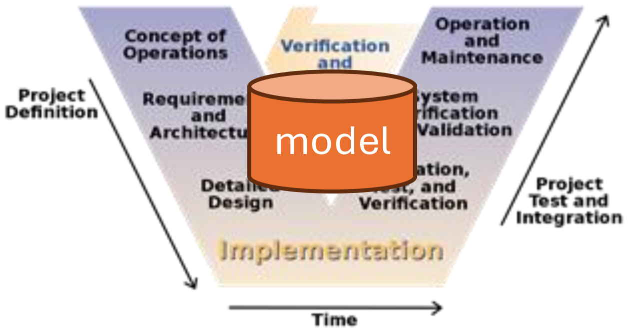
What is a model?

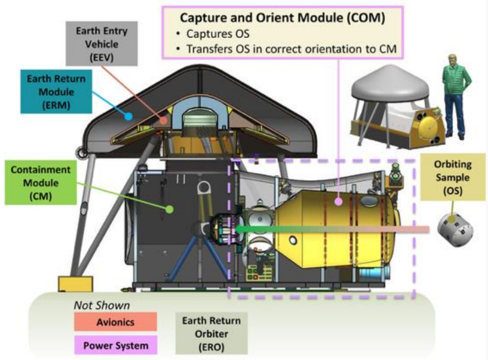


In MBSE, a diagram is NOT the model, it is a **view** (of the model)

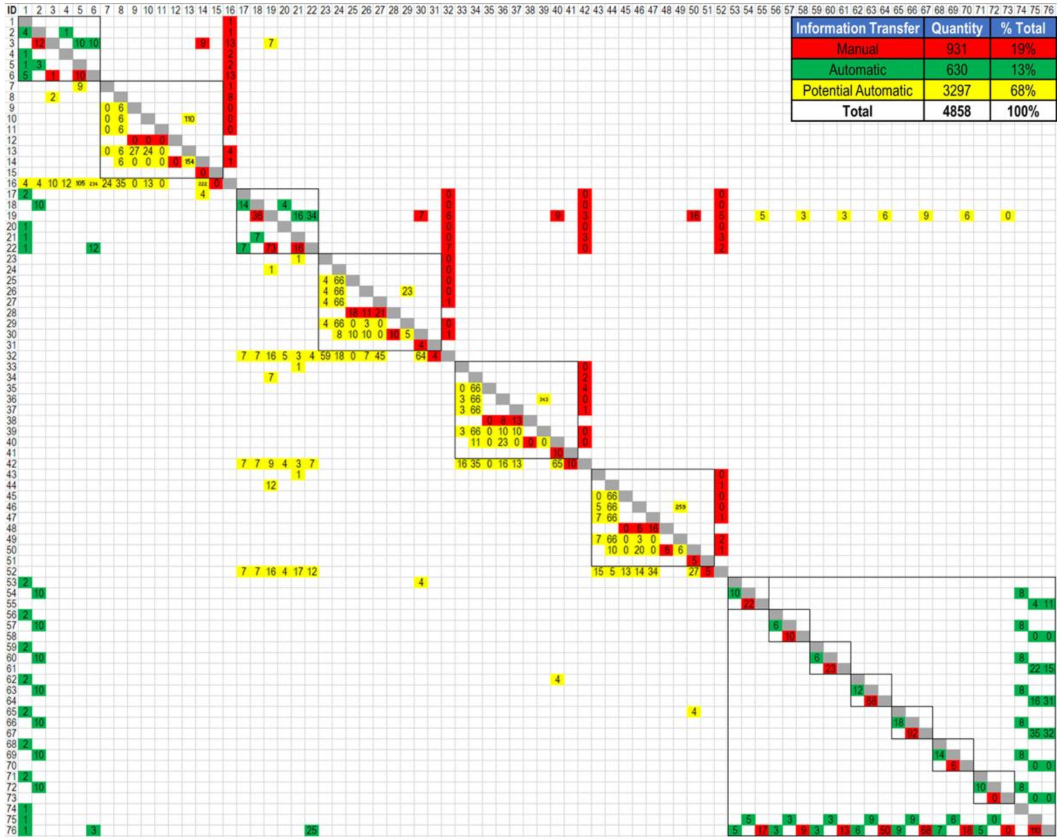
- **View**: *A **representation** of a system from the perspective of a viewpoint.* (ISO/IEC/IEEE 42010)
- **Viewpoint**: *A viewpoint is a specification of the conventions and rules for constructing and using a view for the purpose of addressing a set of **stakeholder** **concerns (E. Aguado's Value! Day2)***

Benefits of MBSE





***automation** in a real space robotics project of information transfers of 13%,*
*with a potential with full-MBSE process to **81%** (Younse-2021)*



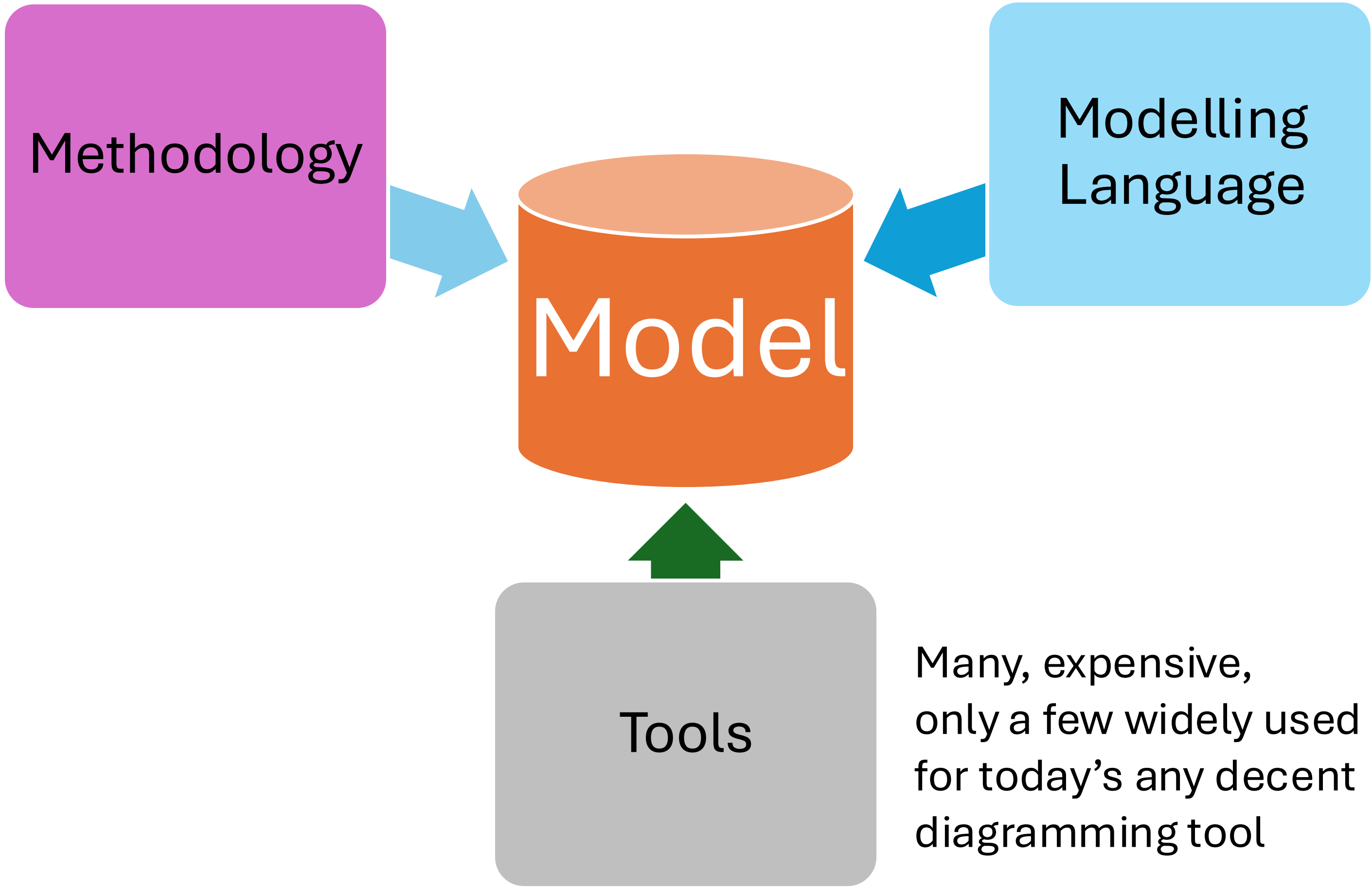
Credit Younse et al (Younse, 2020)

Benefits:

- Reduces design errors
- Prevents rework
- Improves quality and project performance

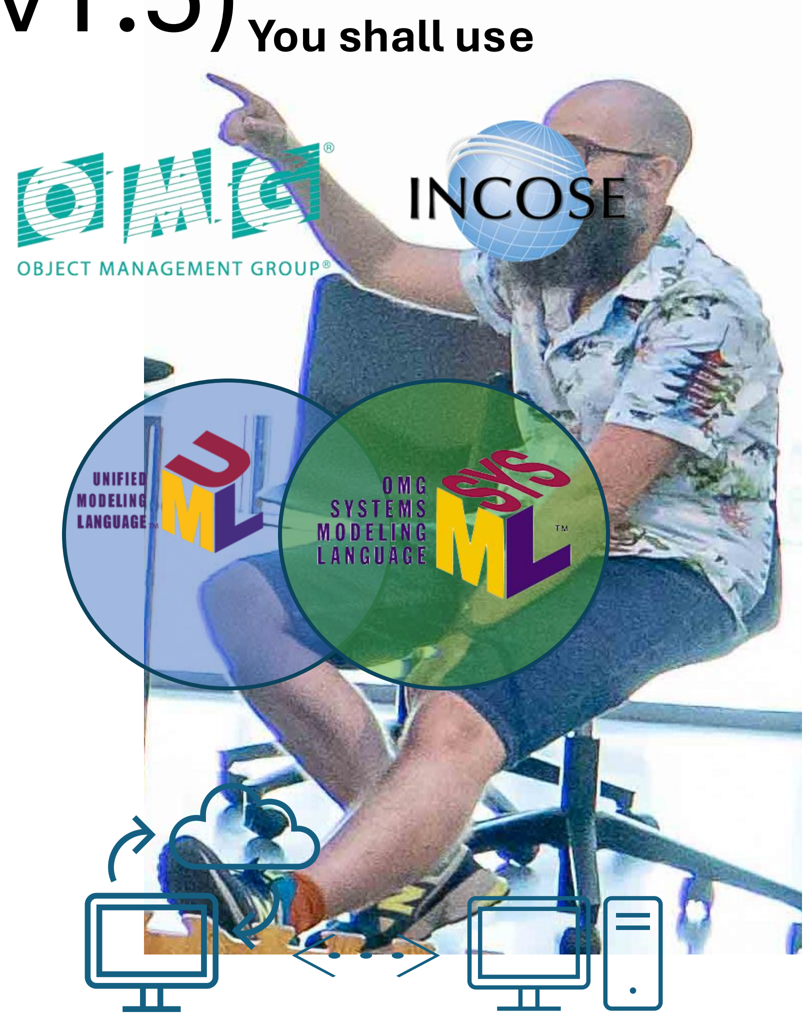
The three pillars of MBSE

Many, we use
ISE&PPOOA



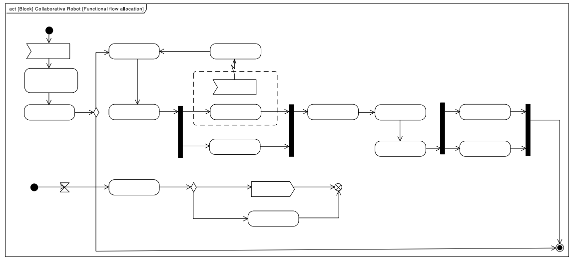
Many, expensive,
only a few widely used in industry,
for today's any decent
diagramming tool

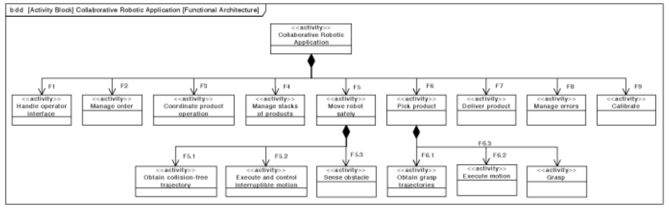
Modelling Language: SysML (v1.5)

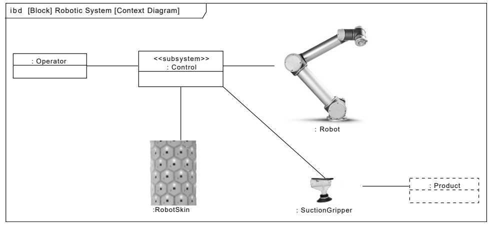


- **general-purpose graphical** modeling language
- for specifying, analyzing, designing, and verifying complex systems (aka **SE**)
- superset of a subset of **UML 2**
- supports **model** and **data interchange** through XML Metadata Interchange

SysML diagrams







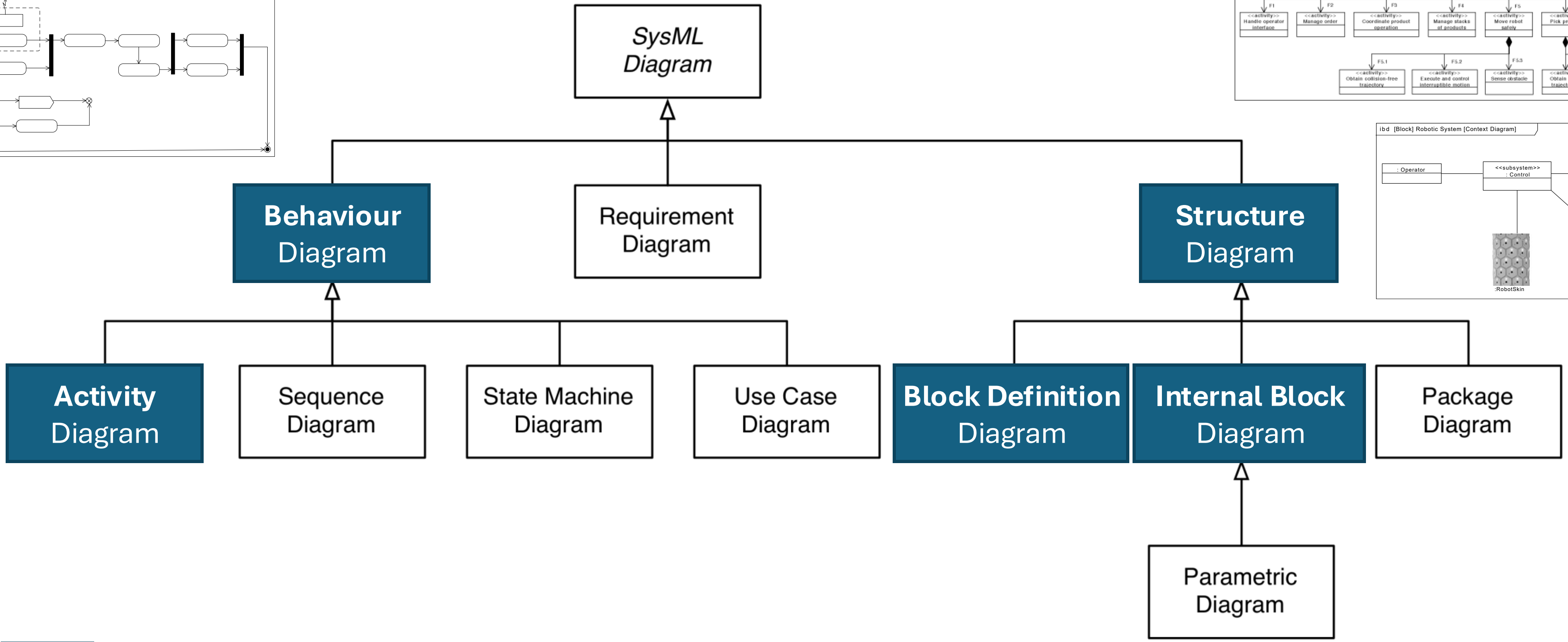
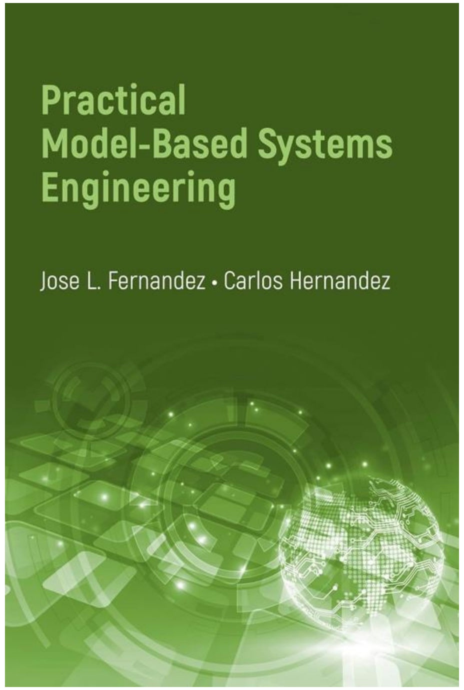


Diagram seen today

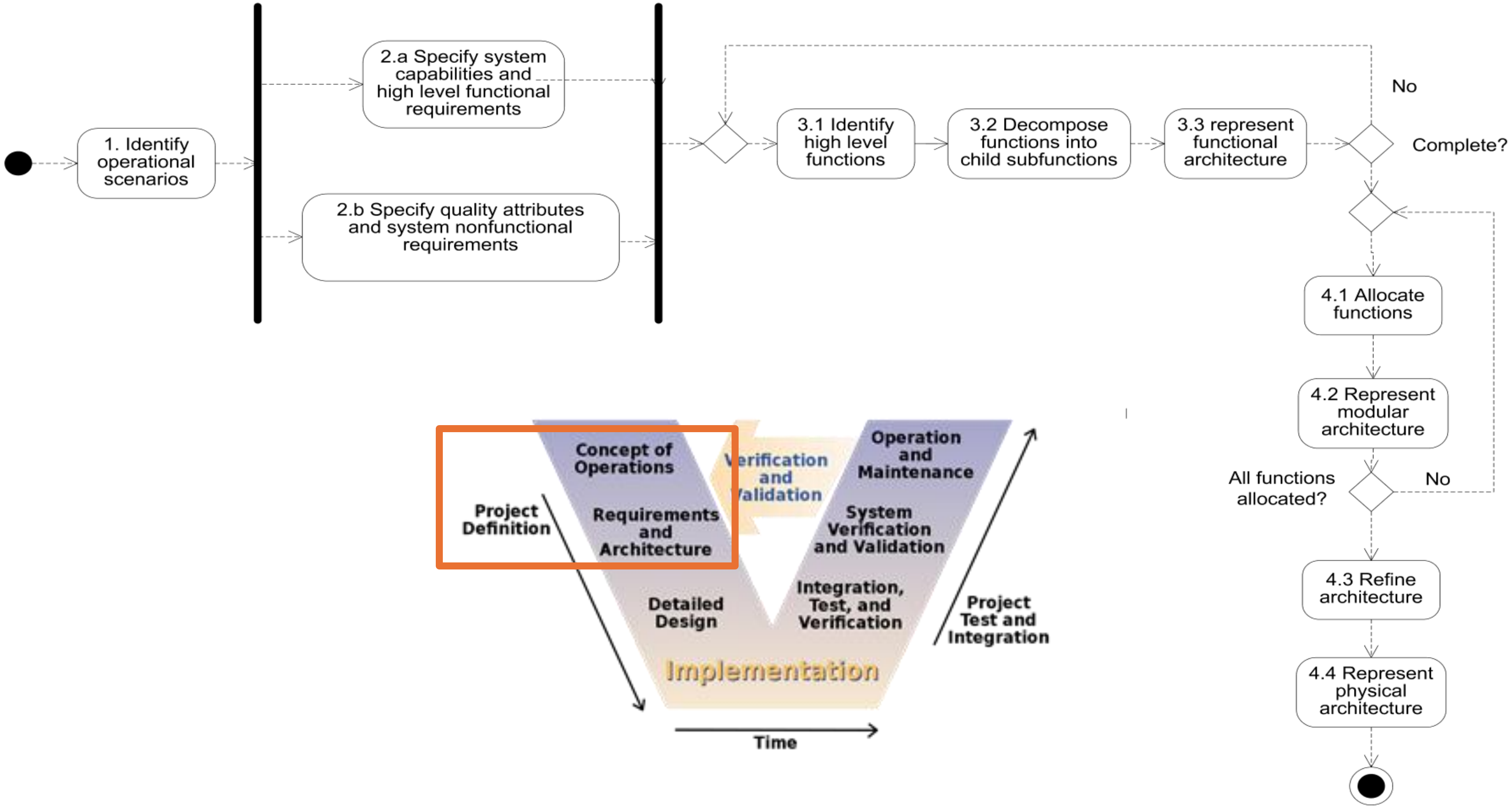
Methodology: ISE&PPOOA

- **I**ntegrated **S**ystems **E**ngineering and **P**ipelines of **P**rocesses in **O**bject-**O**riented **A**rchitectures (**ISE&PPOOA**)
- defines the **processes** (*what*) and **methods** (*how*), e.g. viewpoints
- model-based **systems and software** engineering
- **requirements**-driven,
- focus on the **functional architecture**



[Fernandez-2019]

ISE&PPOOA process



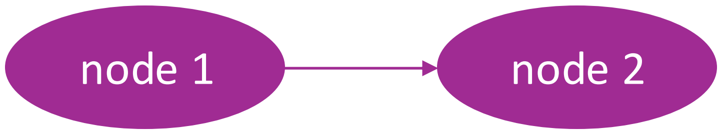
Functional Architecture

Functional architecture

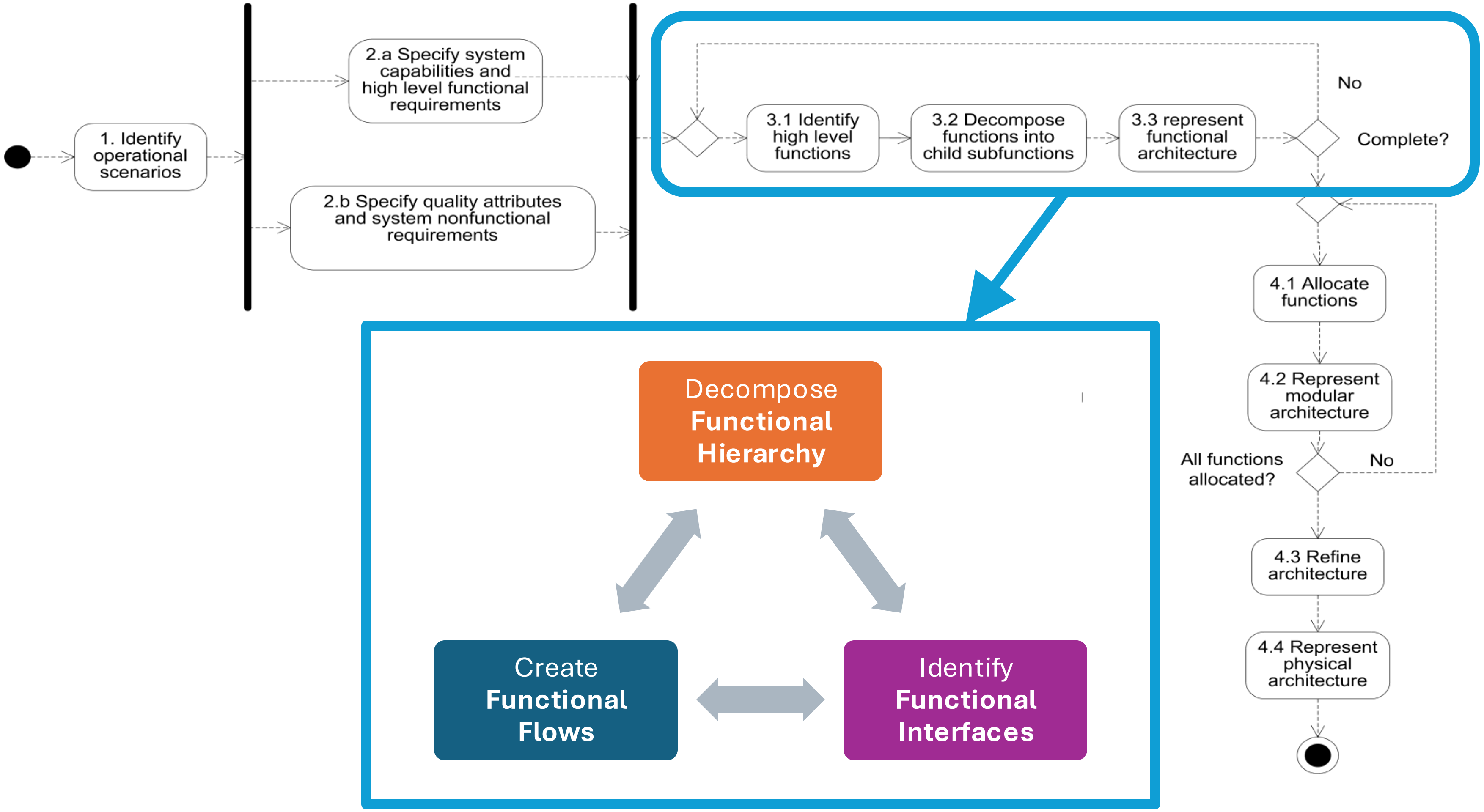
- **function**: transformation of inputs (material, energy or data) into outputs
- **representation of a system's behaviour** (what the system does)
- **independent of technology** solutions (more stable in time)

Physical architecture

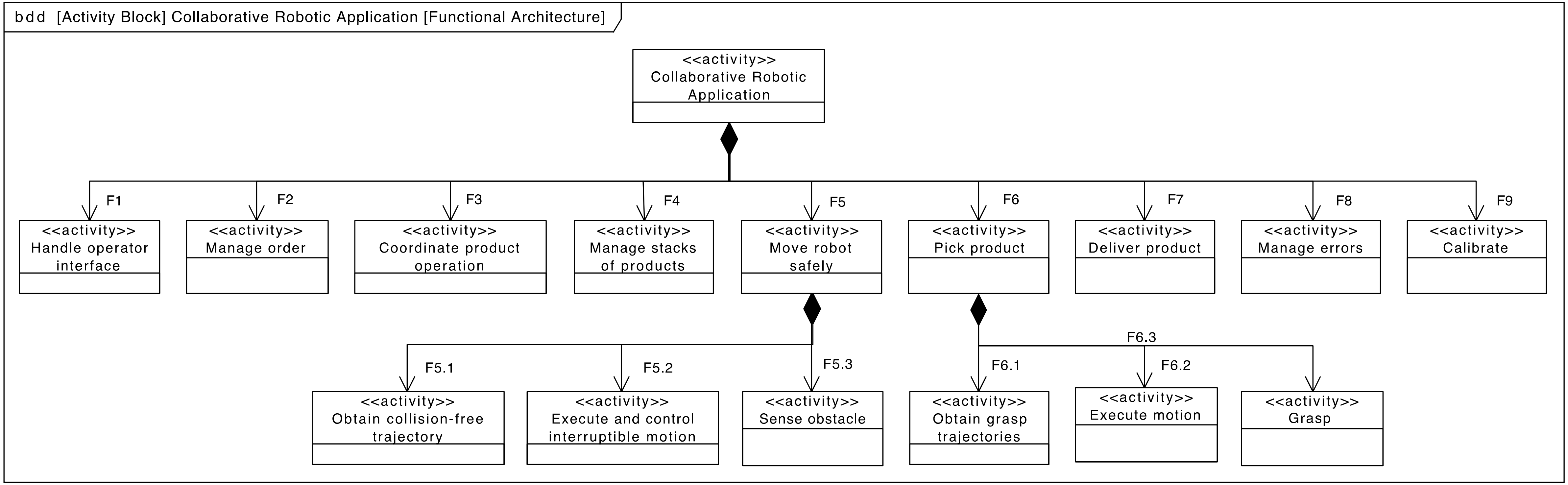
- building blocks (e.g. ROS nodes)
- **represents** the **physical (or software) elements** in the solution
- **dependent of technology** solutions (less stable in time)



Functional architecture



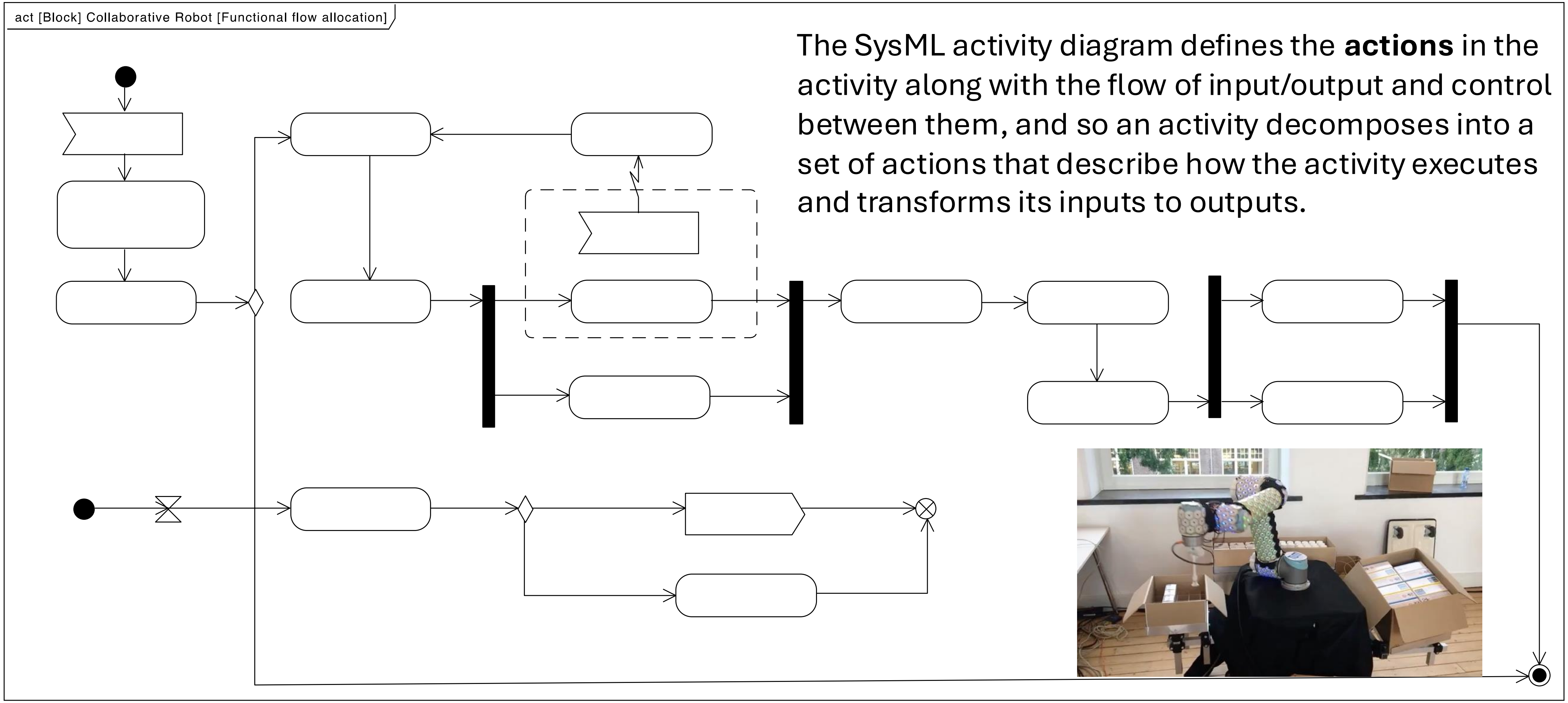
Functional Hierarchy



Functional Flows

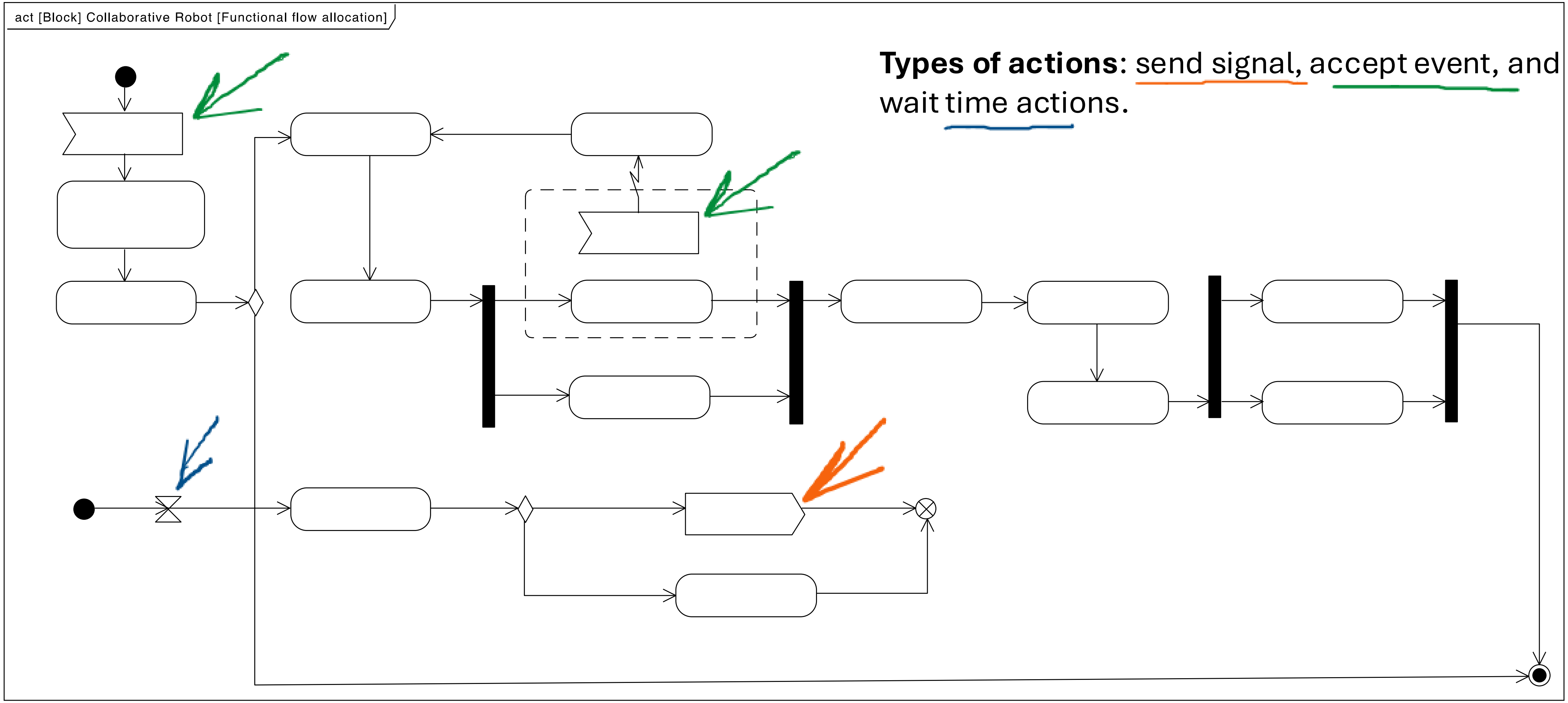
- **Represent** how the system behaves as a **response** to external, internal or time events.
- **Modelled** with **SysML activity diagrams**.
- Maintain **consistency** with Functional Hierarchy, and functional interfaces (see next)

SysML Activity Diagram

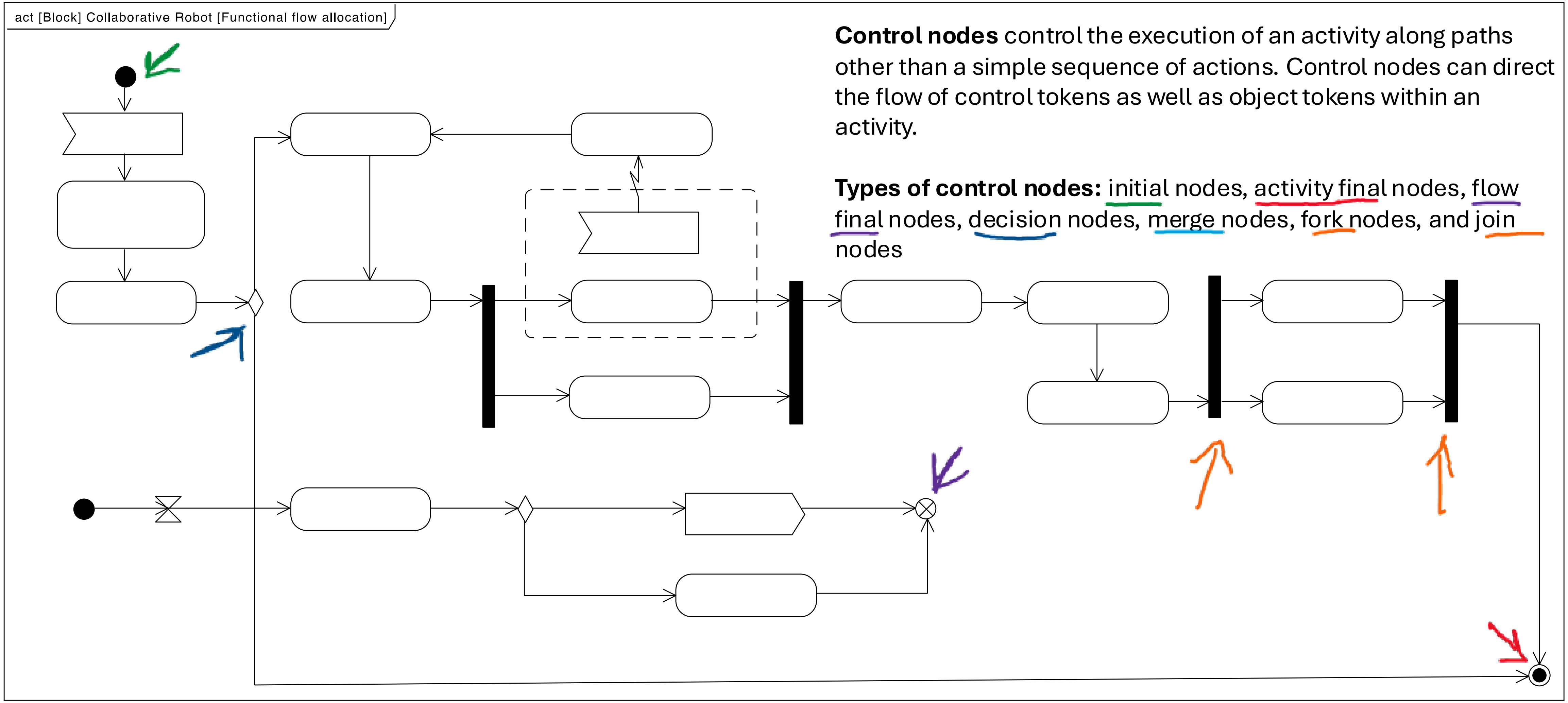


The SysML activity diagram defines the **actions** in the activity along with the flow of input/output and control between them, and so an activity decomposes into a set of actions that describe how the activity executes and transforms its inputs to outputs.

SysML Activity Diagram



SysML Activity Diagram



Control nodes control the execution of an activity along paths other than a simple sequence of actions. Control nodes can direct the flow of control tokens as well as object tokens within an activity.

Types of control nodes: initial nodes, activity final nodes, flow final nodes, decision nodes, merge nodes, fork nodes, and join nodes

Functional interfaces

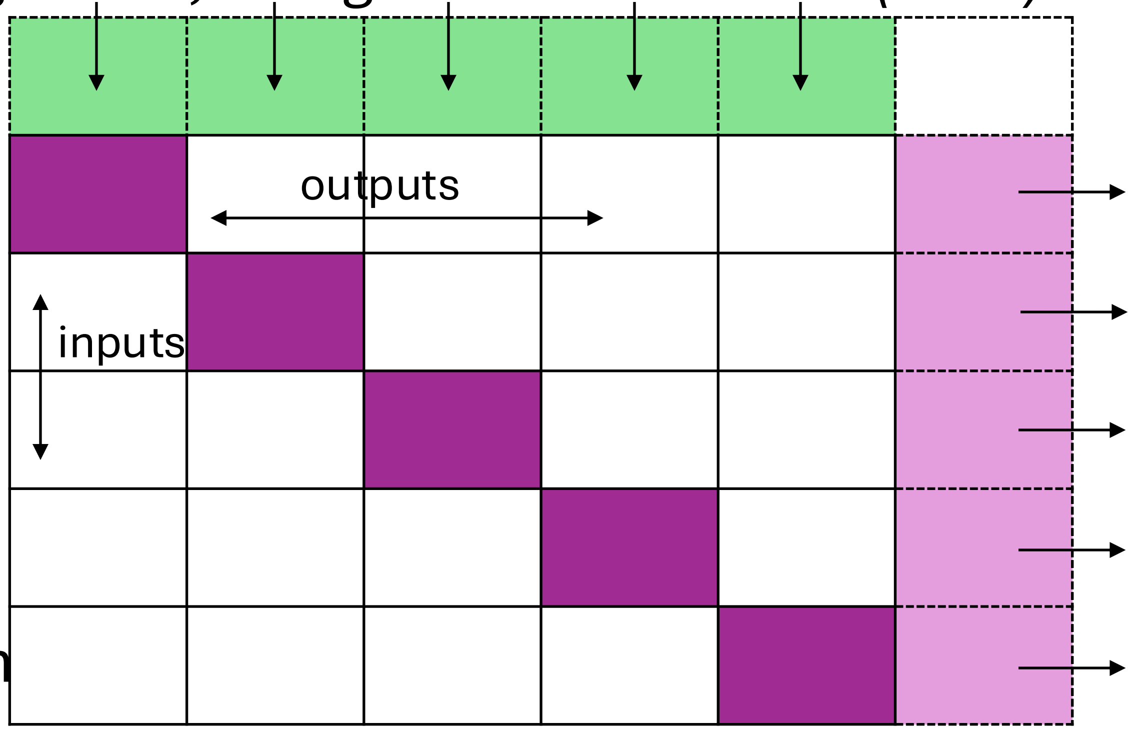
- **Inputs** and **outputs** to a function
- Modelling: N2 chart (and textual description of functions in tabular form)

↓ Product type			↓ Sensor reading	↓ gripper ON		*INPUTS* *OUTPUTS*
Manage stacks	•Stack id			•Next avail. product index		
	Obtain traj. to stack	•Joint trajectory				
		Execute trajectory		•Gripper over stack		
	•Obstacle detected	•Obstacle detected	Sense obstacles			
				Obtain grasp trajectory	•Joint trajectory	
					Execute grasp	→ product picked

What is the N^2 chart?

- **Diagram** that represents **interfaces** between architecture elements (functional or physical)
  - multiple names: *N^2 matrix, Coupling Matrix, Design Structure Matrix (DSM)*
- **matrix** form
  - **N elements** in the **diagonal**
  - **inputs - column**
  - **outputs - row**

  additionally N+1
  - **external inputs**: upper row
  - **external outputs**: rightmost column



Group

3. Activity

Activity: Represent a functional flow given the N2 chart (40min)

Objective: get familiar with MBSE modelling methods and how they relate to traditional SE

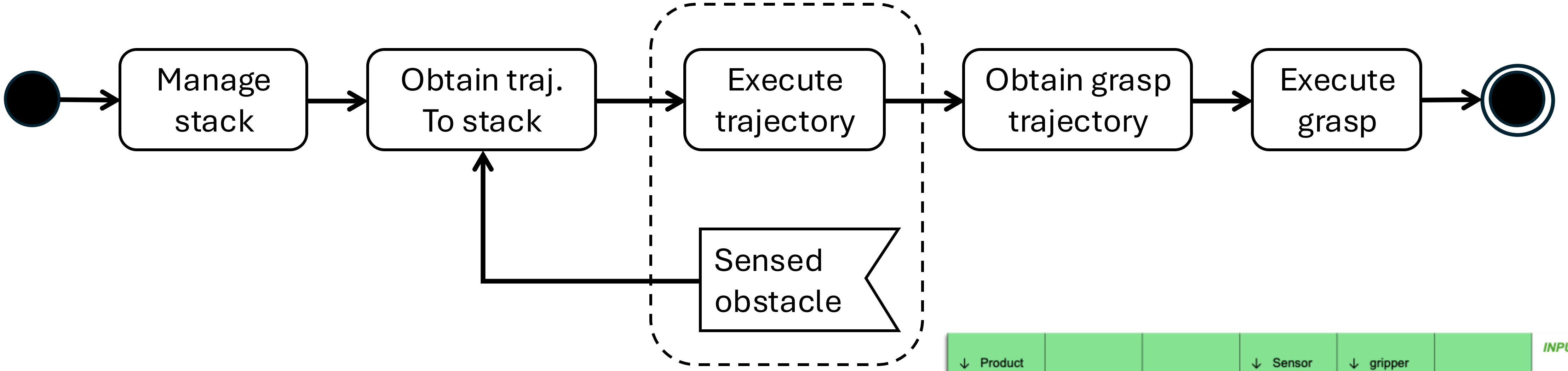
- **What**: propose a specification of the behaviour of our robot as a **functional flow** (only activation) represented in a SysML activity diagram.
- **How**: use the **functions** and their **interfaces** defined in the N2 chat provided. The functions should be the nodes in your diagram.
- **Expected outcome:** SysML activity diagram

- **With your team**, discuss and elaborate the diagram (20min)
  - Use the whiteboards and flipovers to draw your diagrams
- **All together**, discussion of solutions (20min)

N2 chart for “Pick order”

↓ Product type			↓ Sensor reading	↓ gripper ON		*INPUTS* *OUTPUTS*
Manage stacks	• Stack id			• Next avail. product index		
	Obtain traj. to stack	• Joint trajectory				
		Execute trajectory		• Gripper over stack		
	• Obstacle detected	• Obstacle detected	Sense obstacles			
				Obtain grasp trajectory	• Joint trajectory	
					Execute grasp	→ product picked

Possible solution



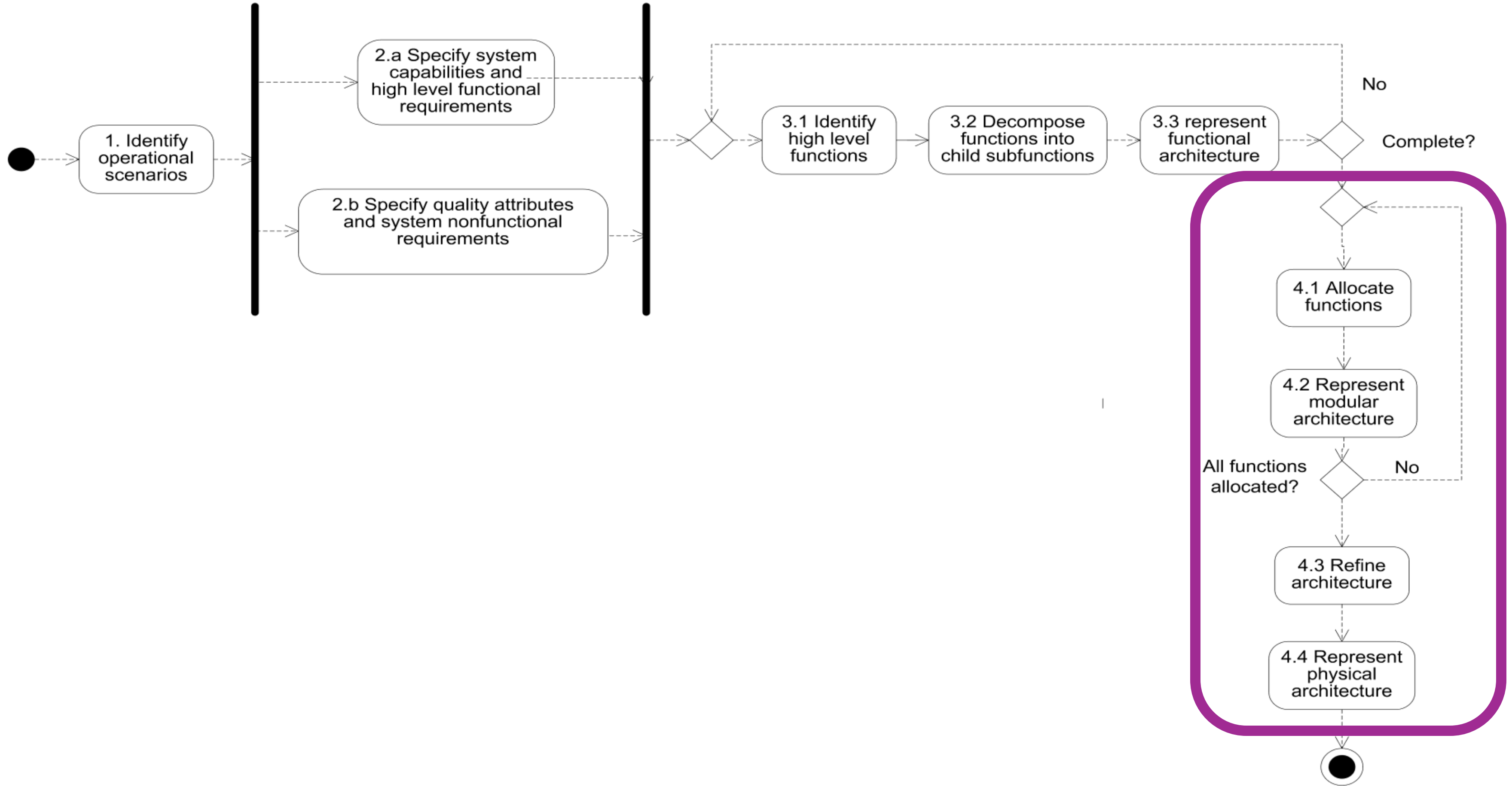
↓ Product type			↓ Sensor reading	↓ gripper ON		*INPUTS* / *OUTPUTS*
Manage stacks	• Stack id			• Next avail. product index		
	Obtain traj. to stack	• Joint trajectory				
		Execute trajectory		• Gripper over stack		
	• Obstacle detected	• Obstacle detected	**Sense obstacles**			
				Obtain grasp trajectory	• Joint trajectory	
					Execute grasp	→ product picked

Tip: for how the use of N2 chart can be integrated into MBSE, check [Garcia-2021]

4. Next

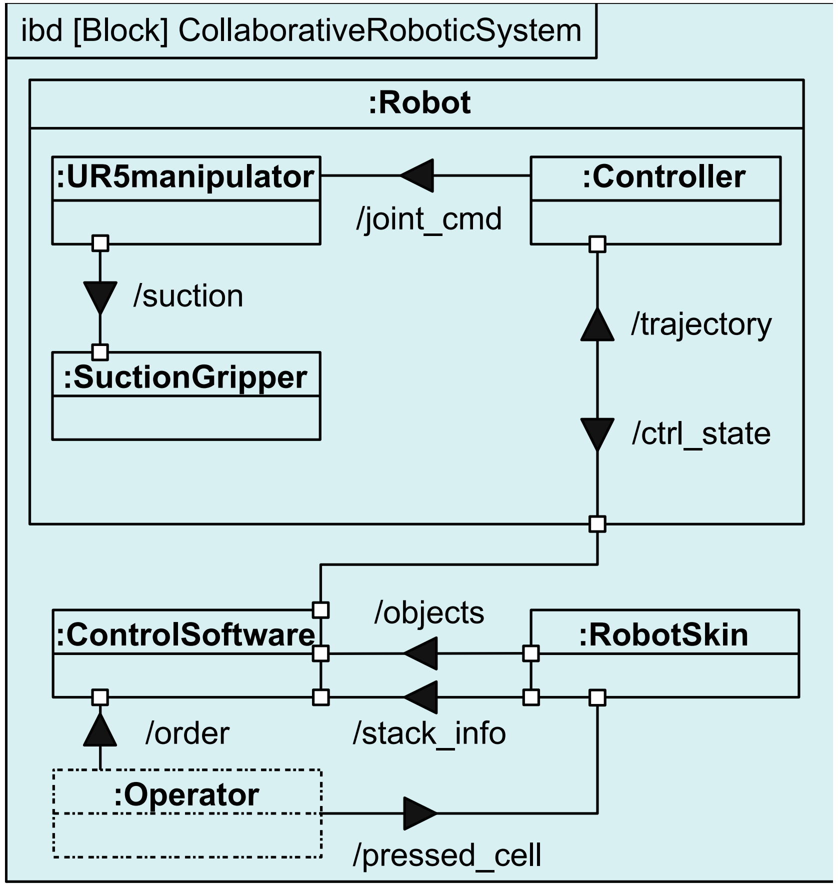
So what do I do with my functional architecure?

What next in ISE&PPOOA MBSE process?
Allocation of functions to physical architecture

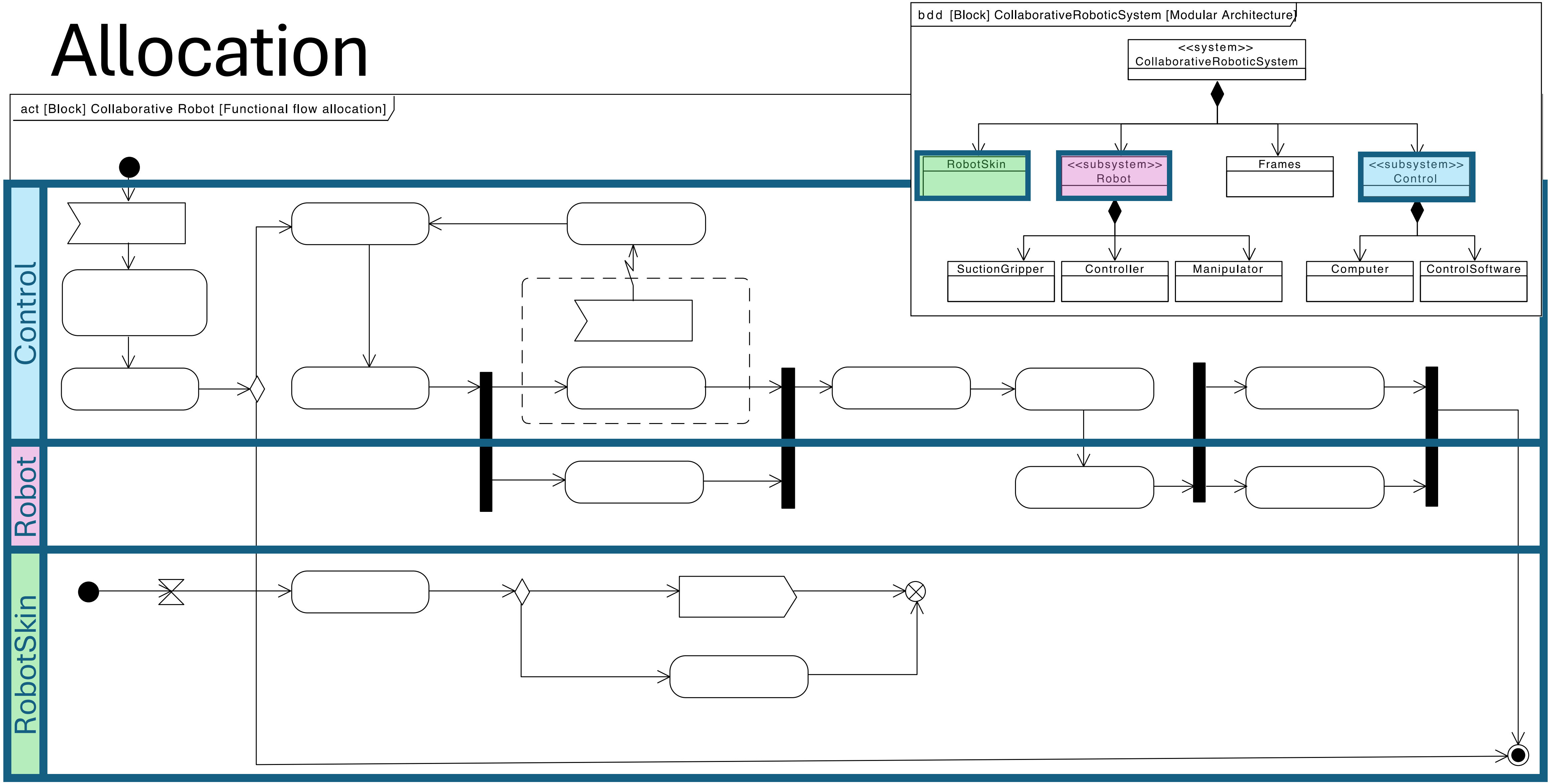


Physical architecture

- Building blocks of your system, including **software (architecture)**
- SysML Internal Block Definition Diagram (ibd)



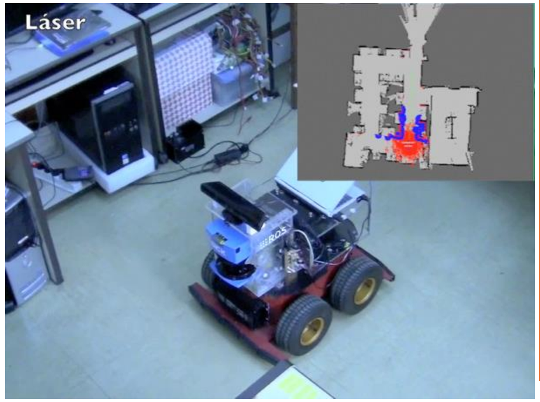
Allocation

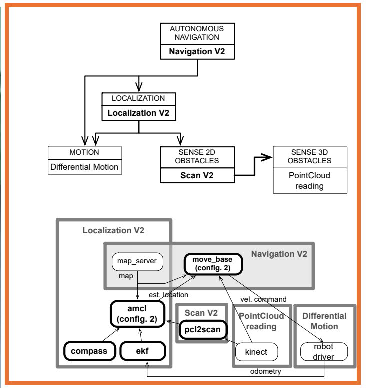

act [Block] Collaborative Robot [Functional flow allocation]
Control
Robot
RobotSkin
bdd [Block] CollaborativeRoboticSystem [Modular Architecture]
<<system>>
CollaborativeRoboticSystem
RobotSkin
<<subsystem>>
Robot
Frames
<<subsystem>>
Control
SuctionGripper
Controller
Manipulator
Computer
ControlSoftware

Beyond

design-time MBSE and into runtime self-engineering

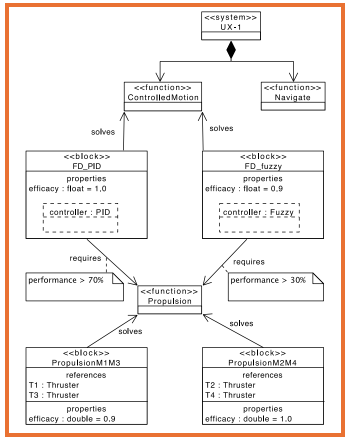
Model-based Architectural Self-adaptation

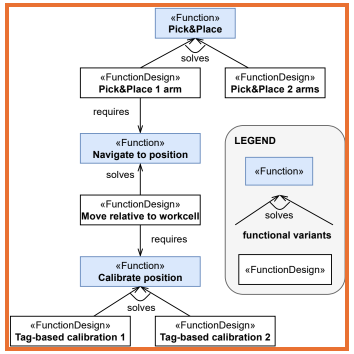




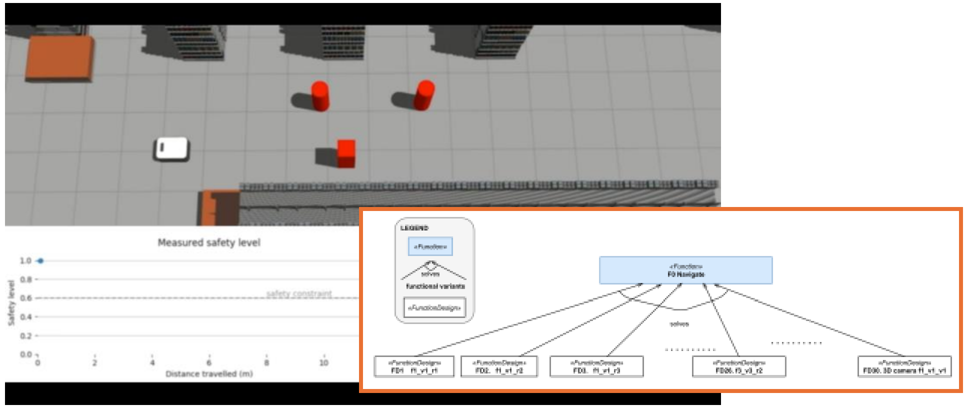
2013

2018





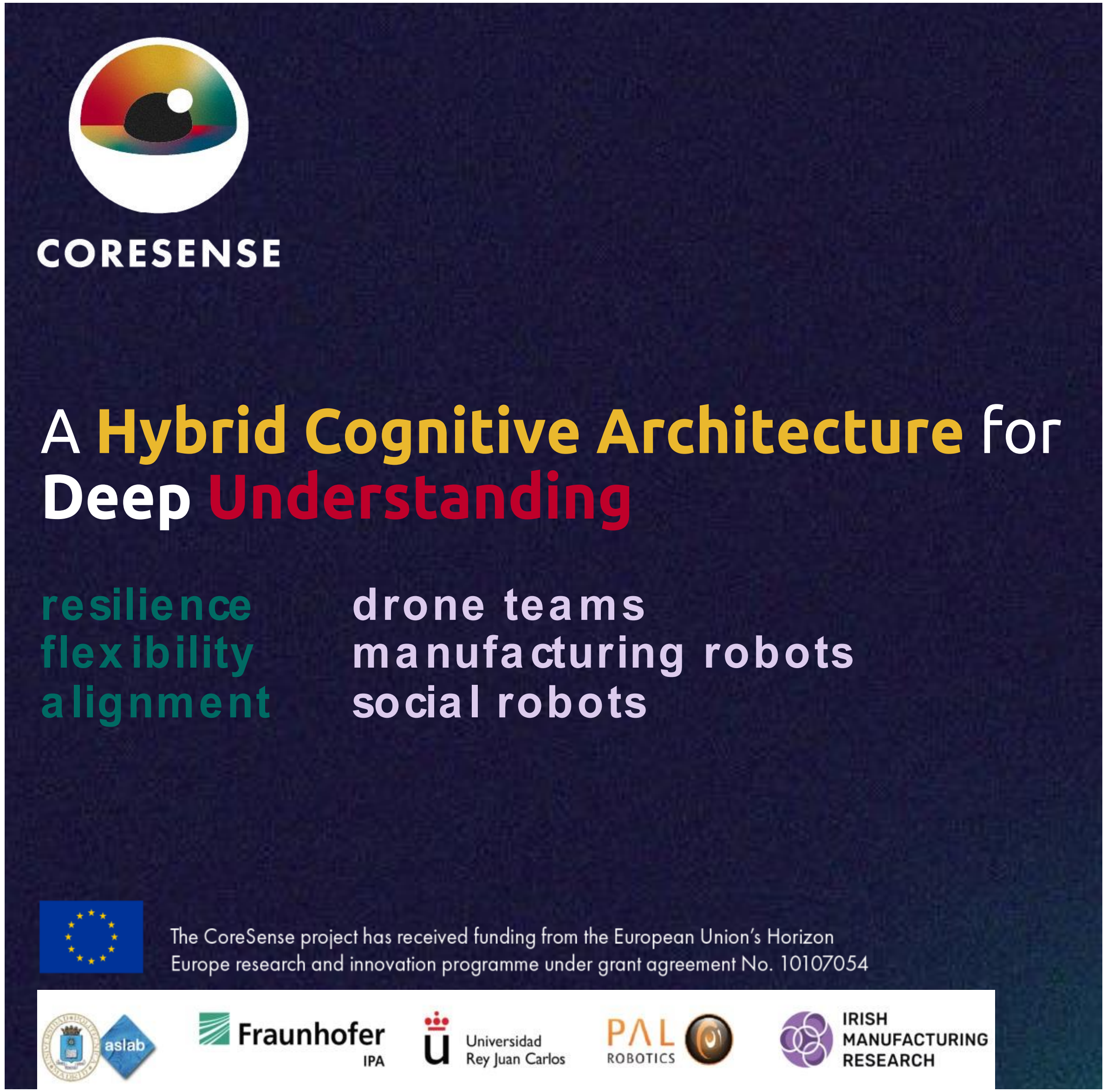
2019



2021

CoreSense

- Robots that **understand** and are **aware** of their stakeholders' **needs**
- Robot's **reasoning** over a **SysML2** model that integrates all system knowledge and concerns.
- **New discipline** of developing autonomous cognitive systems.

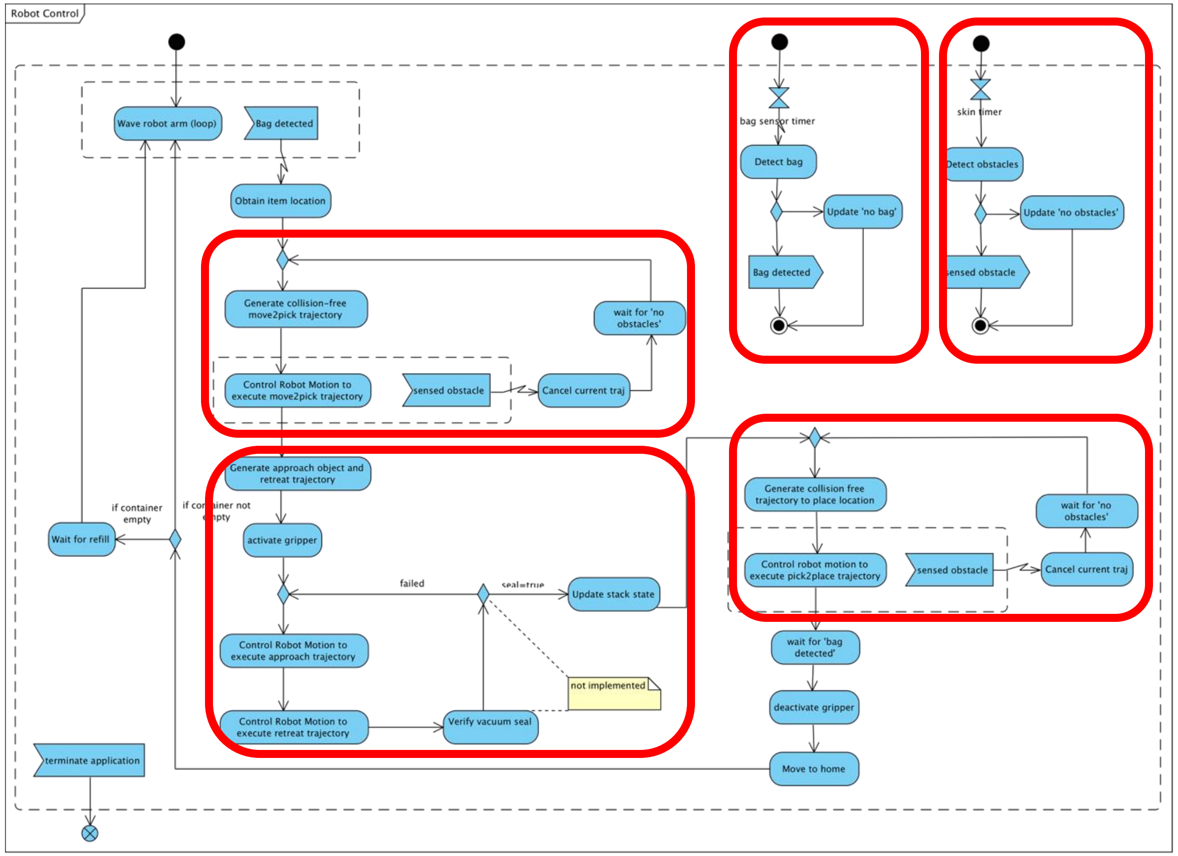


Conclusion: models help you (and robots) think!

Example: Bottom-up

You can obtain or refine the **functional hierarchy** from **activity diagrams**

- Model responses of the system in the scenarios (main flows).
- Clustering of actions into activities (functions)



References

- [Hernandez-2017] C. Hernandez Corbato and J. L. Fernandez-Sanchez. Model-based systems engineering to design collaborative robotics applications. In 2017 IEEE International Systems Engineering Symposium (ISSE), pages1–6, Oct 2017.
- [Hernandez-2018] C. Hernandez Corbato, J. Bermejo-Alonso, and R. Sanz. A self-adaptation framework based on functional knowledge for augmented autonomy in robots. Integrated Computer-Aided Engineering, 25(2):157–172, 2018.
- [Fernandez-2019] J. L. Fernandez-Sanchez and C. Hernandez. Practical Model-Based Systems Engineering. Artech House, 2019.
- [Bozhinoski-2021] D. Bozhinoski, J. Wijkhuizen. Context-based navigation for ground mobile robot in semi-structured indoor environment. In IEEE International Conference on Robotic Computing (IRC), accepted 2021.
- [Aguado-2021] E. Aguado, Z. Milosevic, C. Hernandez, R. Sanz, M. Garzon, D. Bozhinoski, and C. Rossi. Functional Self-Awareness and Metacontrol for Underwater Robot Autonomy. Sensors, 21(4), 2021.
- [Bozhinoski-2022] D. Bozhinoski, M. G. Oviedo, N. H. Garcia, H. Deshpande, G. van der Hoorn, J. Tjerngren, A. Wasowski, and C. Hernandez. MROS: runtime adaptation for robot control architectures. Advanced Robotics, 0(0):1– 17, 2022.
- [IEEE Std. 1220] IEEE 1220 Standard for Application and Management of the Systems Engineering Process. New York, NY: Institute of Electrical and Electronic Engineers, 2005.
- [NASA Handbook] NASA. NASA Systems Engineering Handbook. National Aeronautics and Space Administration,, NASA sp-2016-6105 rev2 edition, 2016.
- [NASA, Exp., 2016] Expanded guidance for NASA systems engineering, volume 2: Crosscutting topics, special topics, and appendices. Technical report, National Aeronautics and Space Administration, 2016.
- [Younse-2021] P. J. Younse, J. E. Cameron and T. H. Bradley, "Comparative Analysis of Model-Based and Traditional Systems Engineering Approaches for Architecting a Robotic Space System Through Automatic Information Transfer," in IEEE Access, vol. 9, pp. 107476-107492, 2021

Additional materials about ISE&PPOOA

- **Website**: https://www.ppooa.com.es/resources/
- **LinkedIn** group: https://www.linkedin.com/groups/9500835/
- **Videos** from the TU Delft REMARO Summer school:
  - Model-Based Systems Engineering [collegerama]:
  - The ISE&PPOOA methodology for MBSE [collegerama]
  - Functional Architecture with the ISE&PPOOA methodology [collegerama]

Thank you for participating!

Happy to follow up with any questions or comments