Welcome to the Webinar! Please note that we have moved to the ZOOM platform. Please join ZOOM audio (Voice over Internet) if you are able to connect. Otherwise, please see the webinar invitation for dial-in phone lines

INCOSE Webinar Series

Wednesday 18th November 2020 – Webinar 145

ISE&PPOOA a MBSE Methodology from System to Software Architecture

Dr Jose L. Fernandez and Dr Carlos Hernandez

With thanks to our Sponsor for 2020

INCOSE

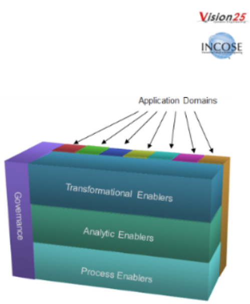
LOCKHEED MARTIN

INCOSE is offering Webinars

INCOSE CSEP — Certified Systems Engineering Professional

27th annual INCOSE international symposium
Adelaide, Australia
July 15 - 20, 2017

INCOSE Corporate Advisory Board Member



INCOSE Academic Council Member

INSIGHT

Systems Engineering

2017 annual INCOSE international workshop
Los Angeles, CA, USA
January 28 - 31, 2017

- To provide a forum for experts in the field of Systems Engineering to present information on the “State of the Art”
- To explain how INCOSE works, and how to make the most out of INCOSE membership

INCOSE Systems Engineering Professional PDU Credit

Please note that you can claim 1PDU credit towards your Systems Engineering Professional re-certification by attending this webinar. INCOSE webinars may also apply to the PDU requirements of other organizations, depending on the subject matter

To qualify, you must have attended through at least 75% of the webinar for webinars that last less than one hour, or through 45 minutes of the webinar for webinars that last for 1 hour or longer.

Here is the link to details about certification renewal, including information on PDUs.
http://www.incose.org/certification/CertProcess/CertRenew

Choreography

1. Andy Pickard (your host) will introduce the Webinar and the speakers
2. Jose and Carlos will speak for about 40 to 45 minutes
3. During their talk, participants can write questions using the Zoom Q&A window
4. After Jose and Carlos complete their talk, they will spend 10 minutes answering questions that Andy selects from those submitted by the audience
5. Andy Pickard will provide information about upcoming Webinars and then end this session
6. This Webinar is being recorded and will be made available on the INCOSE website to members and employees of CAB organizations

ISE&PPOOA a MBSE Methodology from System to Software Architecture.

José L. Fernández
jose.fernandez@incose.org

Carlos Hernández
C.H.Corbato@tudelft.nl
Delft University of Technology

Presenters

Jose L. Fernandez has a PhD in Computer Science, and an Engineering Degree in Aeronautical Engineering, both by the Universidad Politecnica de Madrid.

He has over 30 years of experience in industry as system engineer, project leader, researcher, department manager and consultant. He was involved in projects dealing with software development and maintenance of large systems, specifically real-time systems for air traffic control, power plants Supervisory Control and Data Acquisition (SCADA), avionics and cellular phone applications. He was associate professor at the E.T.S. Ingenieros Industriales, Universidad Politecnica de Madrid (UPM).

He is senior member of the IEEE (Institute of Electric and Electronics Engineering) and member of INCOSE (International Council on Systems Engineering), participating in the software engineering body of knowledge, systems engineering body of knowledge and requirements engineering working groups of these associations. He is member of the PMI (Project Management Institute) participating as reviewer of the PMBoK 6th Edition, 2017, and the Requirements management, Practice Guide, 2016.

Carlos Hernandez has a PhD in AI and Robotics, and MSc degrees in Industrial Engineering and Automation and Robotics, both by the Universidad Politecnica de Madrid.

He is an assistant professor at the Cognitive Robotics Department of TU Delft since May 2019, and principal investigator in the AIRLab Delft, a research lab on AI and robotics for retails sponsored by Ahold Delhaize. In 2016, Carlos led Team Delft to win the Amazon Robotics Challenge. He is currently coordinator of the ROSIN and the MROS European projects on robot software, and he has previously participated in projects related to cognitive robotics and factories of the future.

Contents

1. What do we mean by a MBSE methodology?

MBSE methodology definition

- A **Model-Based Systems Engineering Methodology** can be characterized as the collection of related processes, methods and tools used to support the discipline of systems engineering in a model-based or model-driven context[1]

[1]Stefan et al., “Survey of Model-Based Systems Engineering (MBSE) Methodologies”. INCOSE-TD-2007-003-01)Version/Revision: B, 10 June 2008. INCOSE, Seattle (WA),2008.

- **Methodology**= **processes** + **methods** + **tools**

2.Integrated Systems Engineering and Pipelines of Processes in Object Oriented Architectures (ISE&PPOOA) methodology

PPOOA

- **PPOOA** is an architectural framework ("architecting steps"+"building elements") for real-time software.
- The main building elements of PPOOA framework are software components and coordination mechanisms.
- The main software components are the "domain class" and the "process" implementing an independent thread of control.
- Coordination mechanisms are the building elements supporting synchronization and communication.
- A PPOOA "process component" may be implemented using an Ada task, java thread or by the light processes supported by the real-time operating system used.

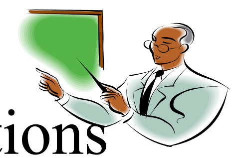
Software components and coordination mechanisms provided by the PPOOA arch. framework

Components

Controller :
Manages external events

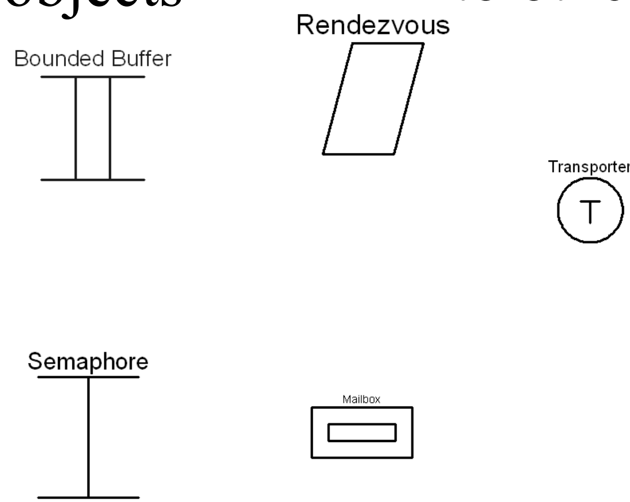
Domain component/ Algorithmic component:
Performs operations

Structure:
Maintains relations between objects



Process:
Coordinates work to others

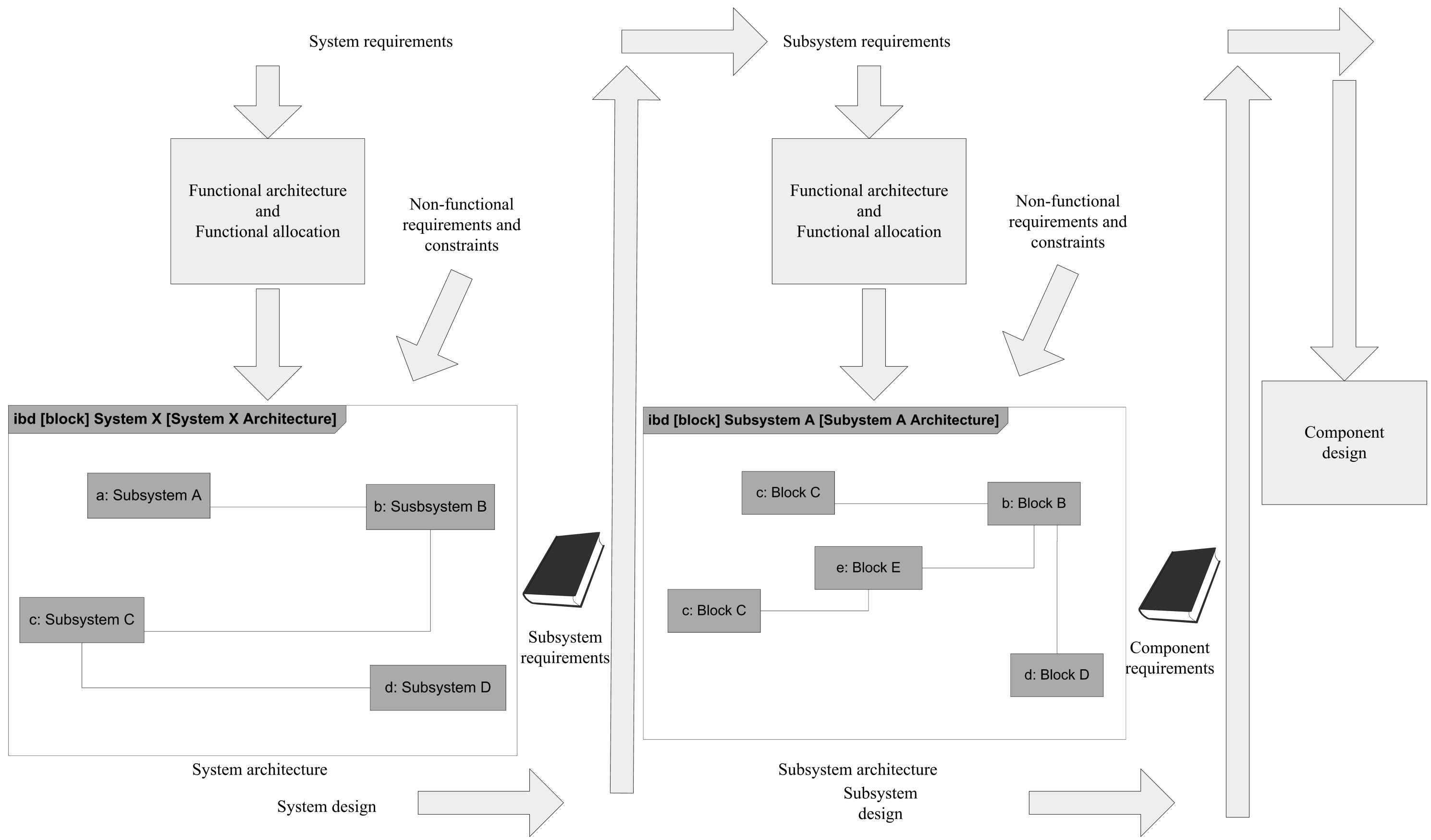
Coordination Mechanisms:
Synchronization+ communication



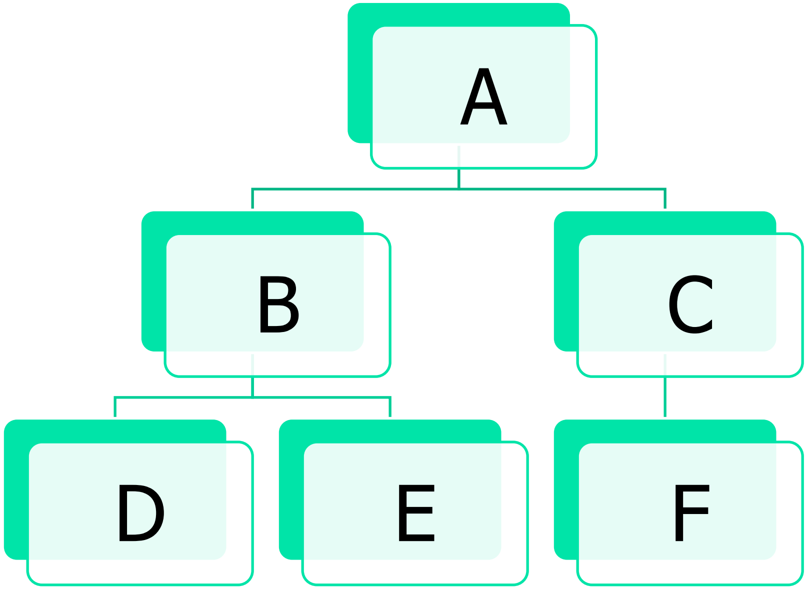
ISE&PPOA

- **ISE&PPOA** integrates systems engineering with PPOOA so it can be applied to non software systems as well
- **ISE&PPOOA** is a requirements-driven, model based systems engineering approach where the main outcomes are the functional and physical architectures of the product, system or service to be developed
- We are proposing a **way of thinking consistently** to solve an engineering problem, where identifying the functions and quality attributes of the product to be developed is a main issue to synthesize the solution

ISE&PPOOA is requirements driven



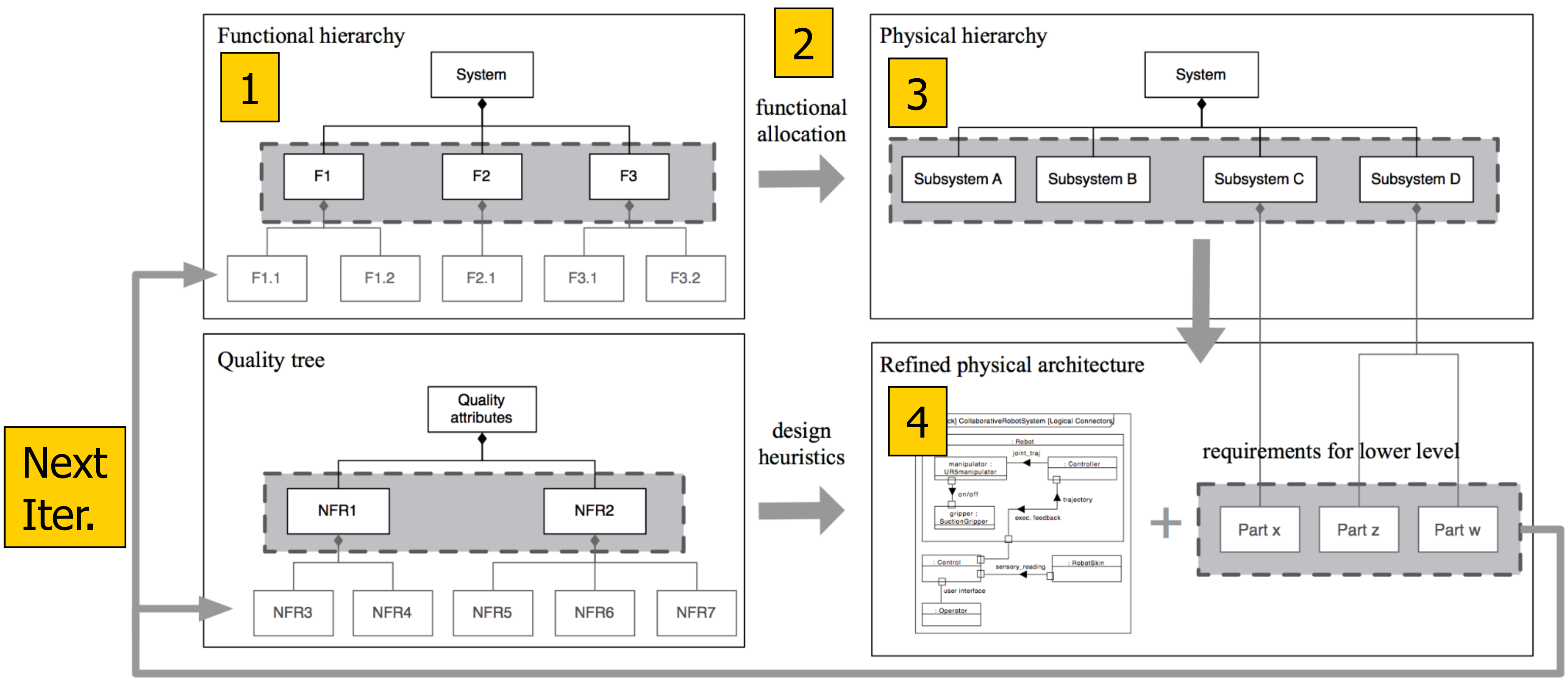
Trees, flows and bridges



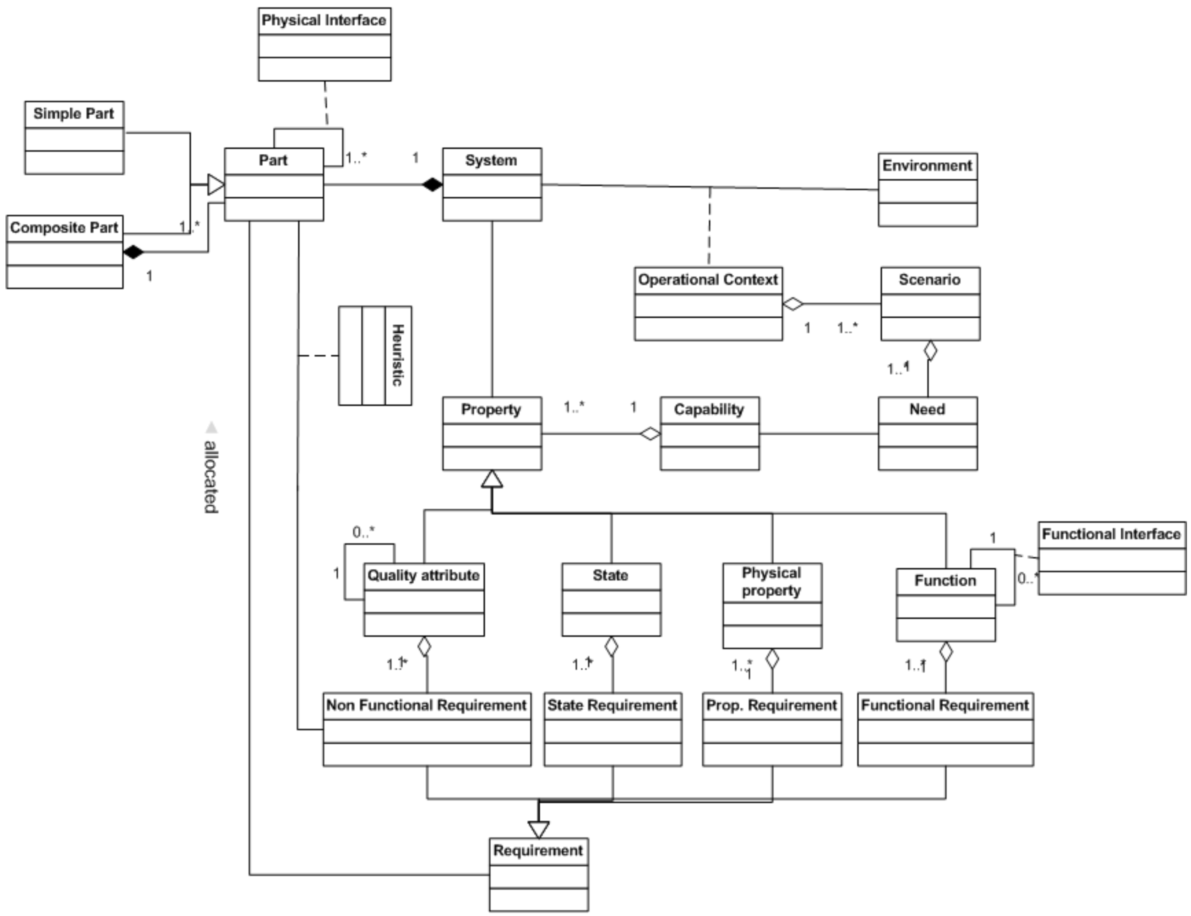
Trees, flows and bridges

- **Trees** are the diagrams used for **hierarchical** representations, for example: functional, quality models, physical and requirements diagrams.
- **Flows** are the diagrams used for representing **behavior** that is functional flows that describe system behavior represented here as SysML activity diagrams.
- **Bridges** are the way to **cross over the semantic gap** between two engineering areas. For example "**heuristics**" bridge nonfunctional requirements and the system refined architecture. The "**domain model**" bridges the system architecture and the software architecture

The use of hierarchies



ISE&PPOOA conceptual model



ISE&PPOOA conceptual model- brief description

The relationships represented in the previous slide are summarized below:

- A **system** has **parts** that may be either simple or composite parts. A system interacts with the **environment**. These interactions are described by an **operational context** that models the interactions as a set of **scenarios**

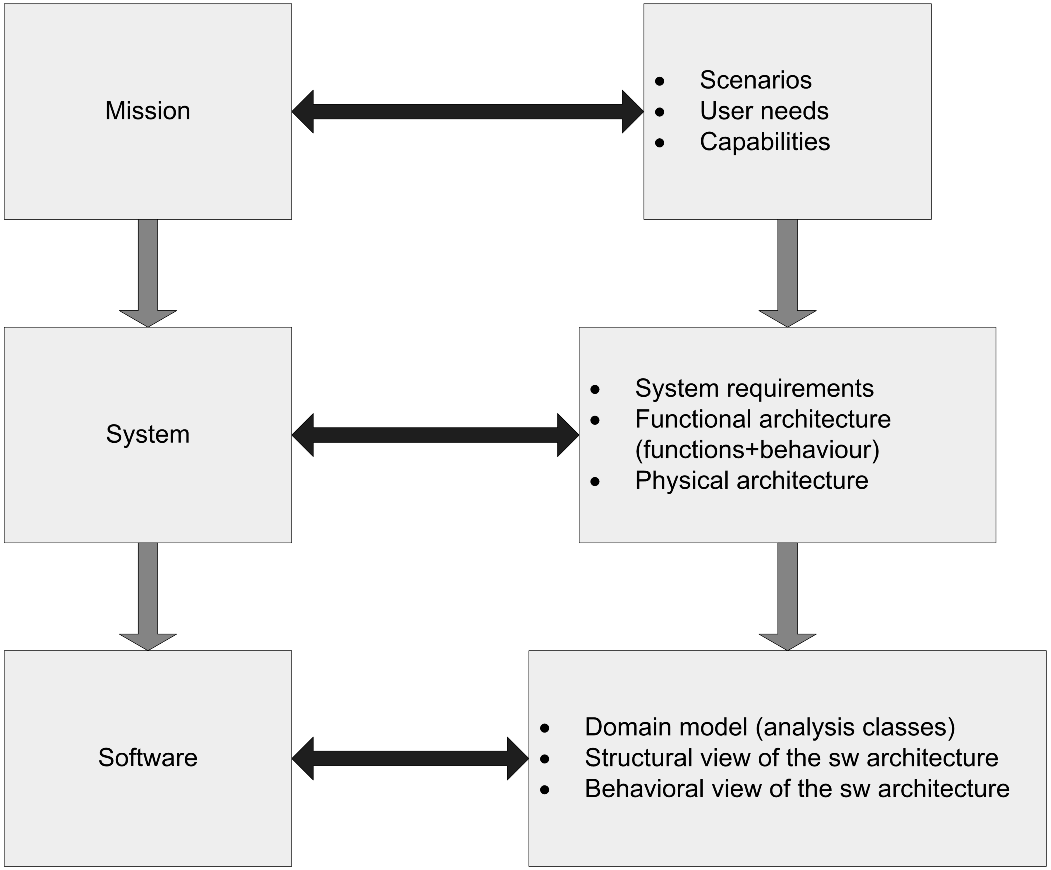
- Based on the operational context and scenarios, the engineer translates the set of specific **needs** into a set of **system capabilities** that should be solution independent. Each capability is a container of system properties that may be either system **quality attributes, physical properties, states or functions**

- In contrast to **functional requirements** that are allocated to system parts, **nonfunctional requirements** implementation is essentially different. Nonfunctional requirements may be met by the application of **design heuristics**. For this reason a specific association is depicted between both concepts

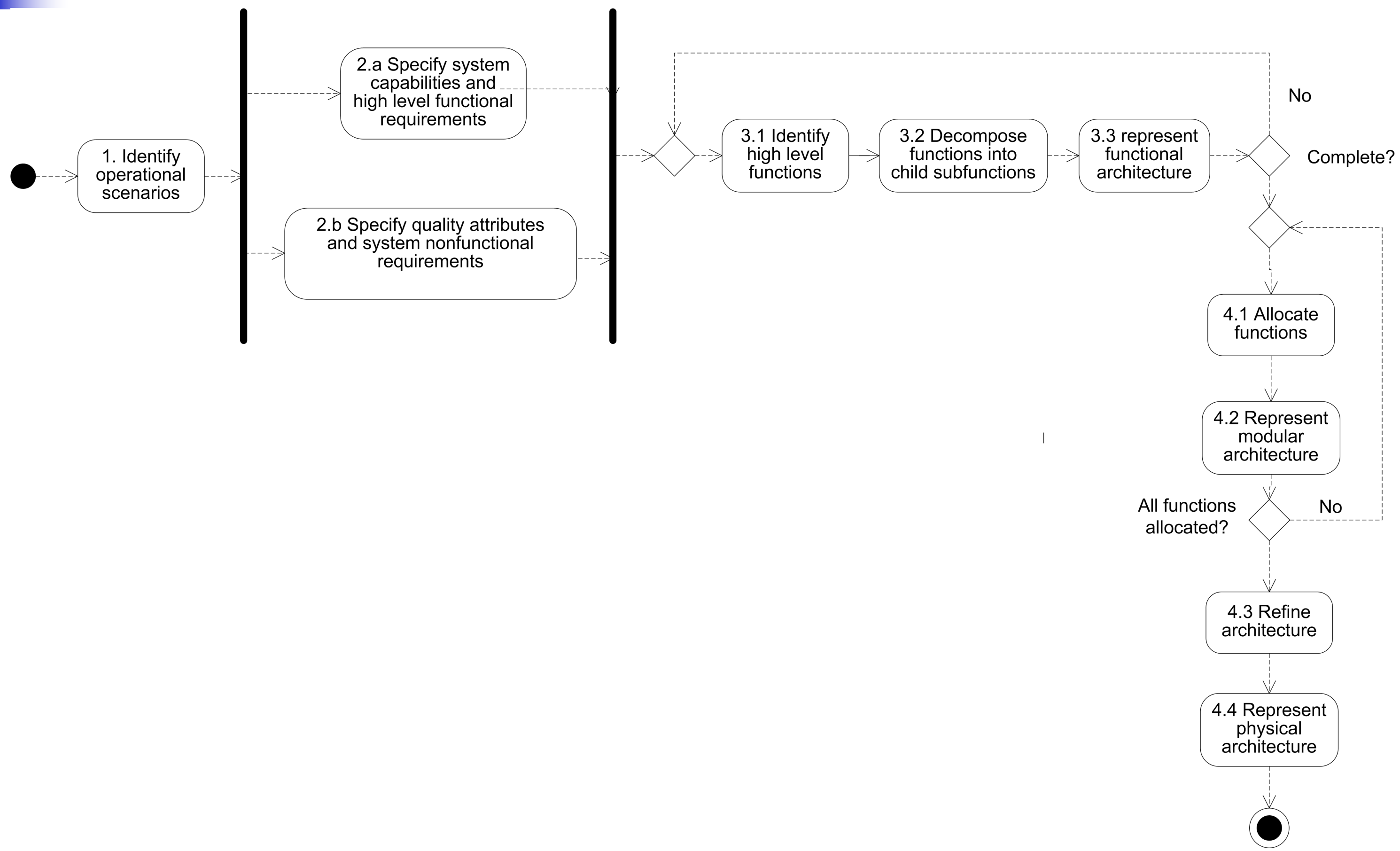
3.ISE&PPOOA dimensions and process

The three dimensions of ISE&PPOOA and the steps to apply the methodology iteratively to derive the low level requirements and create the architecture of the system

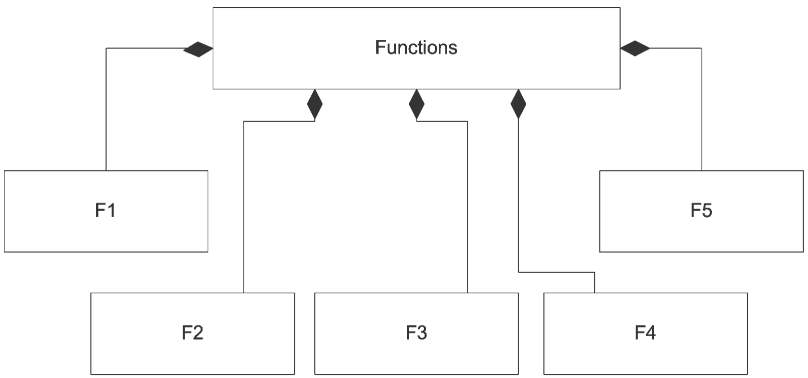
The three dimensions of ISE&PPOOA



ISE&PPOOA process



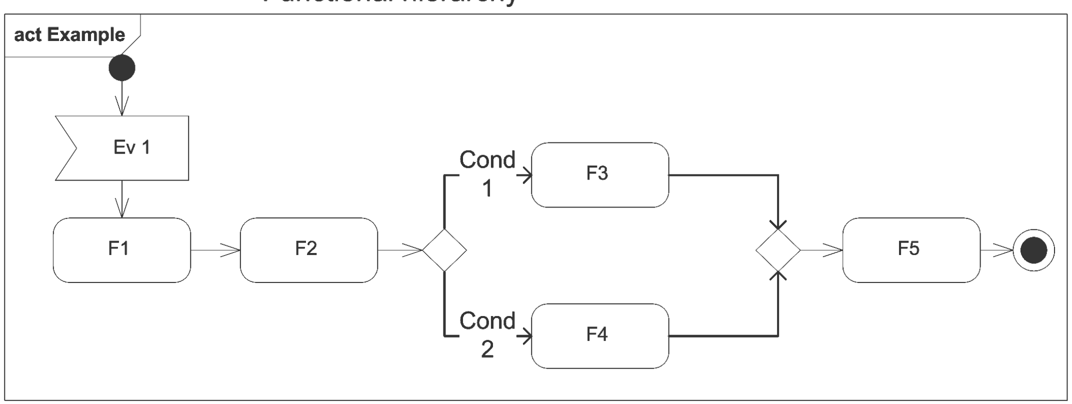
System functional architecture



Functional hierarchy

item1				item 6	
F1	item 2				
	F2	item 3			
		F3	item 4		
			F4	item 5	
				F5	item 7

N square chart

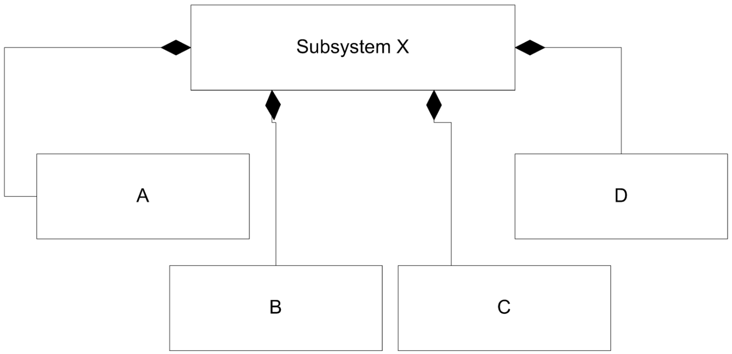


Functional flow

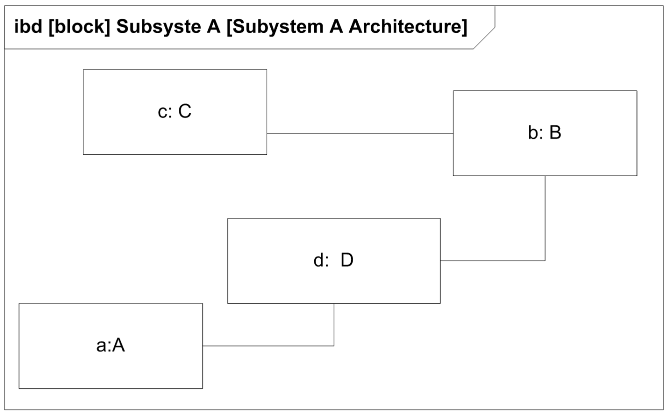
Functions description
(text and tabular format)

Functions description

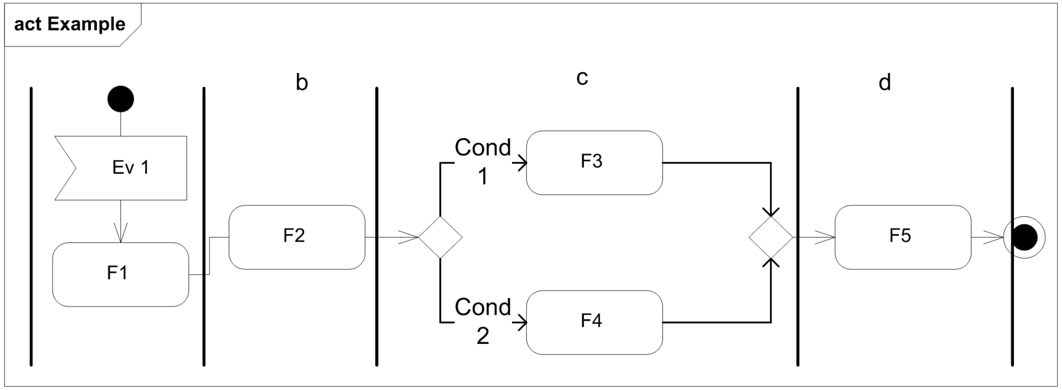
System physical architecture



Physical hierarchy



IBD



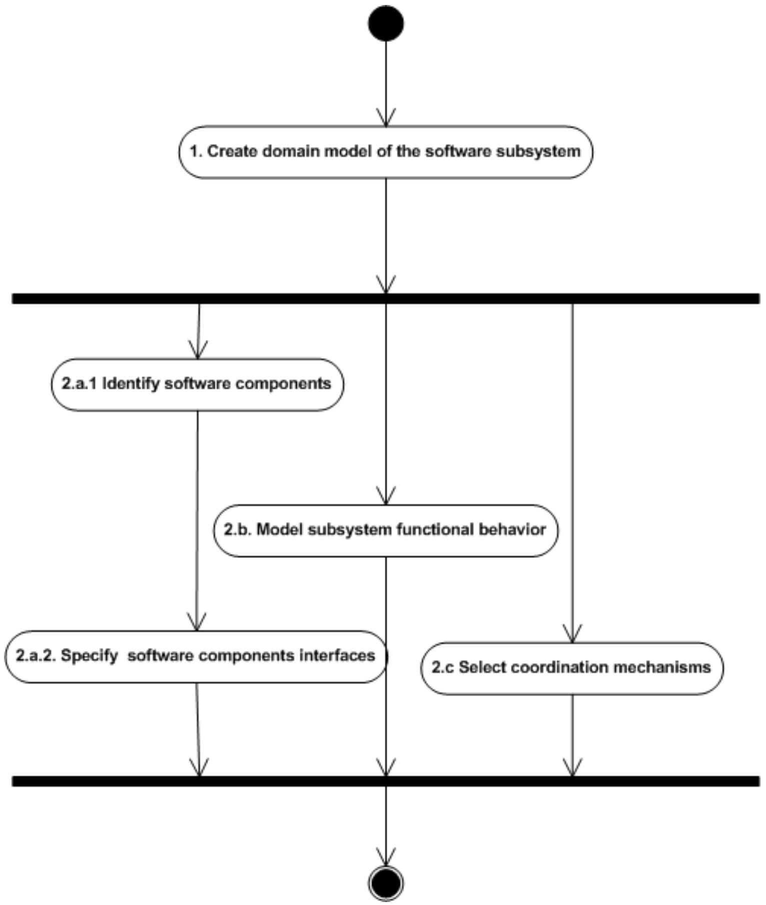
Allocated Functional Flow

Parts Descriptions
(text and tabular format)

4. The importance of the domain model for the software intensive subsystems

The integration between the systems engineering modeling process **ISE** and the software architecture modeling process (here **PPOOA**) is achieved by using **domain modeling** and **responsibility** driven software analysis approach supported by **CRC** (Class Responsibility Collaborator) **cards**, a technique proposed by the object oriented community.

PPOOA process



Domain model

- A domain model is described using more formalism than textual descriptions, for example UML class diagrams.
- The domain model is the result of a **domain analysis**
  - **Classes** in the domain model represent **concepts or terms** derived from use cases and the functionalities to be implemented by software.
  - So, classes are abstractions of physical and non-physical entities for example **things**, **events, roles, descriptions**…
  - The domain model represents the **relations** between classes mainly generalization, specialization, "is part of", "is member of" and associations.
- A domain model is an essential input when the subsystem is shaped in software architecture, design and implementation

CRC cards

- **CRC cards** are index cards, one for each domain model class, upon which the responsibilities of the class are briefly documented and a list of classes collaborated with to achieve those responsibilities

Class name Software class identity
Class responsibilities • What the class knows? • What the class does?
Class collaboration • Other classes the class is collaborating to achieve its responsibilities

Summarizing

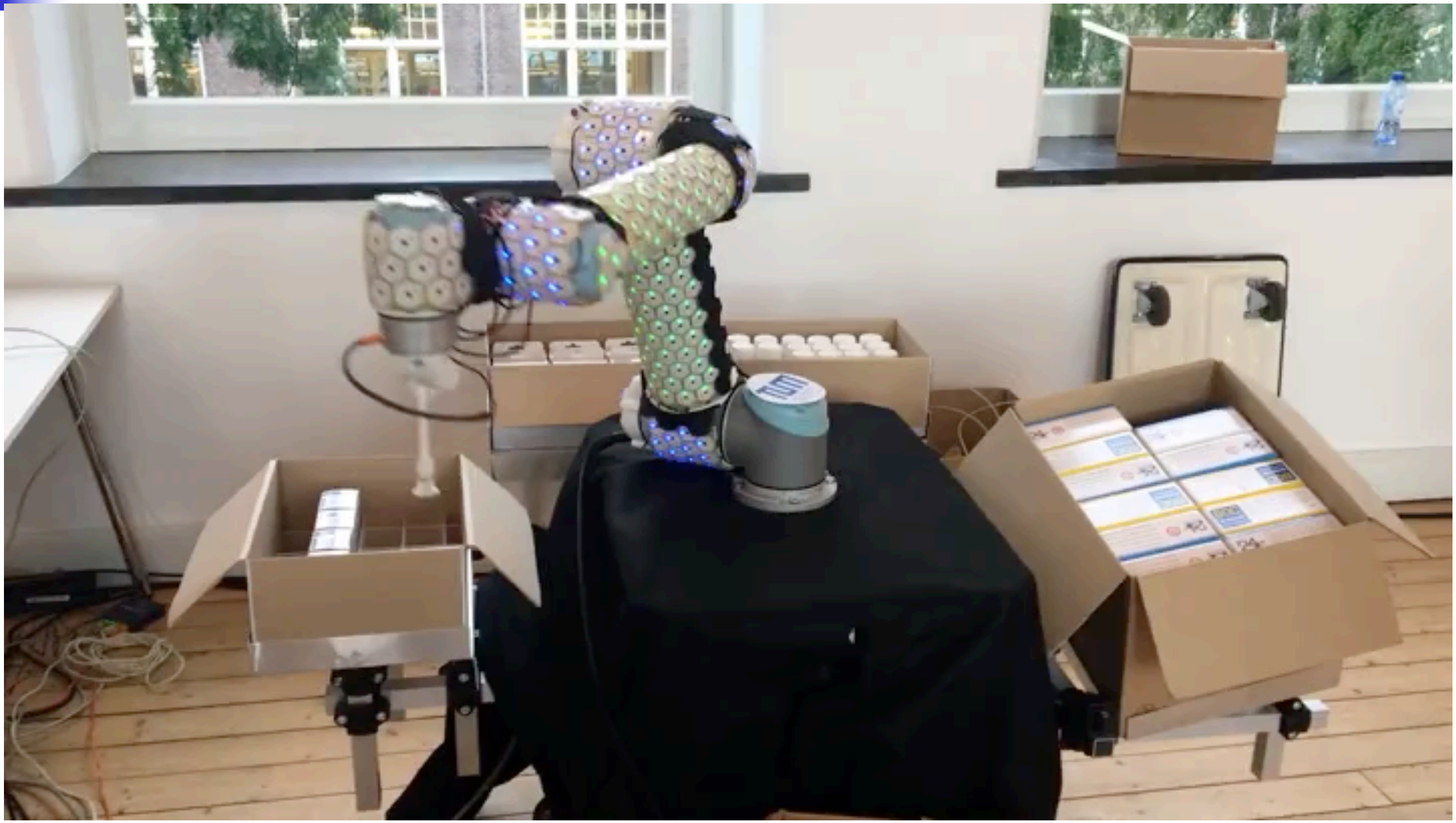
- A mere physical system (traditional SE) and a software system are different
  - Software components can be created and destroyed
  - **Software does not meet Physics laws** of mass, energy and momentum conservation
  - **Software components are abstractions of physical and non-physical entities** for example events, roles, descriptions…
- For complex and software intensive systems we need "**best practices+methodologies"** that bridge the system and software semantic gap. Here we have proposed: **domain modeling+ ISE&PPOOA**

5. **Example**: Robotic Systems

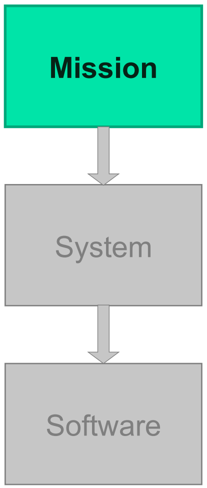
Challenges in robotics for Systems and Software engineering

Team Delft: winner of the Amazon Robotics Challenge 2016

Re-engineering a Collaborative robot

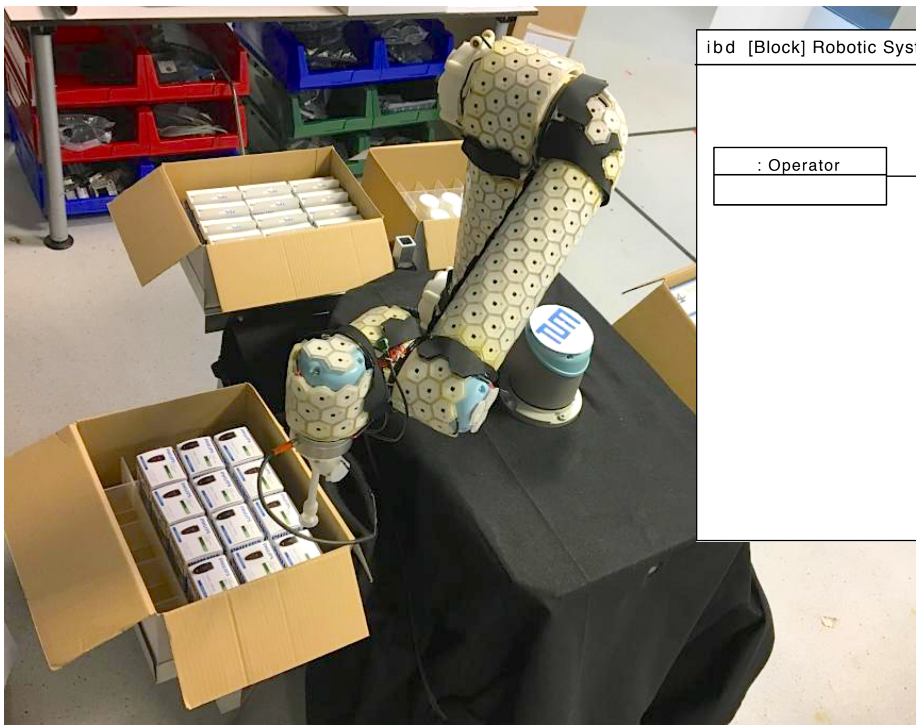


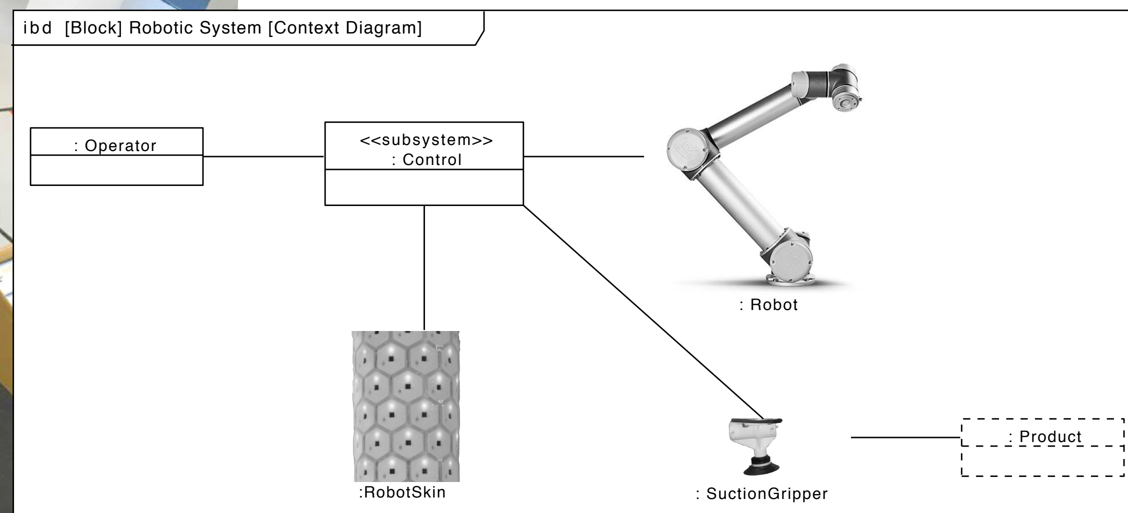
Collaborative robot: **Mission**



Scenarios, needs and capabilities

Scenarios: Context





Operational Scenarios

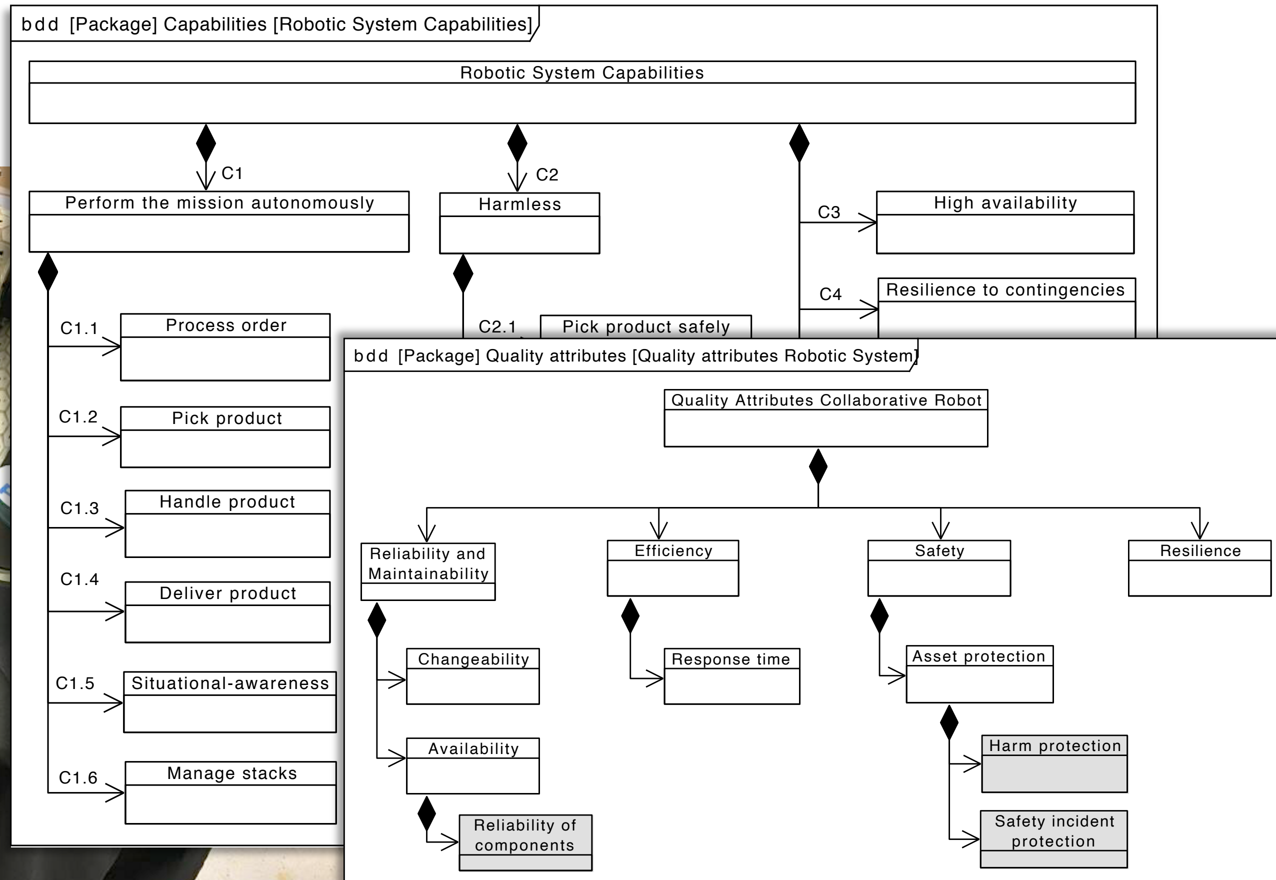
	Scenario	Description
Deployment	S1 System configuration and calibration	The operator installs the robot arm and the product containers in the frames, and moves the robot arm to the reference pose for each container, so all end-effector poses are calibrated off-line for the robot to be able to reach all the products.
Deployment	S2 Configure new product	The operator adds a stack of a new product type, possibly of different dimensions (within a limited variation range), or replaces an existing type with the new one, and informs the system through the operator interface. The system is capable of processing order that include the new product type.
Operation	S3 System start	The operator powers up all subsystems. The robot arm calibrates its joints.
Operation	S4 Manage order	The system receives an order request consisting of several products of one or more product types, and delivers the order in the bin by autonomously handling the products.
Operation	S5 Pick product	The robot moves the gripper to the container of the next product in the order, grasp an available product, and retreats from the container holding it.
Operation	S6 Deliver product	The robot delivers the product it is holding with the gripper in the delivery bin.
Operation	S7 Refill product	When one of the product stacks is empty, the robot notifies the operator and moves to a configuration to allow the operator to replace the container with one filled with products.
Operation	S8 Shut down	The operator shutdowns the system through the user interface. As a result, the status of the containers is stored in memory and the robot arm moves to the stand-by position
Operation	S9 Emergency stop	Upon an anomalous behavior, the emergency stop of the robot arm is activated, stopping any current motion, and this is notified to the operator.

Scenario description and operational needs

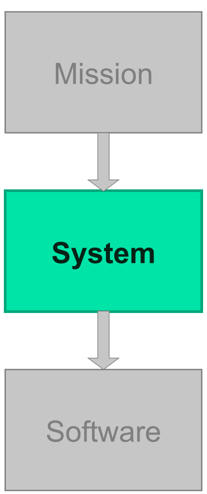
Operational Scenario:	S5 Pick product
Preconditions	• System initialized. • There is an order accepted and being processed. • Non-empty containers of products.
Triggering event	• Order request. The operator enters the order description.
Description	The robot arm performs the actions required to pick a product from the stack. The system uses the skin sensing information to avoid any collisions while moving.
Postconditions	• Robot is holding a product. • The number of available products has decreased by one unit.

Operational needs for S5 "Pick product"	
ON 2_1	The system shall be able pick products of cylinder or box shape, with a maximum dimension of 20 cm, offering a top flat surface available for suction.
ON 2_2	The system shall pick the products from container boxes that have an opening at the top of maximum dimensions 0.6×0.6m, stacked in rows and standing in vertical orientation. The maximum depth of the containers is 0.3m.
ON 2_3	The system shall complete the pick task in less than 2 seconds if there are no potential collisions while performing.
…	…

Capabilities and quality attributes

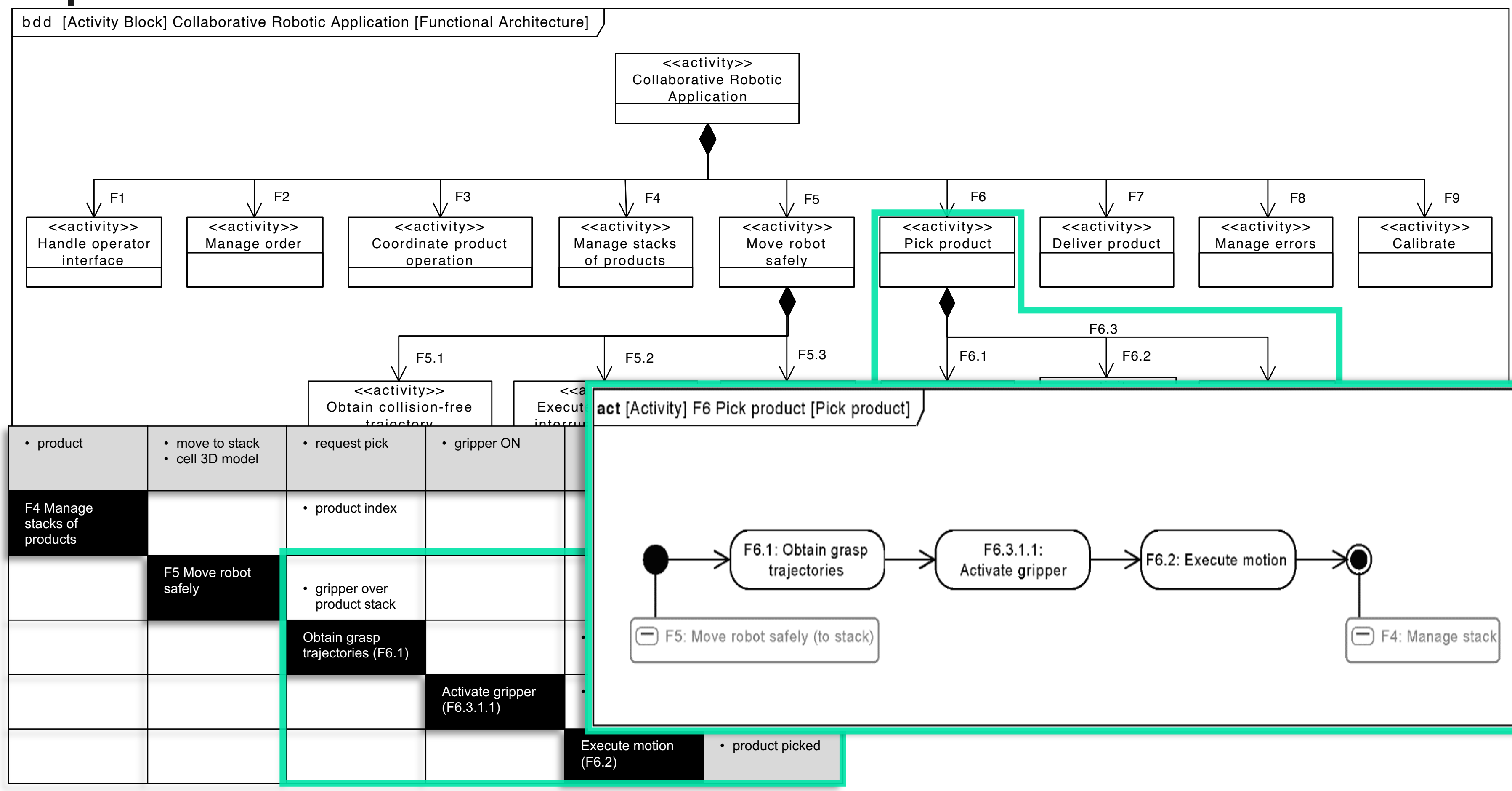


Collaborative robot: **System**

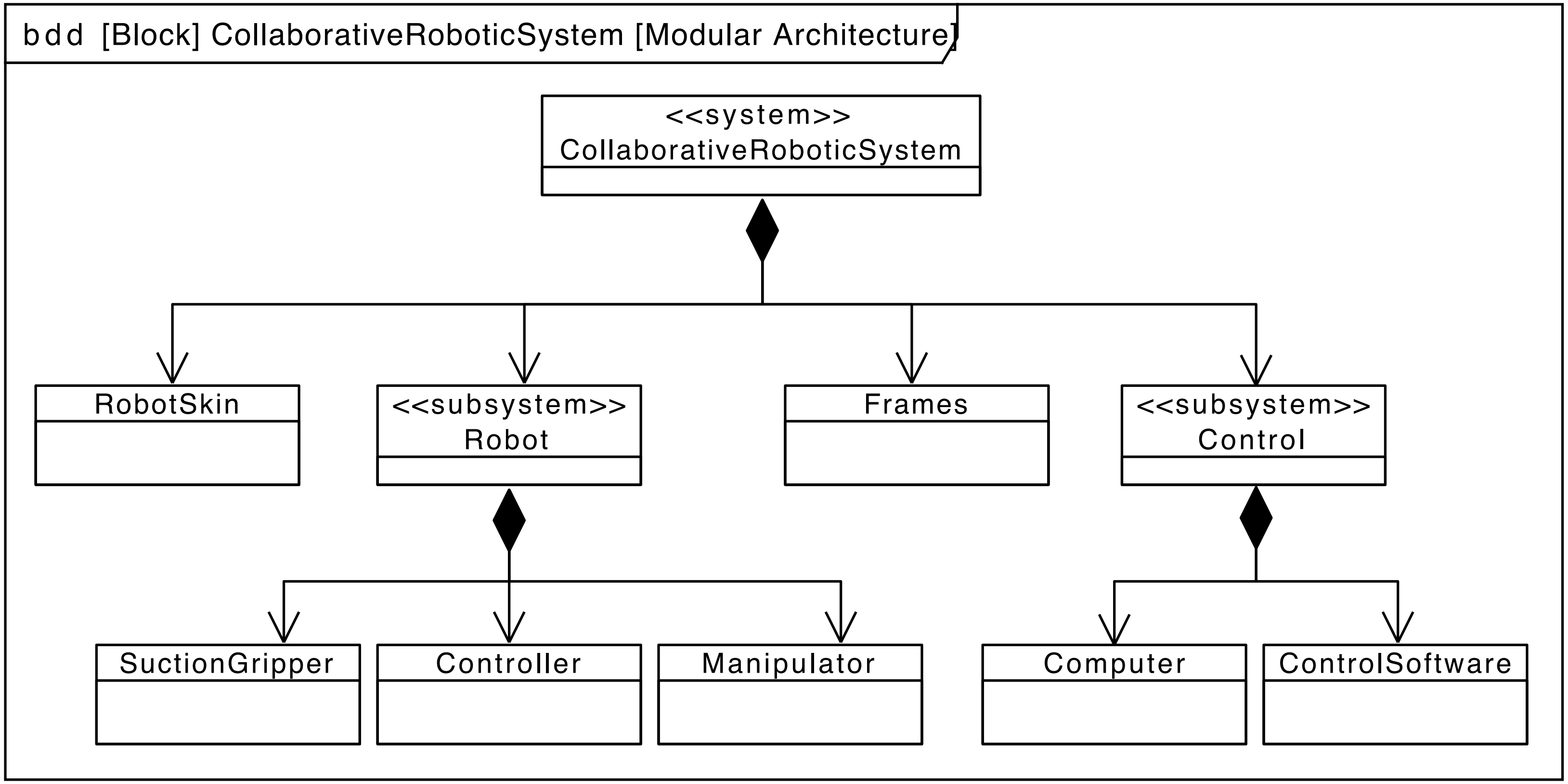


Functional architecture, (requirements) and physical architecture

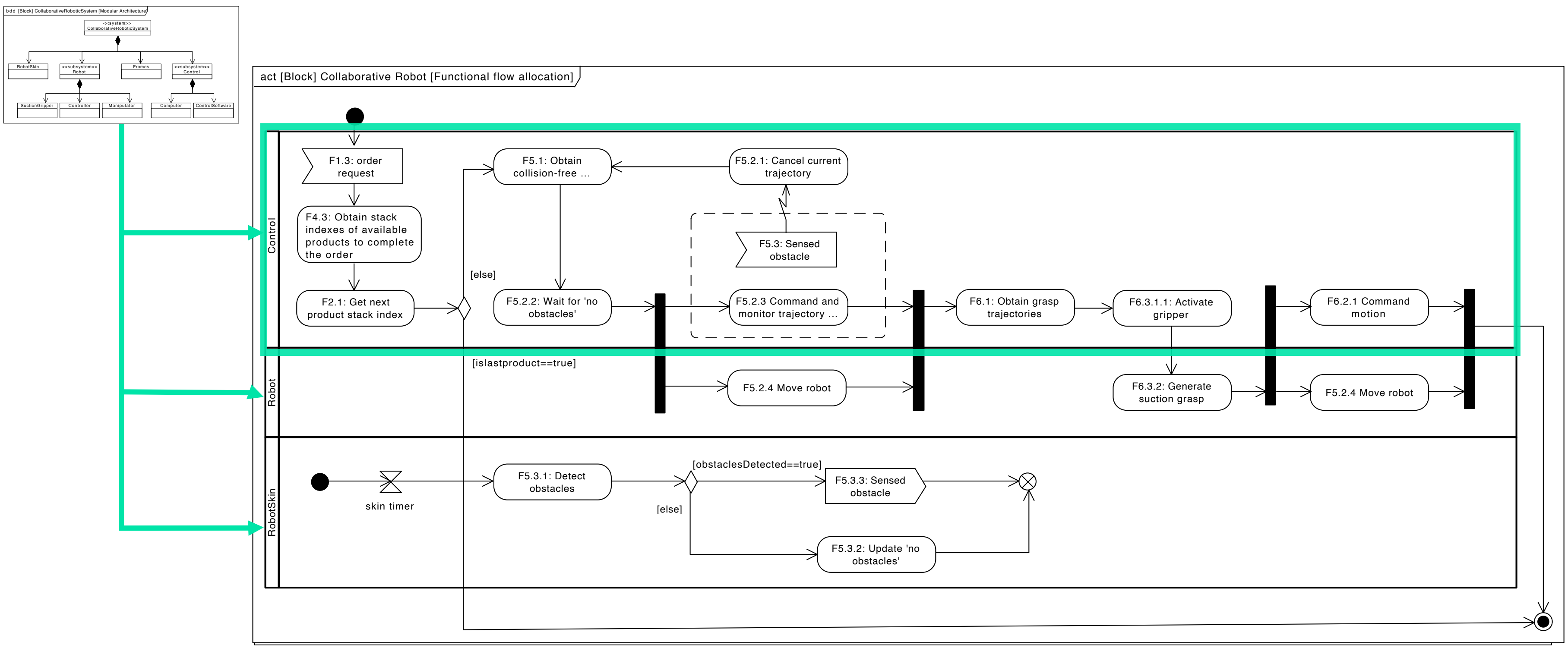
Functional Architecture



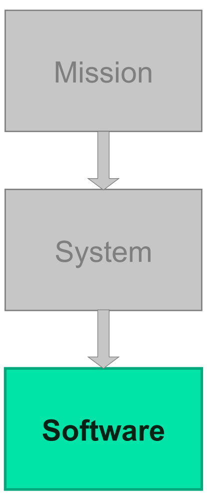
Physical architecture



Functional flow allocation

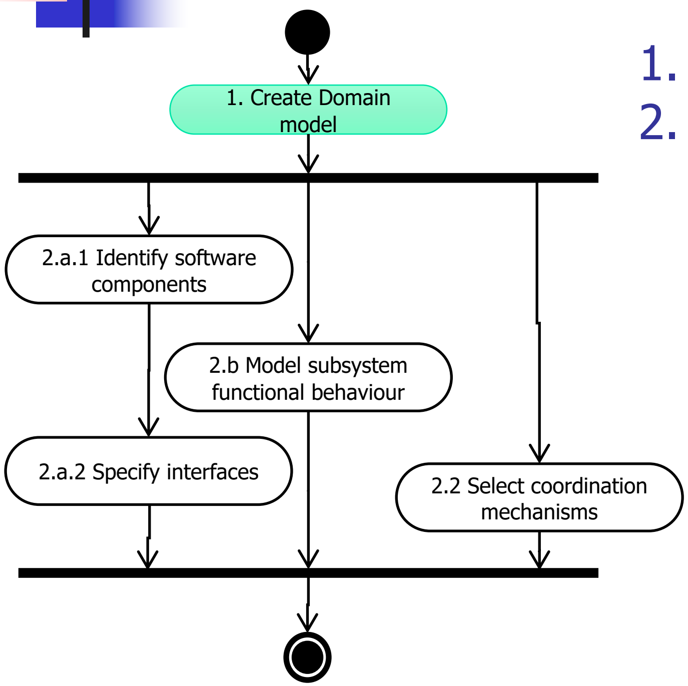


Collaborative robot: **Software**



Domain model, structural view, behavioral view

Domain model



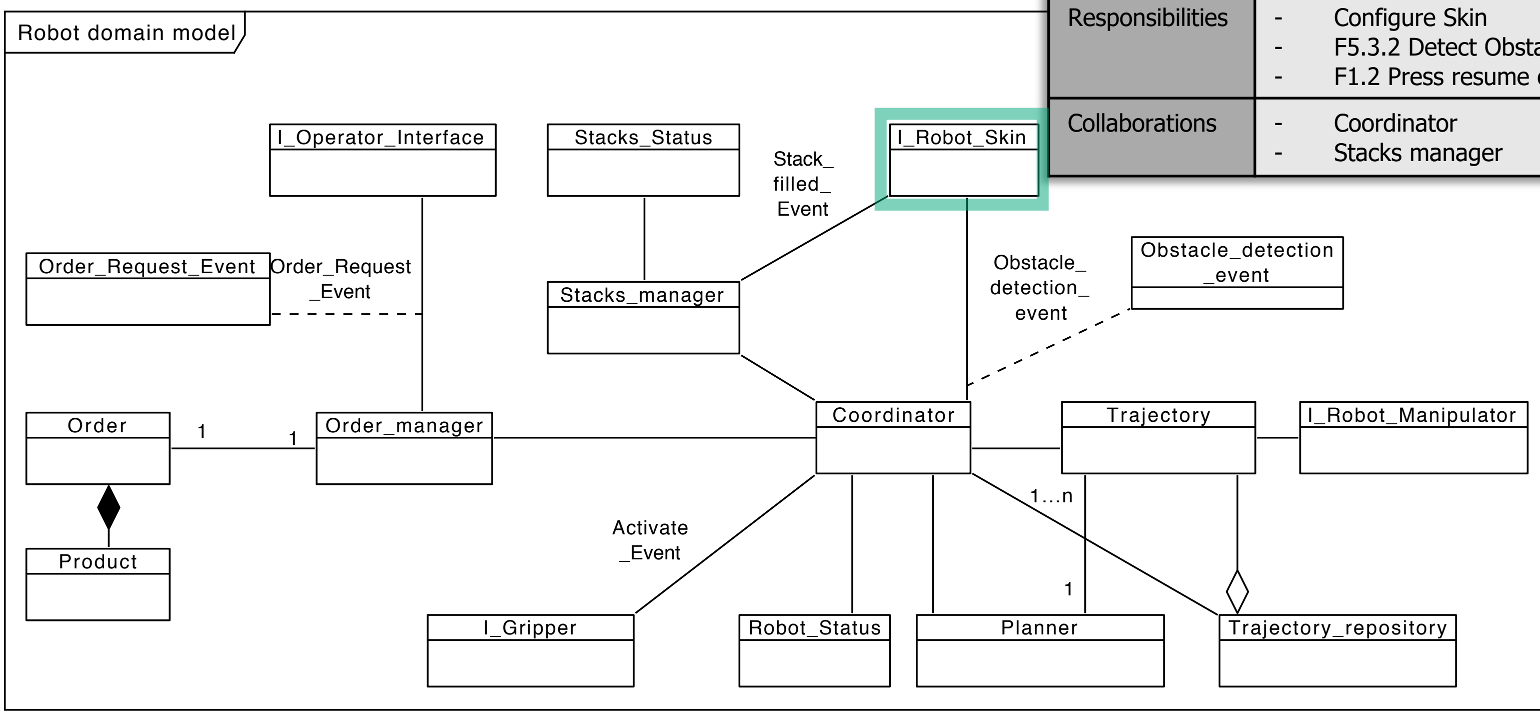
1. Identify **functions** allocated to **software**
2. the **nouns** in their description

	RobotSkin	Gripper	Robot	Control	Franes
F1 Handle **Operator Interface**	X			X	
F2 Manage **Order**				X	
F3 Coordinate **product** operation				X	
F4 Manage **stacks** of products				X	
F5.1 Obtain collision-free **trajectory**			X	X	
F5.3.1 Sense distance to robot arm	X				
F5.2 Execute and control interruptible **trajectory**				X	
F5.3.2 Detect **obstacle**				X	
F6.1 Obtain grasp **trajectories**				X	
F6.2 Execute uninterruptible **trajectory**			X	X	
F6.3.1 Control **grasp**		X	X	X	
F7 Deliver **product**				X	
F8 Manage **errors**				X	X

Domain model

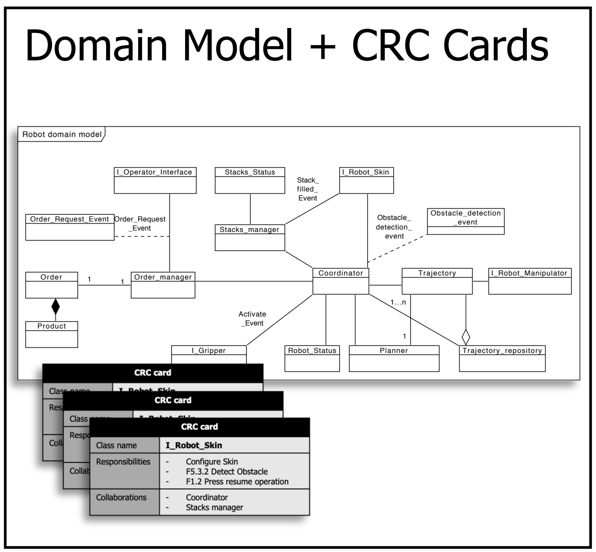
- **Nouns** are **classes** in the domain model.
- Verbs are relations between the classes.
- **CRC cards** are used to define the responsibilities for each class of the domain model.

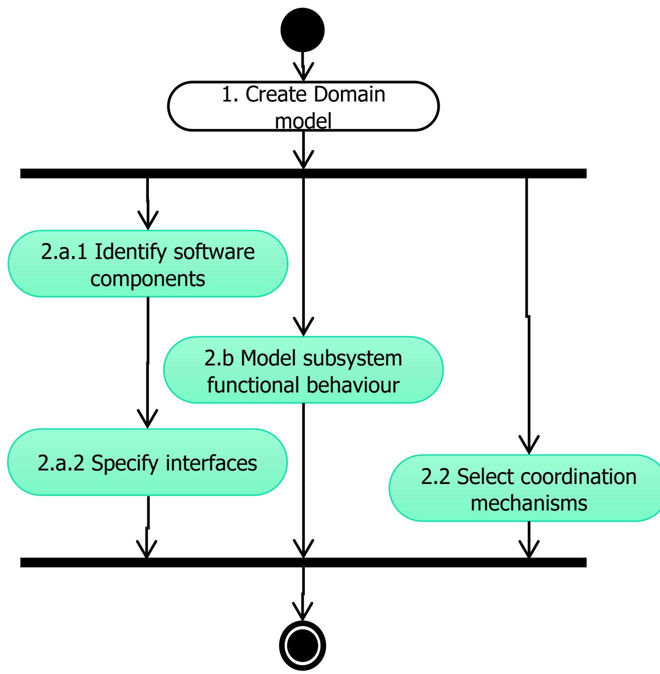
CRC card	
Class name	**I_Robot_Skin**
Responsibilities	- Configure Skin - F5.3.2 Detect Obstacle - F1.2 Press resume operation
Collaborations	- Coordinator - Stacks manager

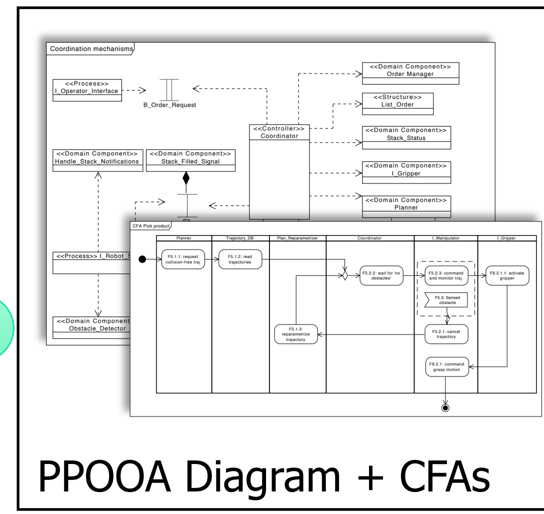


PPOOA architecture

- Refining process
- Use of heuristics

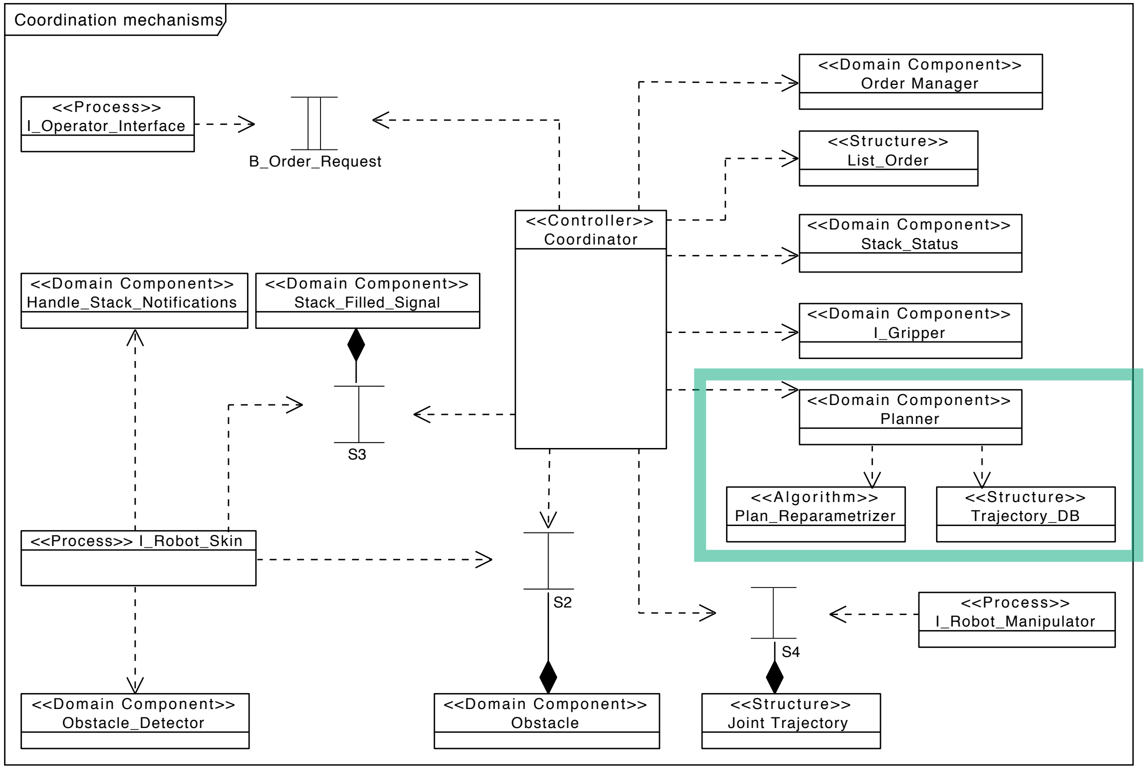






Structural View

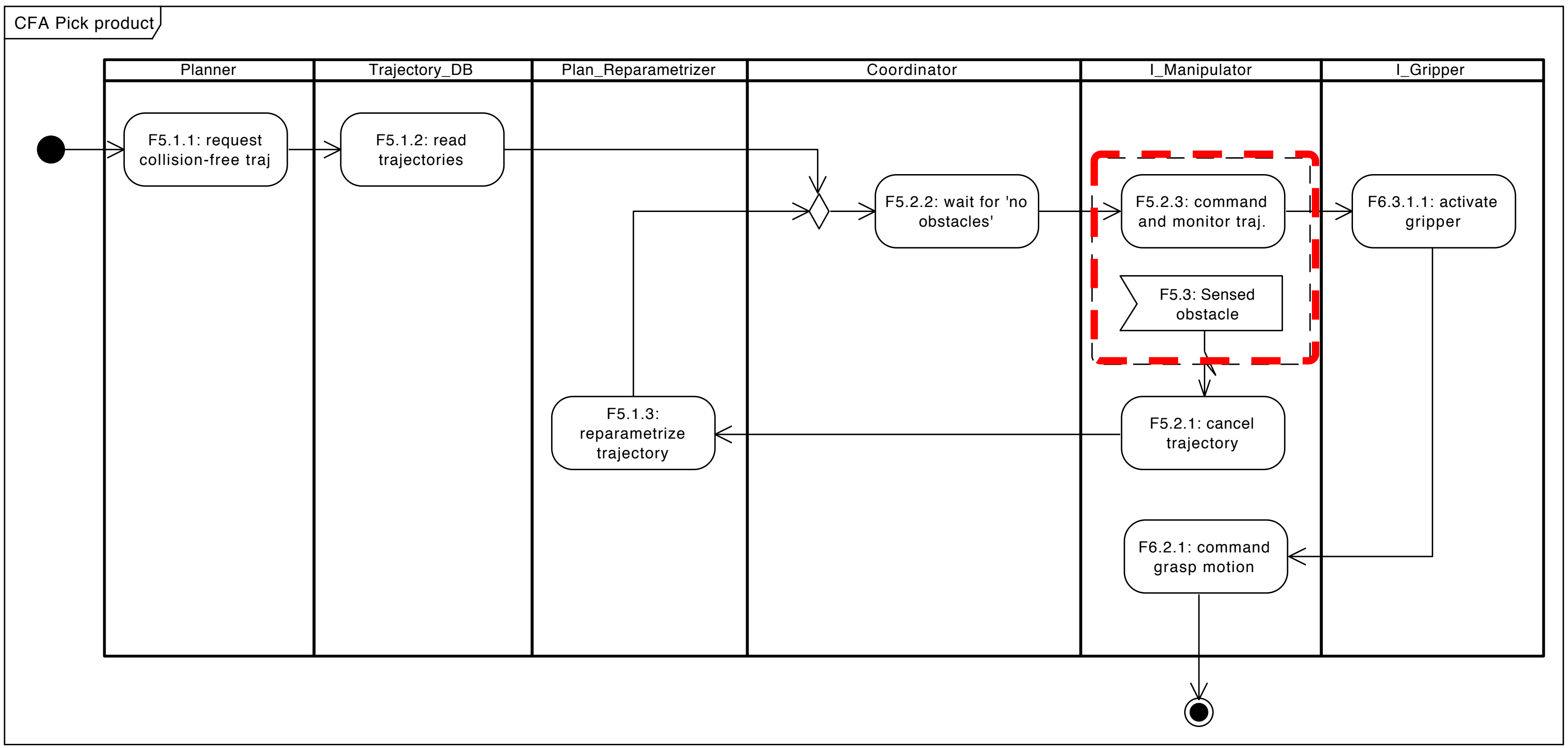
- **Heuristic - Safety:** Avoid Non-deterministic Behavior
  "Solution based on a database with trajectories obtained offline is used to generate the motions to retrieve products."



Behavioral View

- **Heuristic - Safety:** Avoid Non-deterministic Behavior

  "Interruptible regions only during motion execution in the shared workspace"

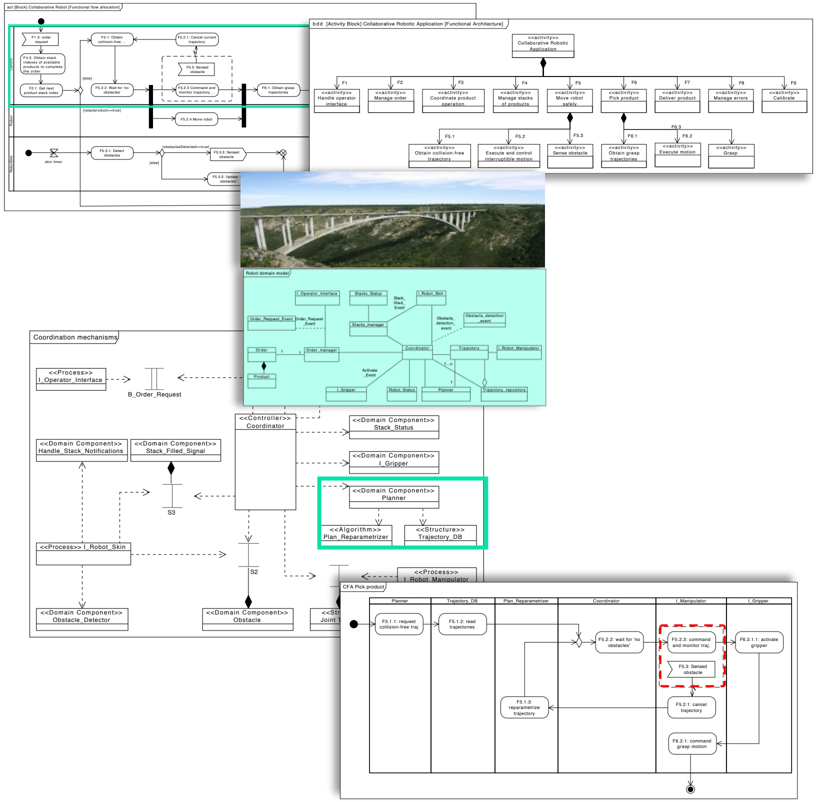


Conclusion: ISE&PPOOA for robotics

- ISE&PPOOA provides a method to formalize and **use behavior models** to obtain the system **architecture**.

- **Domain model** allows to **bridge** the **system** and the **software** architectures.

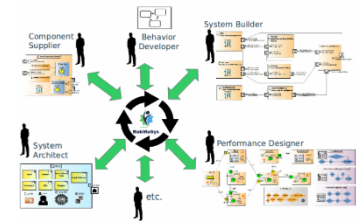
- **Quality attributes** and **heuristics** implementing them allow to reuse known solutions to **obtain** and **refine** the **architecture.**



TU Delft research on MBSE for robotics

- Composeable models and software in robotics

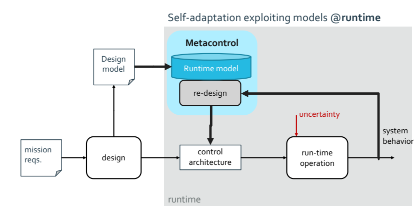
RobMoSys
https://robmosys.eu



© RobMoSys https://robmosys.eu/wiki/_media/general_principles:ecosystem:roles-ecosystem.png

- Metacontrol:
  ISE functional model
  for self-architecting

MROS
TU Delft
POLITÉCNICA
Universidad Rey Juan Carlos
BOSCH
Invented for life



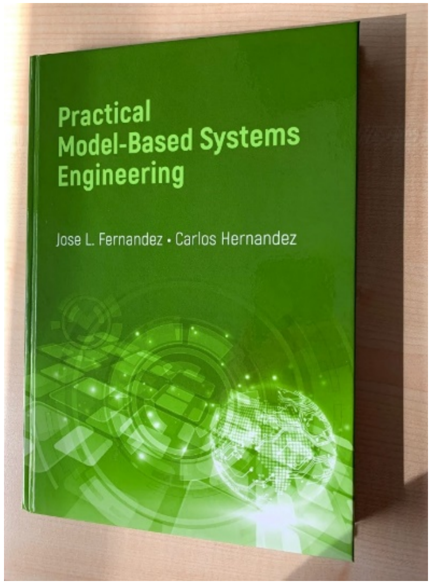
- AIRLab Deflt:
  AI and Robotics for Retail

Questions?

For more about ISE&PPOOA

Book Practical Model-Based Systems Engineering

- **Authors**: Jose L. Fernandez and Carlos Hernández
- **Publisher**: Artech House, 2019
- **ISBN-13:978-1-63081-579-0**



Other resources

- ISE&PPOOA in **OMG wiki**: https://www.omgwiki.org/MBSE/doku.php?id=mbse:ppooa
- **PPOOA**: http://www.ppooa.com.es/

This research has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) and Horizon H2020 projects:

Factory-in-a-day. Grant agreement No. 609206

ROSIN. http://rosin-project.eu Grant agreement no. 732287

RobMoSys ITP MROS Grant agreement no. 732410

Upcoming Webinars (tentative schedule)

Who	What	When
No webinar scheduled for December		

Invitations will be emailed in advance and informational updates will be placed on www.incose.org
Go to http://www.incose.org/products-and-publications/webinars for more info on the webinar series, including a way to view the last 144 Webinars and soon – this one!

Information on the webinars is now being posted in INCOSE Connect, in the INCOSE Library area, at https://connect.incose.org/Library/Webinars/Pages/INCOSE-Webinars.aspx . Joining instructions will added around two weeks before the webinar is scheduled to take place.

INCOSE International Workshop 2021

INCOSE

Website: https://www.incose.org/iw2021 Registration not open yet

INCOSE Systems Engineering Professional PDU Credit

Please note that you can claim 1PDU credit towards your Systems Engineering Professional re-certification by attending this webinar. INCOSE webinars may also apply to the PDU requirements of other organizations, depending on the subject matter

To qualify, you must have attended through at least 75% of the webinar for webinars that last less than one hour, or through 45 minutes of the webinar for webinars that last for 1 hour or longer.

You can also claim credit for previous webinars you have attended; please contact info@incose.org if you wish to know which webinars you attended and if you met the qualification requirements

With thanks to our Sponsor for 2020

INCOSE

LOCKHEED MARTIN