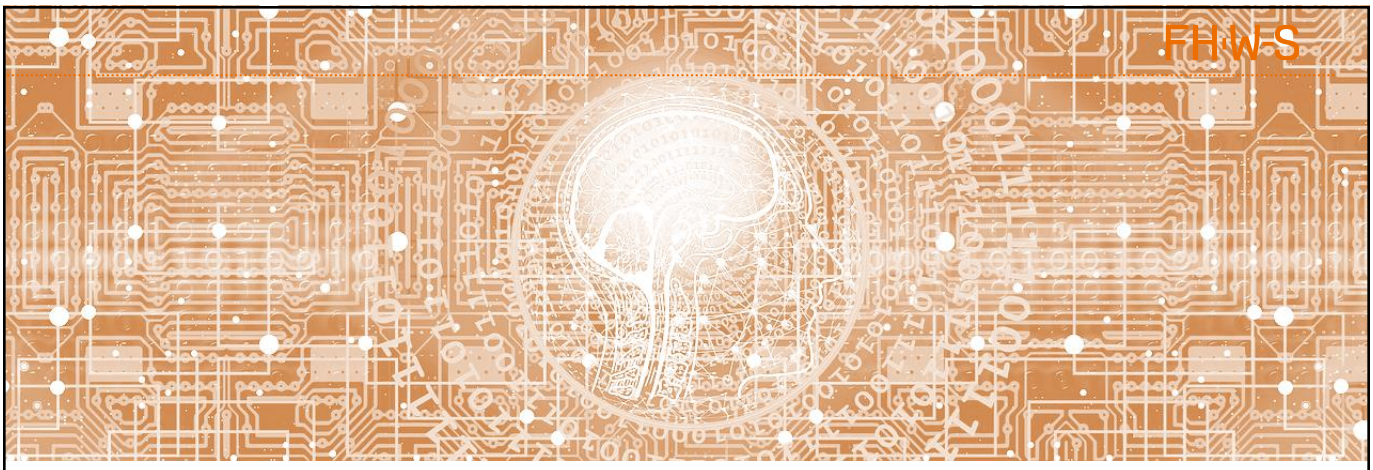


Embedded Systems and Field Buses

Prof. Dr. Marian Daun

FH·W·S University of Applied Sciences Würzburg-Schweinfurt

Agenda

- Fundamentals
- Structure of Embedded Systems
- Behavior of Embedded Systems
- Design of Embedded Systems
- Communication
- Real-time
- Collaborative Embedded Systems

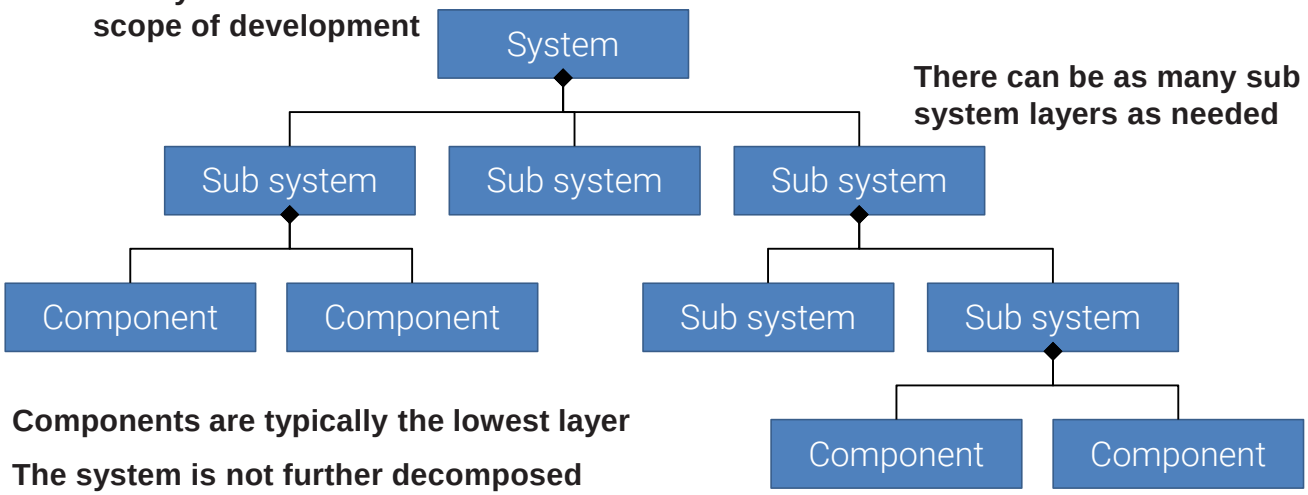
FH·W-S

Structure of Embedded Systems

Hierarchies

System Layers

The system defines the scope of development



There can be as many sub system layers as needed

Components are typically the lowest layer

The system is not further decomposed

Group Discussion

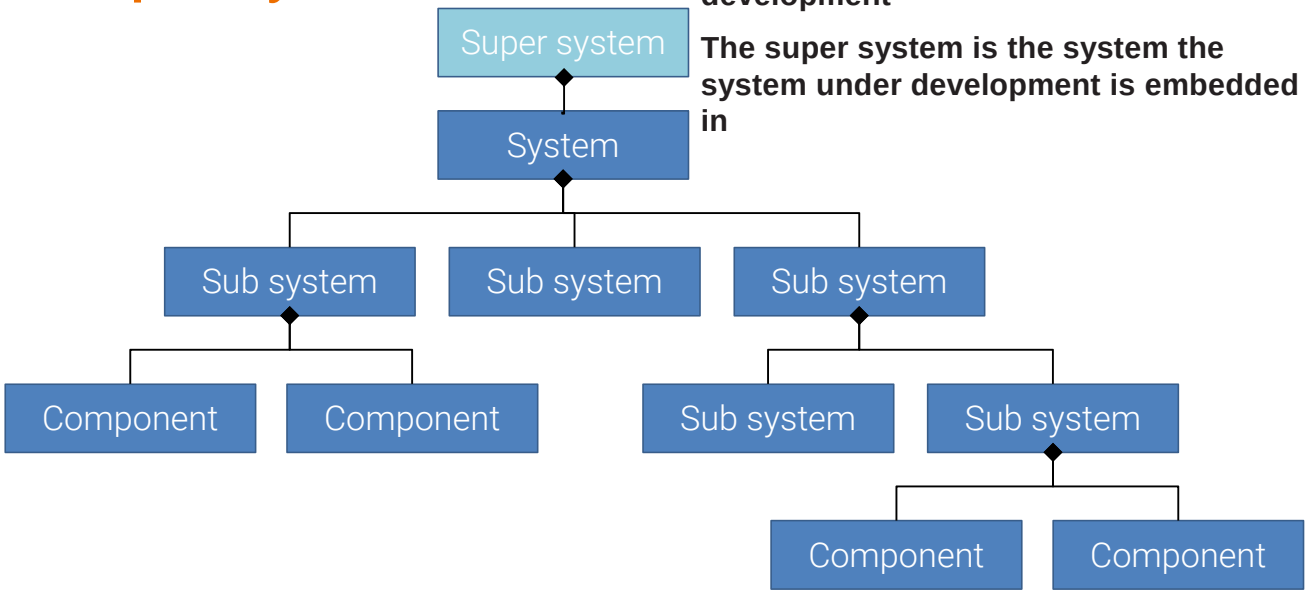
When has Development reached the Component Layer?

when the level of clarity has been reached

Super System

The super system is not the scope of development

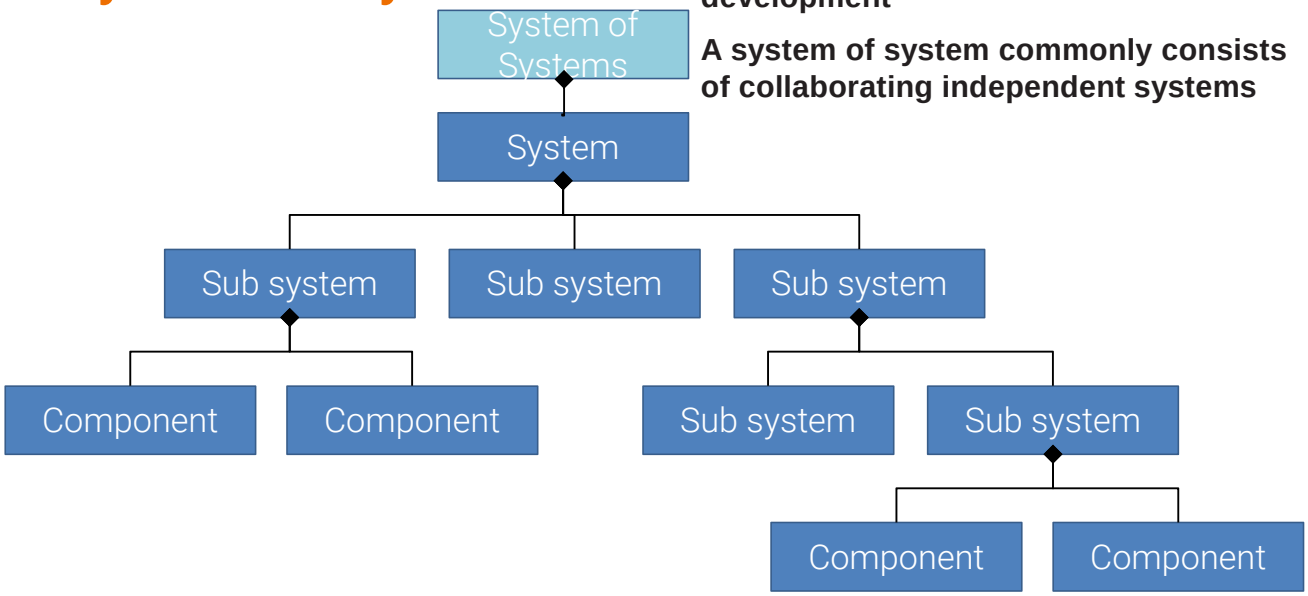
The super system is the system the system under development is embedded in



System of Systems

A system of systems can be the scope of development

A system of system commonly consists of collaborating independent systems



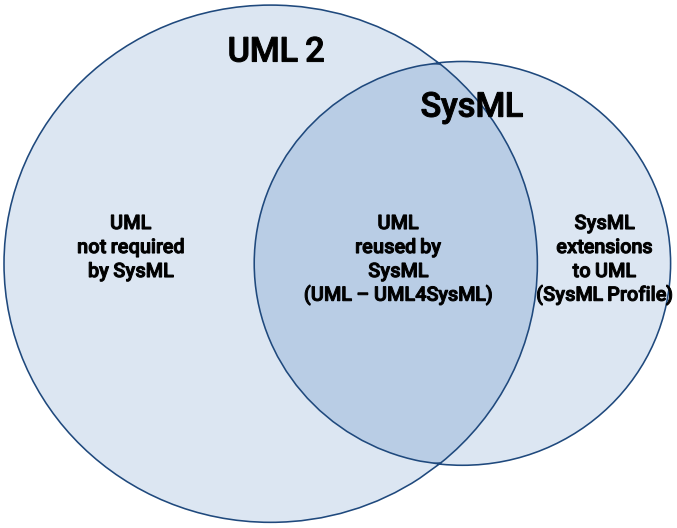
Exercise

Define the System Layers for:

- An Aircrafts
- A Car
- A Smart Factory
- Autonomous Driving

Modelling Structure with SysML Block Definition Diagrams

SysML



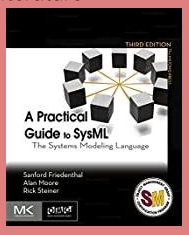
[OMG Systems Modeling Language v1.6, 2019]

Standards

https://www.omg.org/spec/SysML/

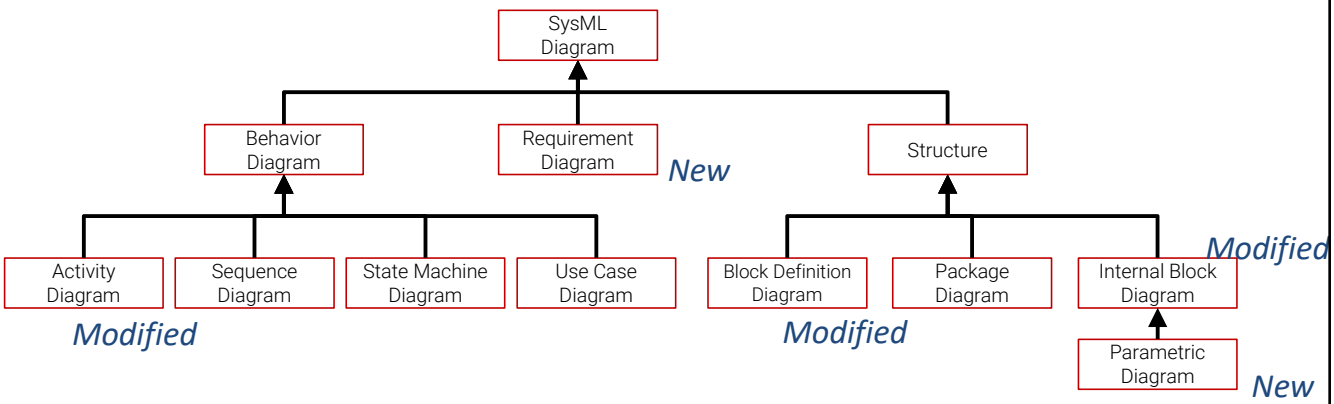
https://www.omg.org/spec/UML/

Literature



Sanford Friedenthal, Alan Moore & Rick Steiner: A Practical Guide to SysML – The Systems Modeling Language. *Morgan Kaufmann OMG Press, 2014.*

SysML Diagram Taxonomy

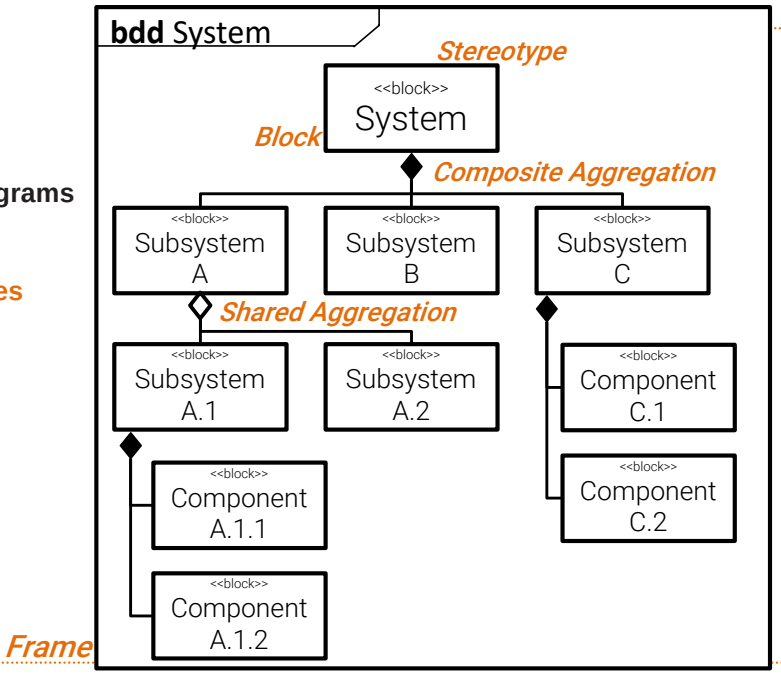


[OMG Systems Modeling Language v1.6, 2019]

Block Definition Diagram

BDDs are based on UML class diagrams

BDDs are used to model hierarchies

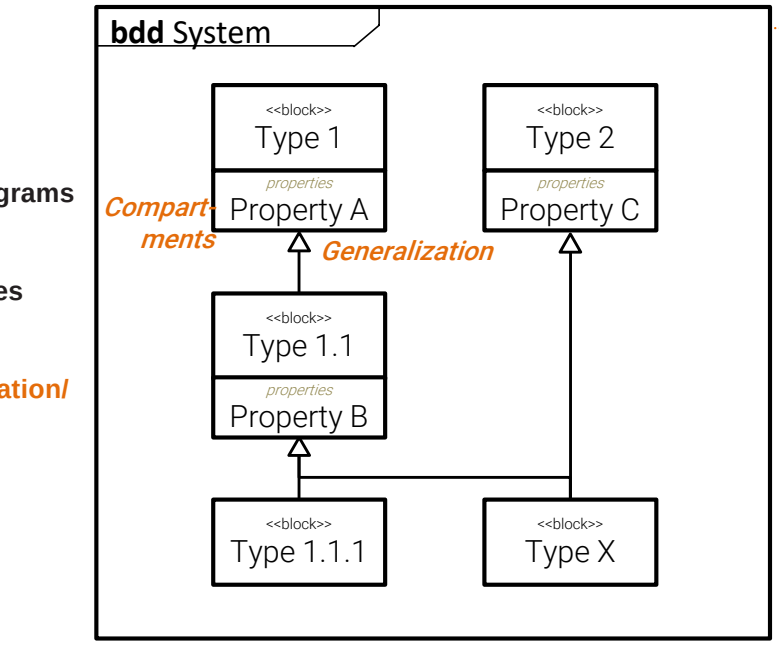


Block Definition Diagram

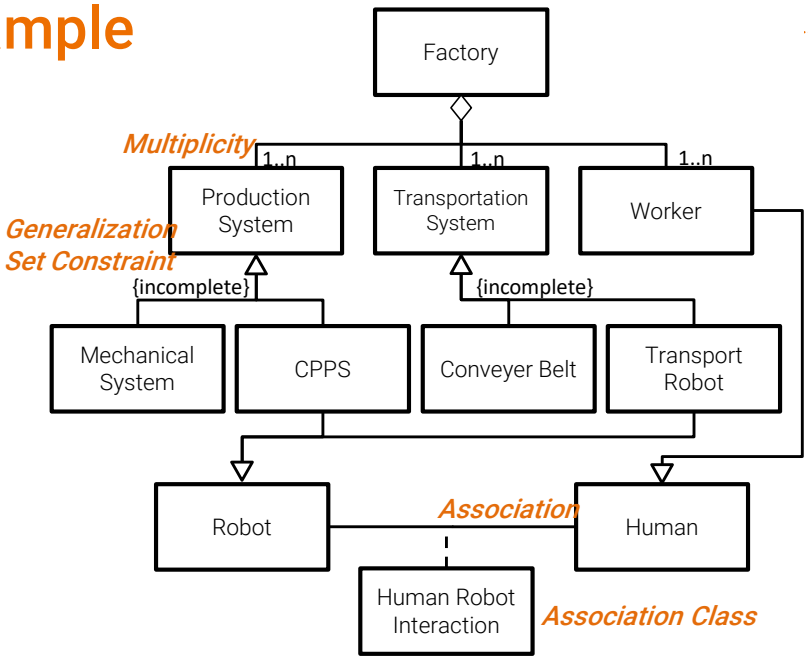
BDDs are based on UML class diagrams

BDDs are used to model hierarchies

BDDs are used to model generalization/ specialization relations



Example



Exercise

Create SysML Block Definition Diagrams for:

- An Aircrafts
- A Car
- A Smart Factory
- Autonomous Driving

Scope of Development

From the point of an original equipment manufacturer a car might be the system under development. However, it is uncommon to refer to a car as the system. Typically, the system under development is a system embedded in the car.

An aircraft, a car, a smart factory are thus super systems.
For autonomous driving a platoon is a system of systems.

Exercise

Create SysML Block Definition Diagrams for:

- An embedded system of an Aircraft Aircraft Environmental Control System
- An embedded system of a Car
- A cyber-physical production system within a Smart Factory
- The transport system of a smart factory as a System of Systems

Group Discussion

What do we notice when comparing the model of the super system with the model of the embedded system?

Context

Context is Important

Separation of **Context** and **System** allows to differentiate between the scope of development (i.e. **what can be changed**) and the **constraining** environment

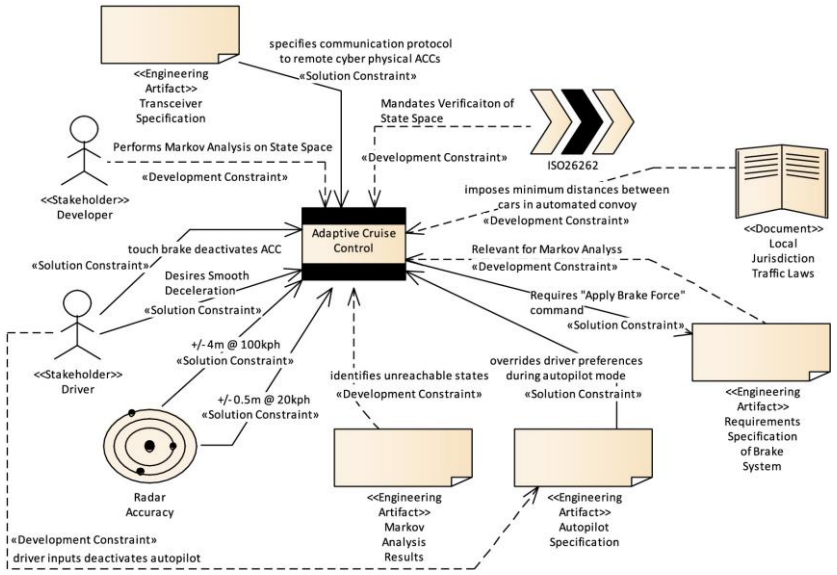
- Context is main source of requirements and rationales
- Context is source of safety and security risks/threats

Context changes can severely affect the system

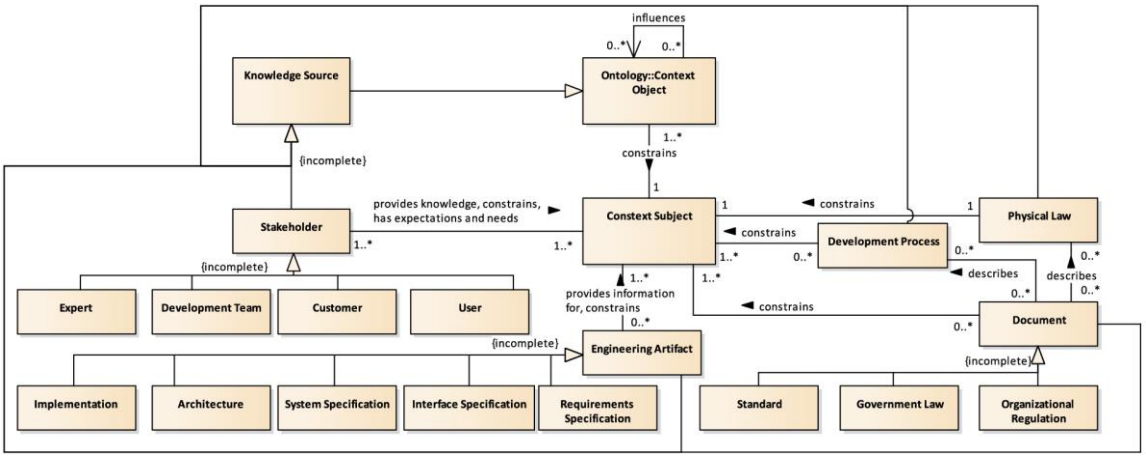
- Functionality
- Correctness
- Safety
- Security

Context of Knowledge

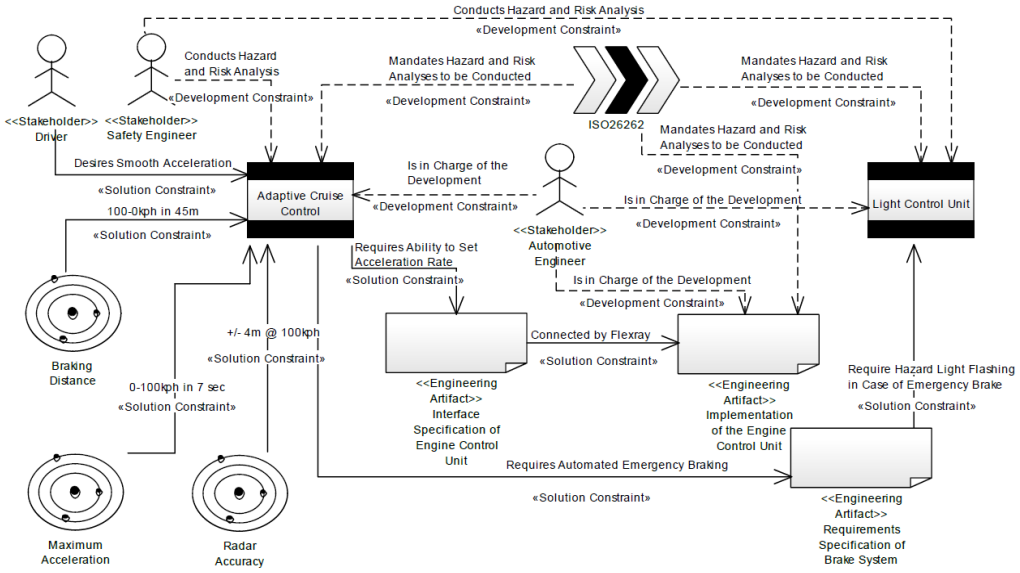
The context of knowledge defines the major requirements sources and important knowledge bearers for the development of a system.



Context of Knowledge Meta Model



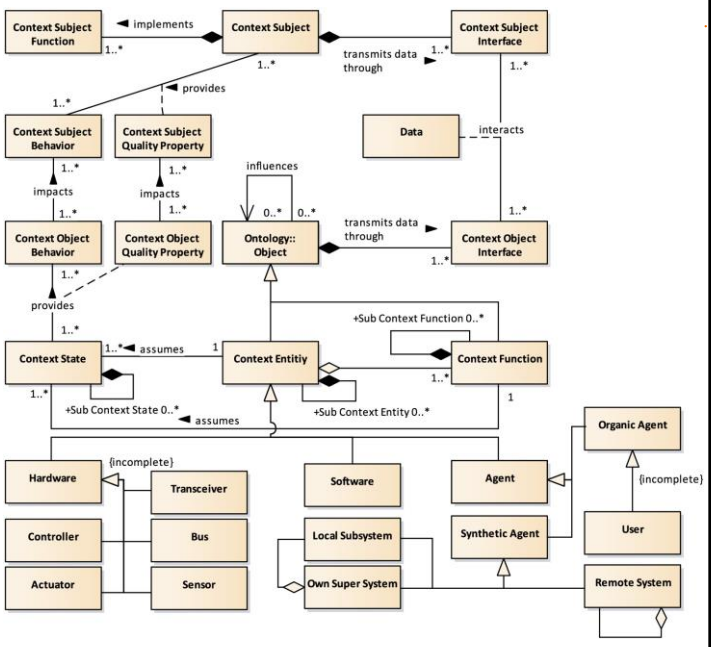
Combining different Contexts



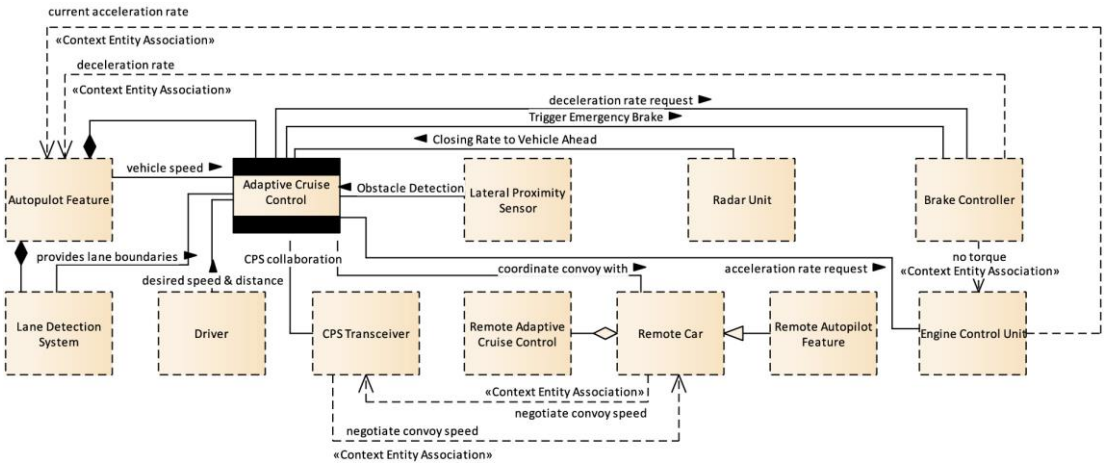
Operational Context

The operational context focusses on context objects that intetract with the system during runtime.

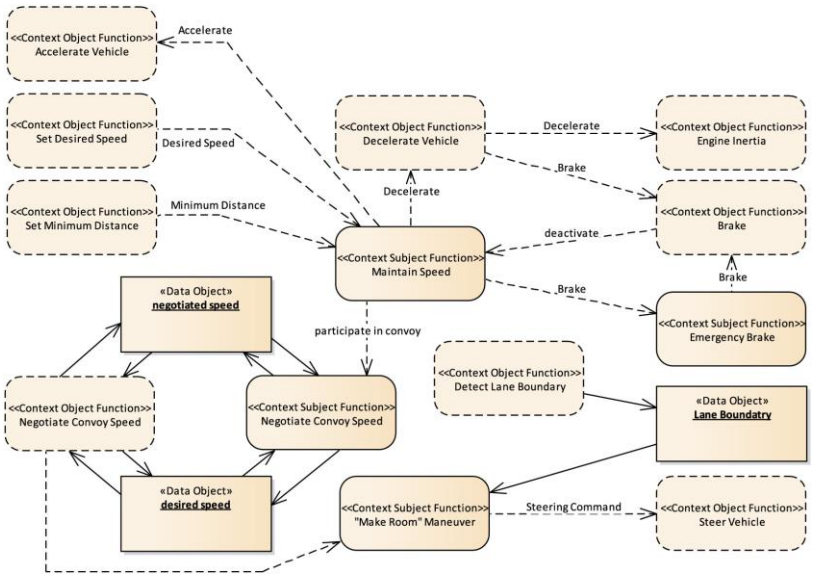
Thus, consideration of the operational context is important for defining interfaces and system behavior in accordance with its environment



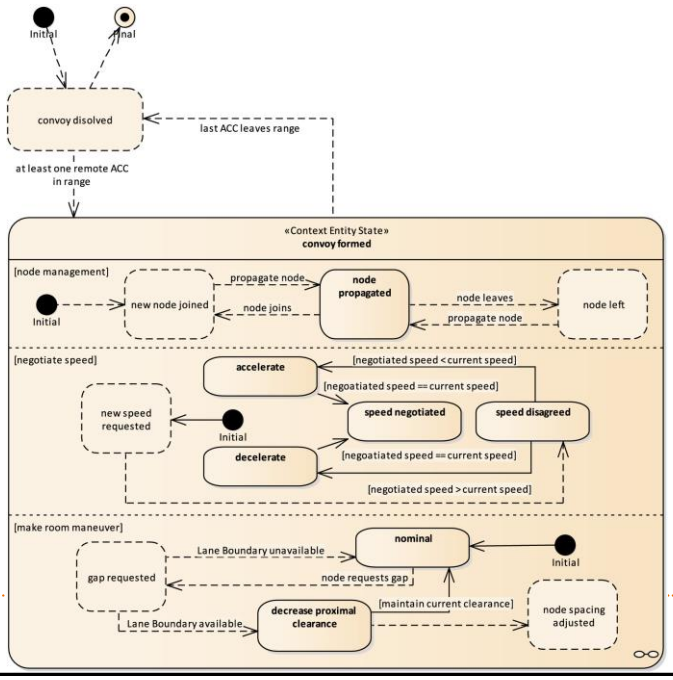
Structural Operational Context



Functional Operational Context



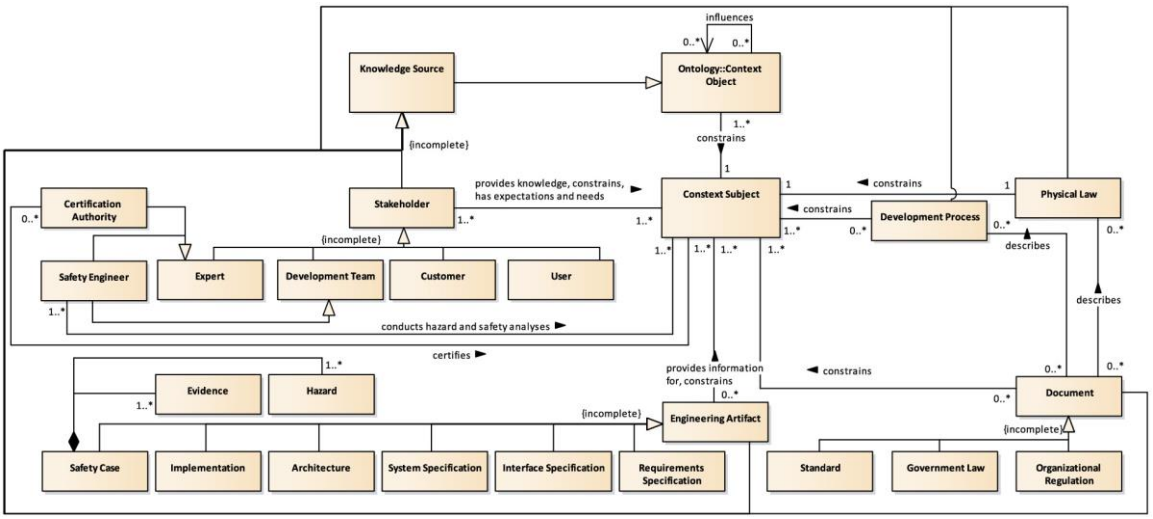
Behavioral Operational Context



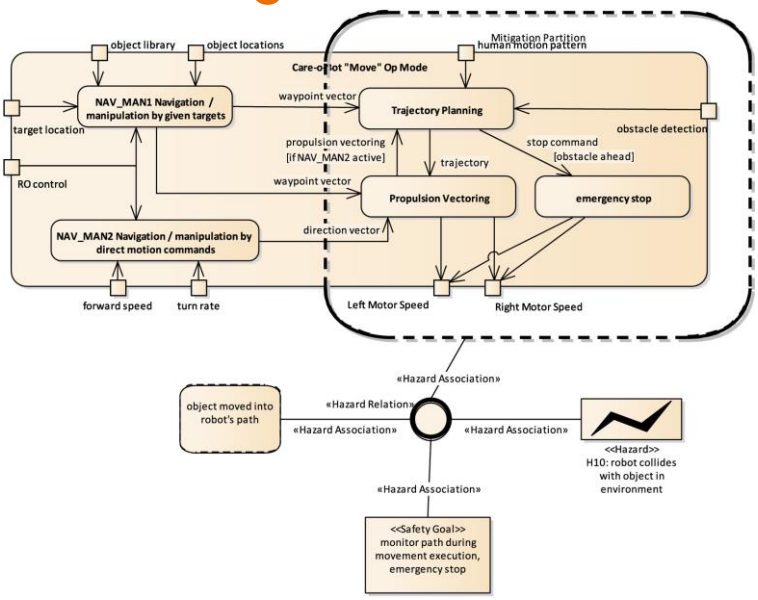
Exercise

Define the Context of
- A Cobot
- A Transport Robot

Safety Analyses



Hazard Modelling



Exercise

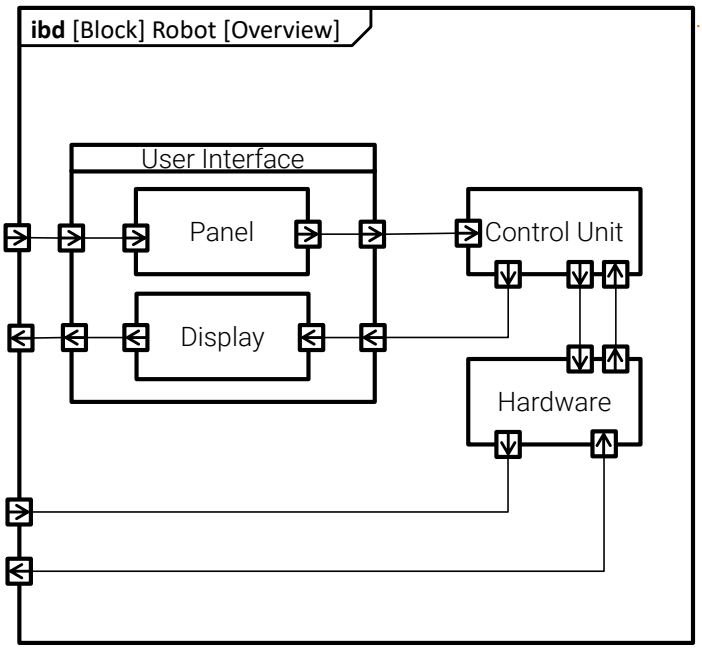
Identify Safety Hazards for
- A Cobot
- A Transport Robot

Modelling Structure with SysML Internal Block Diagrams

Internal Block Diagram

IBDs define the internal structure

IBDs define interfaces



What is an IBD?

Internal Block Diagram (ibd): An *Internal Block Diagram* is a static structural diagram owned by a particular Block that shows its encapsulated structural contents: Parts, Properties, Connectors, Ports, and Interfaces. Stated otherwise, an IBD is a "white-box" perspective of an encapsuated ("black-box") Block.

- Blocks can be recursively decomposed ("nested") into Parts by alternating between Block Definition Diagram (BDD) *definitions* and Internal Block Diagram (IBD) usages (See *Usage Notes* below.)
- Behaviors can either be encapsulated by Blocks (e.g., Operations, Signals, and State Machines) or Allocated (via «allocate» Dependency) to Blocks (e.g., Activities/Actions) directly or indirectly (via Interfaces).
- […]

From: https://sysml.org/sysml-faq/what-is-internal-block-diagram.html

Purpose of IBDs

The purpose of Internal Block Diagrams (IBDs) is to show the encapsulated structural contents (Parts, Properties, Connectors, Ports, Interfaces) of Blocks so that they can be recursively decomposed and "wired" using Interface Based Design techniques. […]

From: https://sysml.org/sysml-faq/what-is-internal-block-diagram.html

BDD vs IBD

BDD Block *Definition* vs. IBD Block *Usage* Dichotomy

BDDs and IBDs complement each other (cf. black-box vs. white-box) and support recursive structural decomposition techniques during System Analysis & Design.

- A BDD *defines* a Block's Properties, including its Part Properties (strongly owned Parts) and Reference Properties (shared Parts)
- IBD specifies Part Properties and Reference Properties *usages* or roles in the structural context of the Block that encapsulates them. Stated otherwise, Part Properties and Reference Properties in an IBD can have a different usages or roles depending upon how they are *realized* ("wired") in the IBD.
- […]

From: https://sysml.org/sysml-faq/what-is-internal-block-diagram.html

Exercise

Model BDDs and IBDs for describing
- A Cobot
- A Transport Robot

Questions for Self-Assessment

What is a Block Definition Diagram?
What is an Internal Block Diagram?
How do BDDs and IBDs complement each other?
What are important system layers in the development of embedded systems?
What influences the definition of components?
Why is context important?
How can context be used to identify safety hazards?

Literature

[Daun et al. 2016] M. Daun, B. Tenbergen, J. Brings, T. Weyer: SPES XT Context Modeling Framework. In: K. Pohl, M. Broy, H. Daembkes, H. Hönninger (eds.) Advanced Model-Based Engineering of Embedded Systems, Springer, 2016, pp. 43-57.

[Daun & Tenbergen 2022] M. Daun, B. Tenbergen: Context modelling for cyber-physical systems. In: J. Softw. Evol. Proc., Wiley, 2022.

[Friedenthal et al. 2014] Sanford Friedenthal, Alan Moore & Rick Steiner: A Practical Guide to SysML – The Systems Modeling Language. Morgan Kaufmann OMG Press, 2014.

[OMG SysML] OMG System Modeling Language. Version 1.6, Object Management Group, 2019.

[OMG UML] OMG Unified Modeling Language. Version 2.5.1, Object Management Group, 2017.