Imperial College London

Systems Engineering

Mars Rover 2022

Adam Bouchaala,
Teaching Fellow
Imperial College London
a.bouchaala@imperial.ac.uk
London, United Kingdom

Imperial College London

Ariane 5 Flight 501, 4 June 1996

(Link)

Imperial College London

1996: Ariane 501

" The failure of Ariane 501 was caused by the complete loss of guidance and attitude information 37 seconds after the start of the main engine ignition sequence (30 seconds after lift-off). This loss of information was due to **specification** and **design errors** in the software of the inertial reference system. The extensive reviews and tests carried out during the Ariane 5 development programme did not include adequate analysis and testing of the inertial reference system or of the complete flight control system, which could have detected the potential failure."

Mr Jean-Marie Luton, ESA Director General

Imperial College London

How to manage a complex system

In this presentation, we will discuss how to manage and design the Mars rover. A complex system is usually a system that has different engineering components. Mars rover is a complex system and adequate tools are needed to design it and define the interfaces between each subsystem! How to manage?

Systems Engineering!

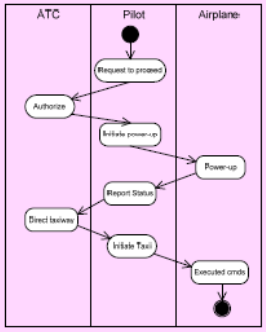
Imperial College London

Practices for describing systems

Past

- Specifications
- Interface requirements
- System design
- Analysis & Trade-off
- Test plans

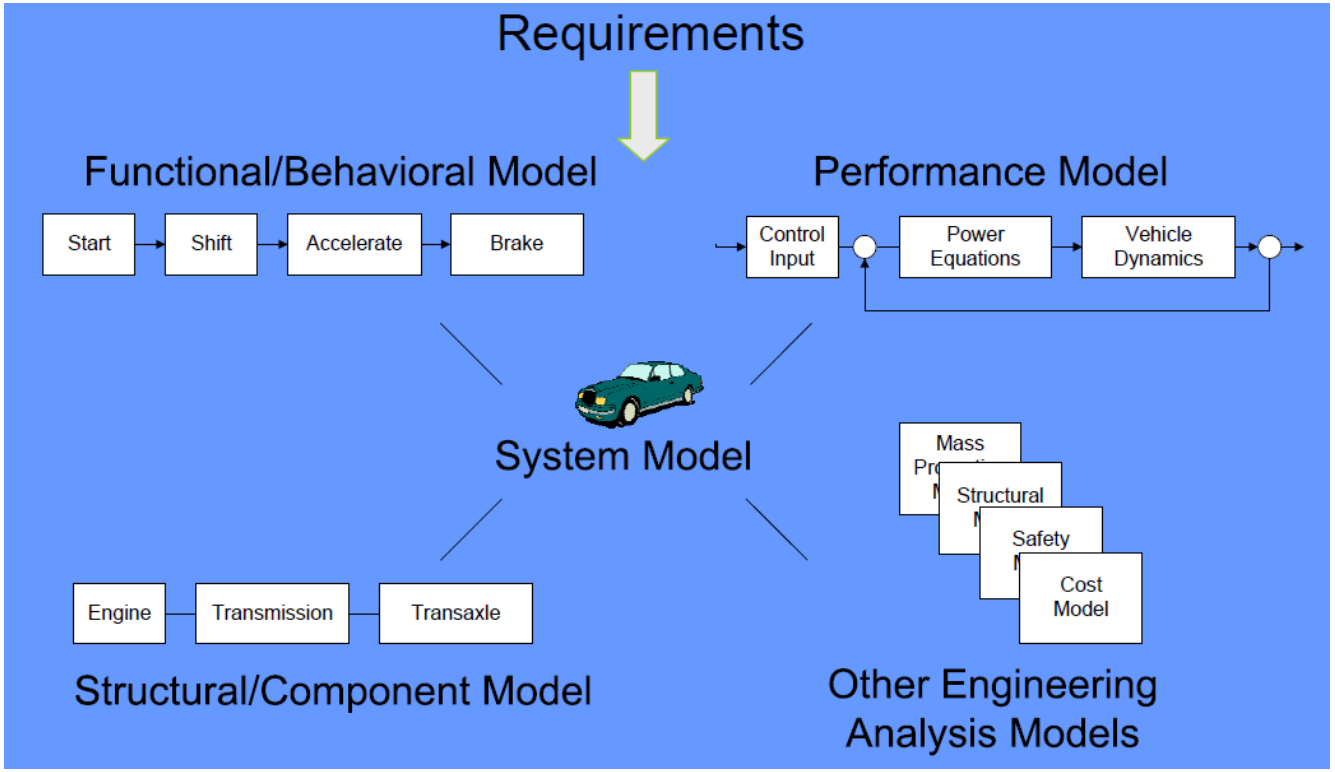
Future



Moving from Document centric to Model centric

Imperial College London

System Modelling



Integrated System Model Must Address Multiple Aspects of a System

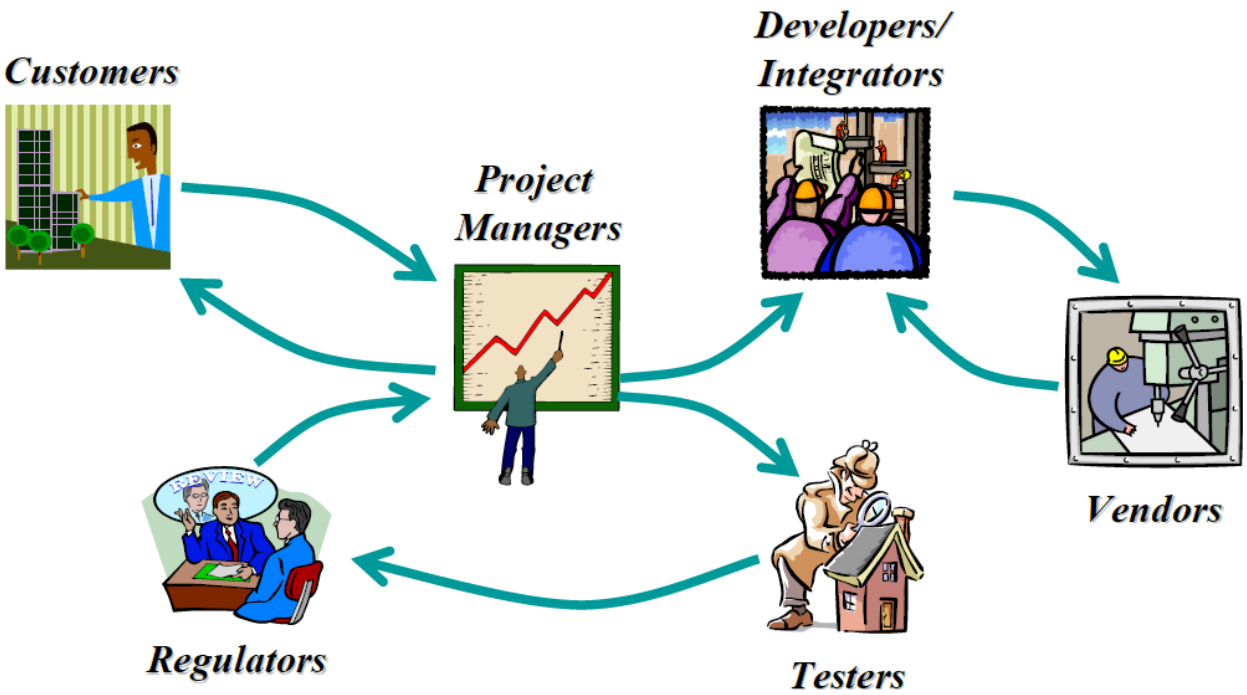
Imperial College London

MBSE benefits

- Shared understanding of system requirements and design
  - Validation of requirements
  - Common basis for analysis and design
  - Facilitates identification of risks
- Assists in managing complex system development
  - Separation of concerns via multiple views of integrated model
  - Supports traceability through hierarchical system models
  - Facilitates impact analysis of requirements and design changes
  - Supports incremental development & evolutionary acquisition
- Improved design quality
  - Reduced errors and ambiguity
  - More complete representation
- Supports early and ongoing verification & validation to reduce risk
- Provides value through life cycle (e.g., training)
- Enhances knowledge capture

Imperial College London

Stakeholders



Modelling needed to improve communications

Imperial College London

Model Based Systems Engineering (MBSE)

Systems engineering is a process that we can use to develop something that is too complex to design and build as a single monolithic entity.

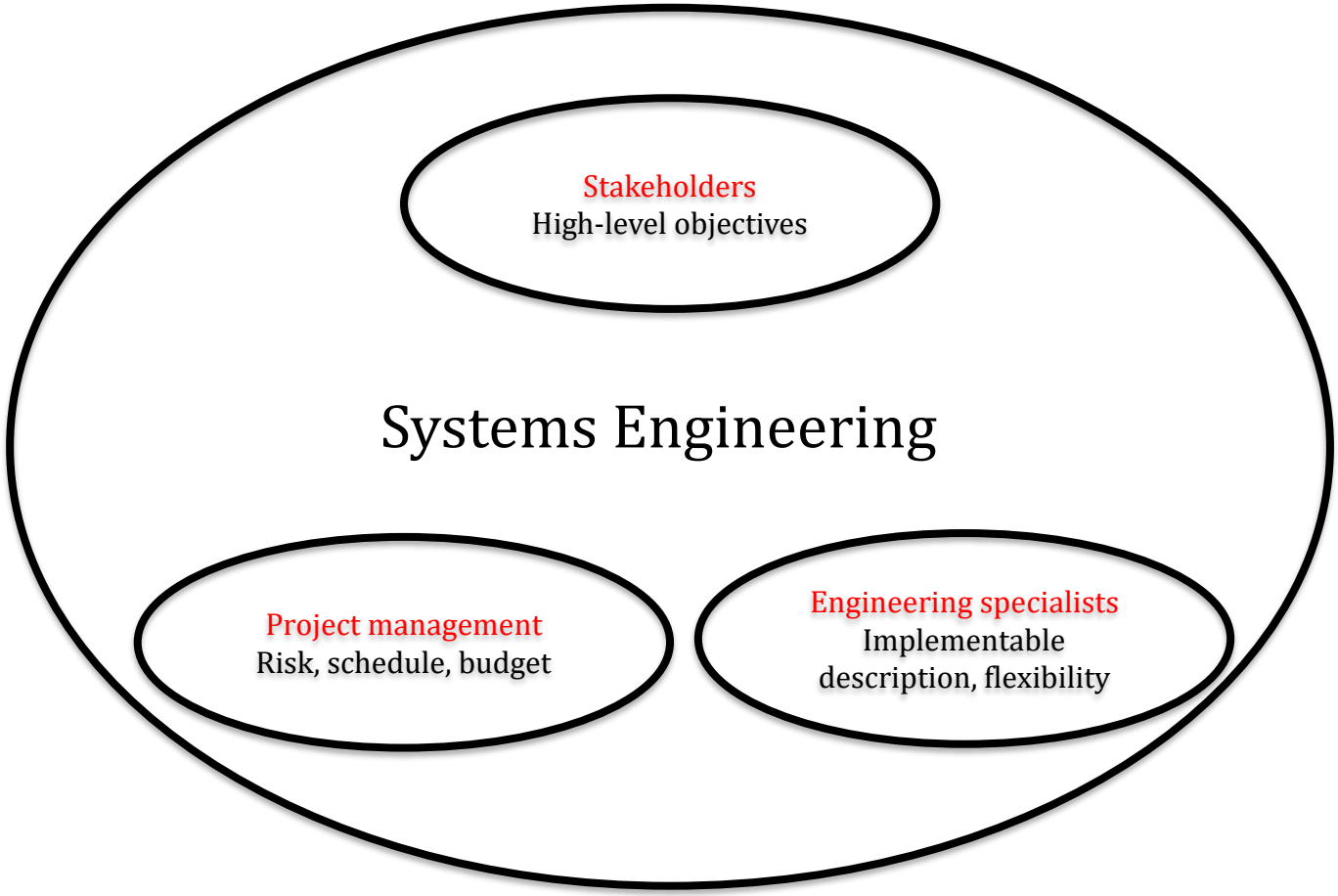
Idea

drone

It is guiding the engineering process so that the system can be implemented and meets the needs of the project.

Imperial College London

Project needs



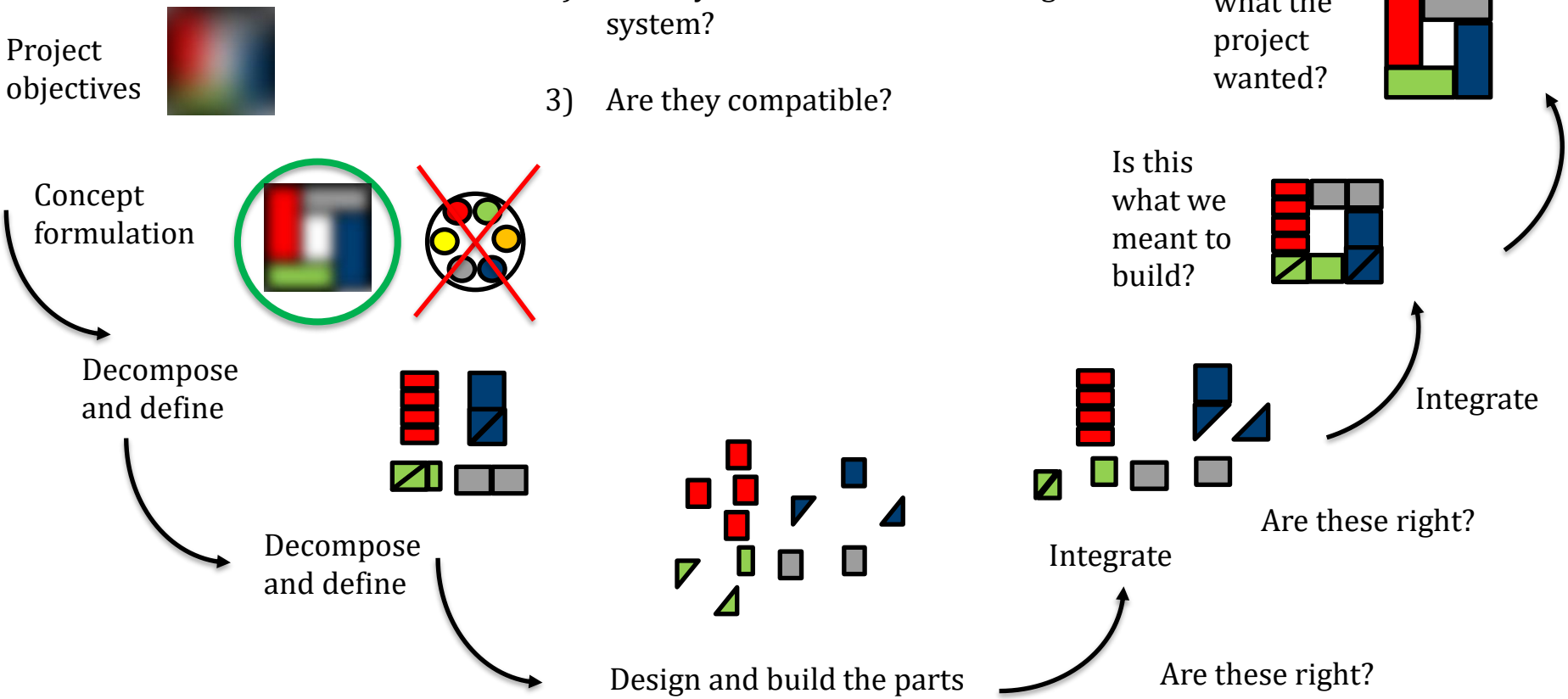
Imperial College London

Challenges:

1) Are these the right components? Functionally, logically, physically

2) Do they meet the needs of the higher system?

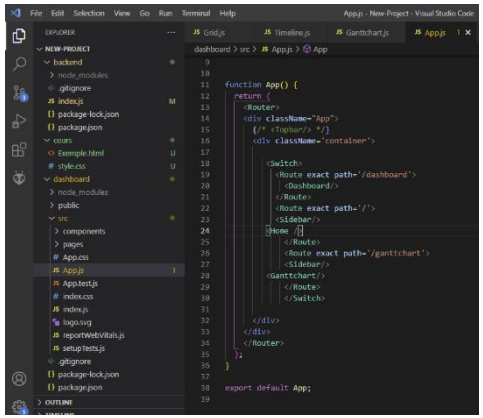
3) Are they compatible?



Imperial College London

Systems engineering is additional work

Valuable!



Required

Not required

Valuable?

Talking about writing code:

1) Requirements review
2) Architecture diagrams

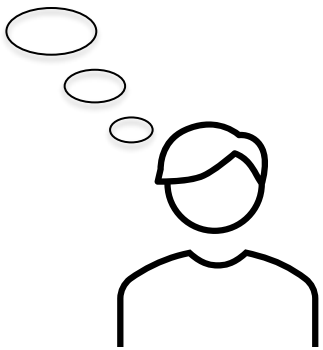
Necessary for complex systems

Why take a systems engineering approach?

Imperial College London

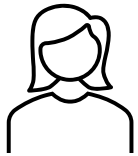
I want a car door!

- Strong structure
- Hinged
- Latch
- Can be opened from both side
- I want to see out!
- And open the window!
- Also electric!
- And safe!



Stakeholder

Mechanical engineer

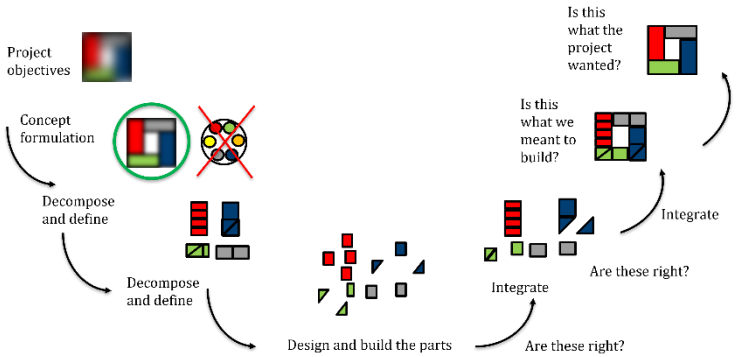


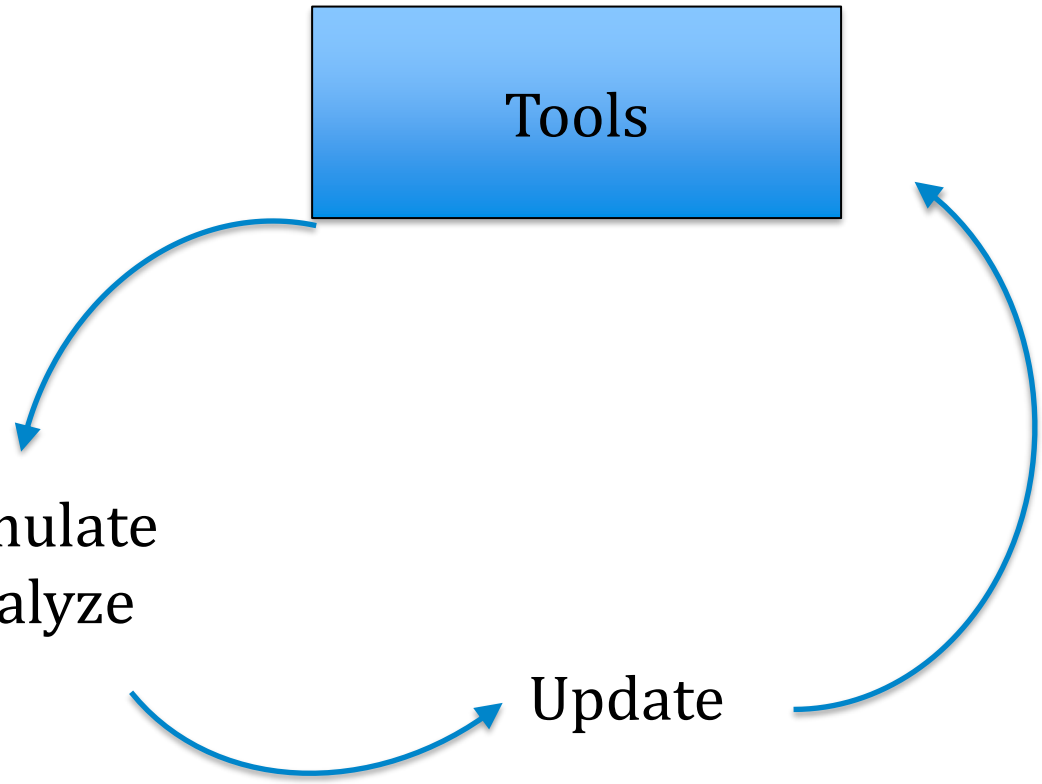
Electrical engineer

Software engineer

Imperial College London

What do we need?





Imperial College London

Systems engineering steps!

Stakeholder needs

What does the system need to do?

Concept exploration

Examine potential system concepts

Performance requirements

What are the key characteristics

Functional analysis and formulations

Functional, physical, logical, architectures.

Detailed development

Low-level requirements
Define subsystems configuration

Engineering design

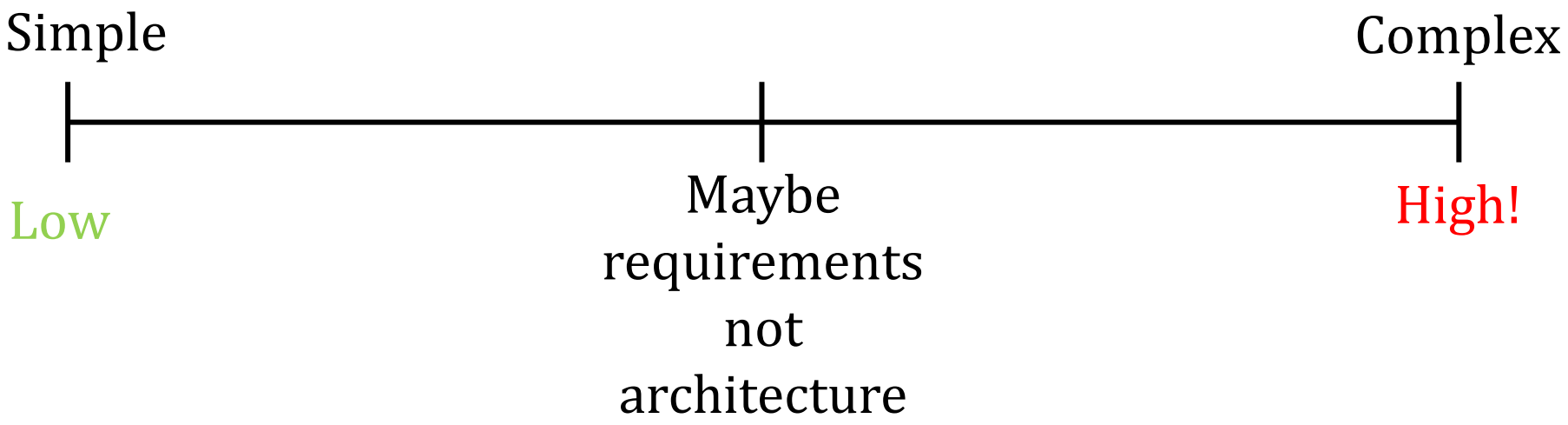
Detail design and manufacturing

Integration and test

Build and verify
Validate system

Imperial College London

Formal systems engineering need!



Imperial College London

Systems Modelling Language (SysML)

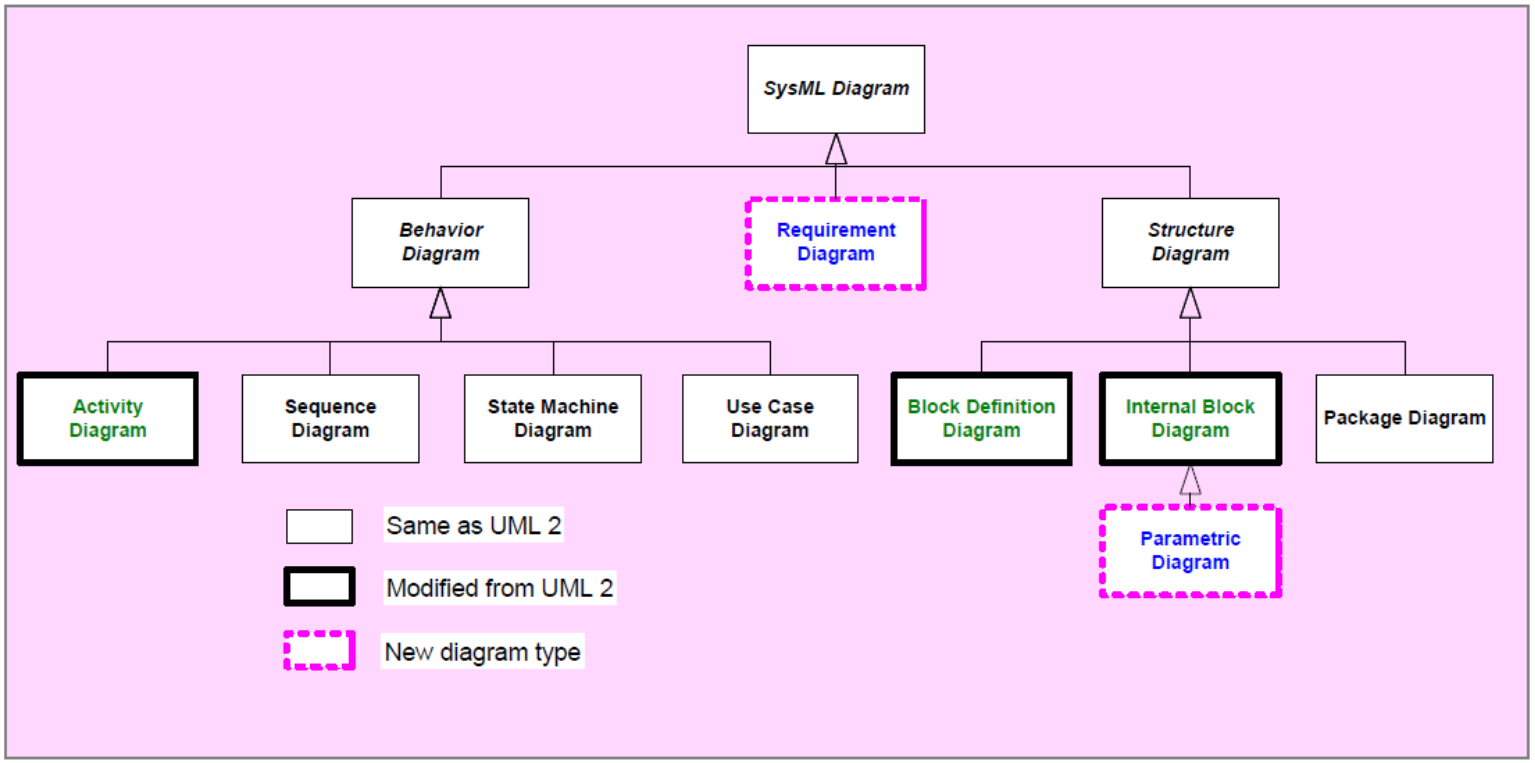
- A graphical modelling language in response to the UML for Systems Engineering RFP developed by the OMG, INCOSE, and AP233.

- Supports the specification, analysis, design, verification, and validation of systems that include hardware, software, data, personnel, procedures, and facilities.

- ***Is*** a visual modelling language that provides
  Semantics = meaning
  Notation = representation of meaning
- ***Is not*** a methodology or a tool
  SysML is methodology and tool independent

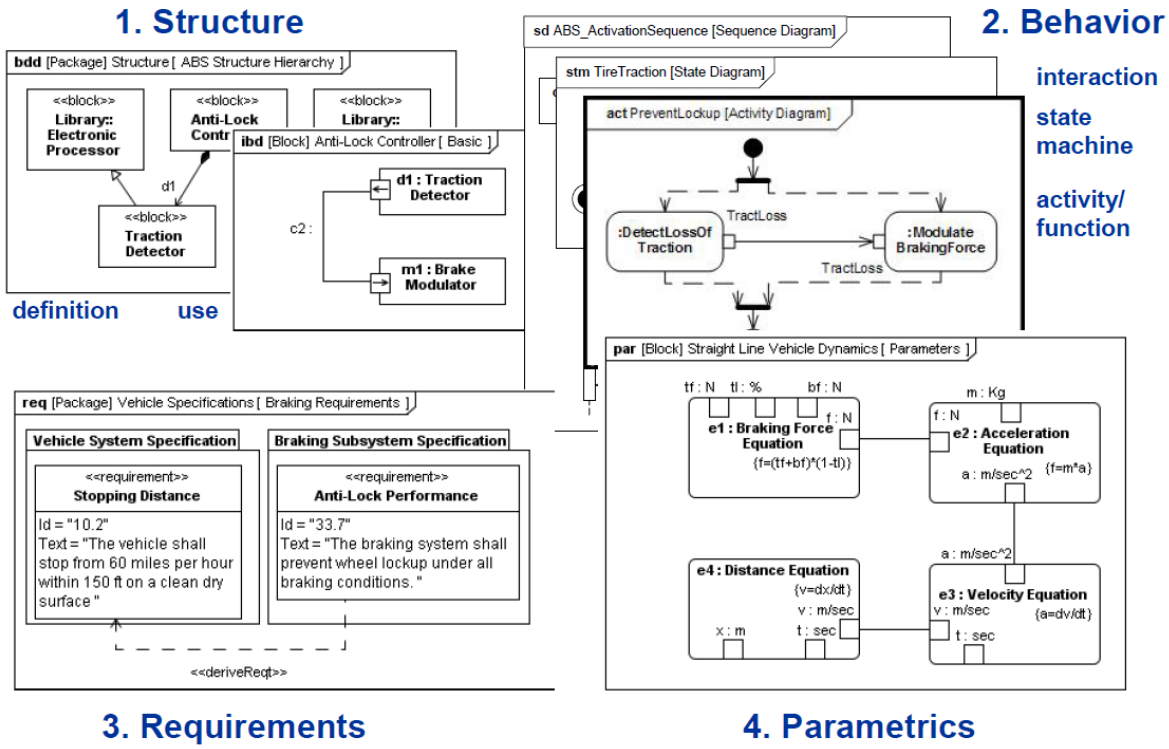
Imperial College London

SysML Diagram Taxonomy



Imperial College London

SysML pillars



Imperial College London

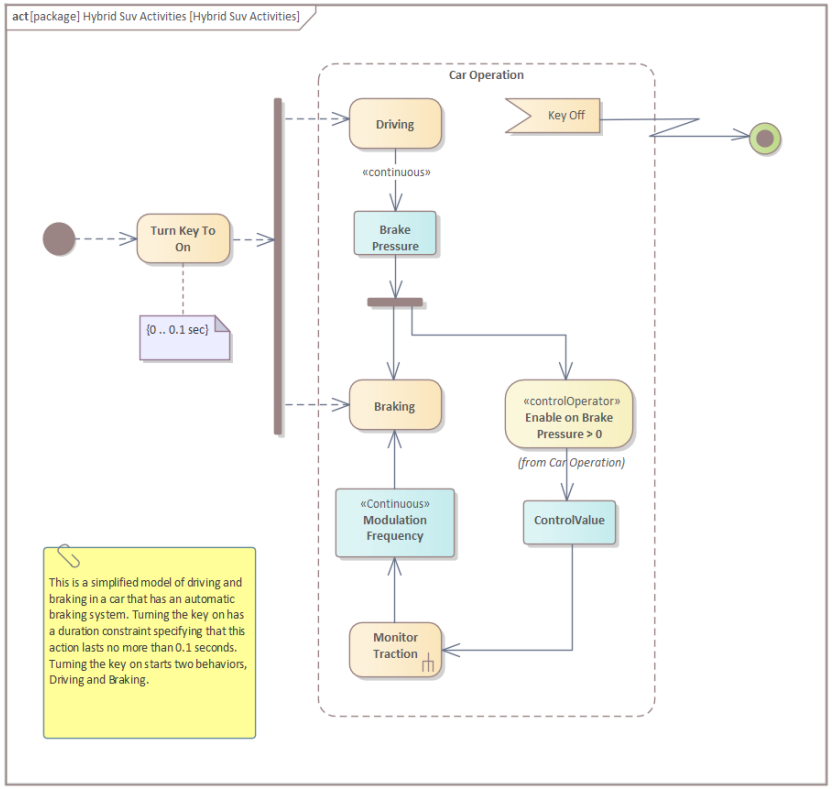
Behaviour diagrams

Imperial College London

Activity diagram

- The Activity diagram is a powerful tool for representing the sequence of Actions that describe the behaviour of a Block or other structural element; the sequence is defined using Control Flows. Actions can contain Input and Output Pins that act as buffers for items that flow from one Action to another as the task carried out by the Action either consumes or produces them. The items can be physical materials, energy, power, data, information, or anything else that can be produced, conveyed or consumed, depending on the system and the activity being described.

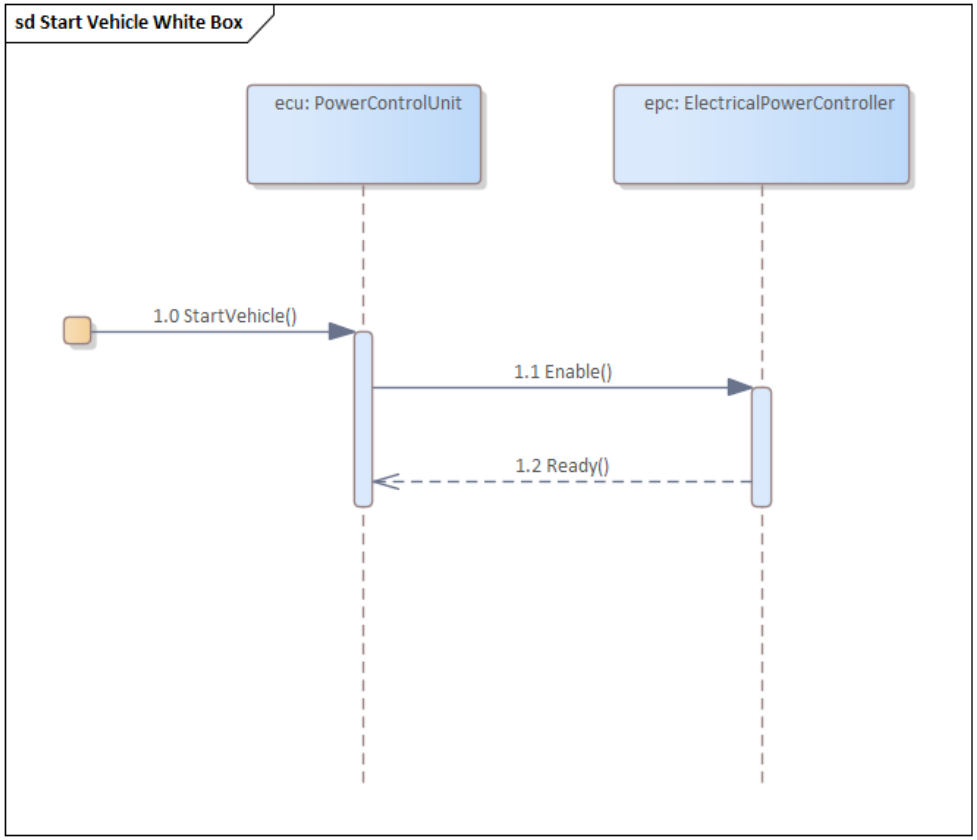
- Activity Diagrams can be used to define situations where parallel processing occurs in the execution of some activities. Activity diagrams are useful for engineering modeling, where they detail the processes involved in system activities.



Imperial College London

Sequence diagram

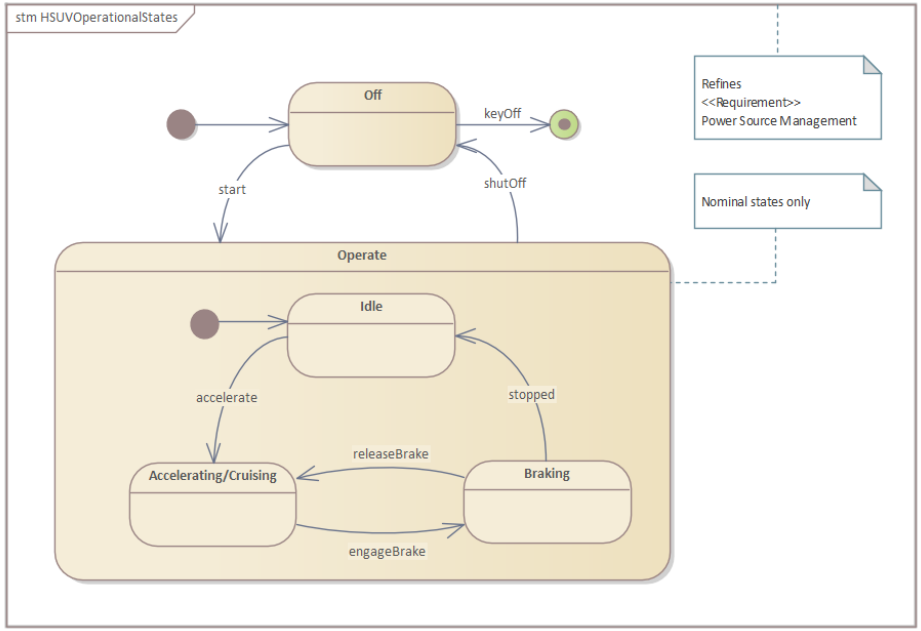
- A SysML Sequence diagram, as for a UML Sequence diagram, is used to display the interaction between users, screens, objects and entities within the system. It provides a sequential map of messages passing between objects over time. Frequently, these diagrams are placed under Use Cases in the model to illustrate the Use Case scenario - how a user will interact with the system and what happens internally to get the work done.
- A Sequence diagram is a form of Interaction diagram, which shows objects as Lifelines running down the page, with their interactions over time represented as Messages drawn as arrows from the source Lifeline to the target Lifeline. Sequence diagrams are good at showing which objects communicate with which other objects, and what messages trigger those communications.



Imperial College London

State Machine diagram

- A **StateMachine diagram** is a powerful vehicle for presenting information about the lifetime of a system element such as a Block, which might have complex behavior and might have life cycles that are difficult to understand. The diagram can be used to describe the important conditions (States) that an entity might pass through during its lifetime or life cycles. Typically, only entities that have important stages in their lifetime are modeled with StateMachine diagrams. The entity is said to transition from one State to another as defined by the StateMachine. Triggers and Events can be described that allow the State transition to occur, and Guards can be defined that restrict the change of State. Each State can define the behaviors that occur on entry to the State, while existing in the State, and on exit from the State

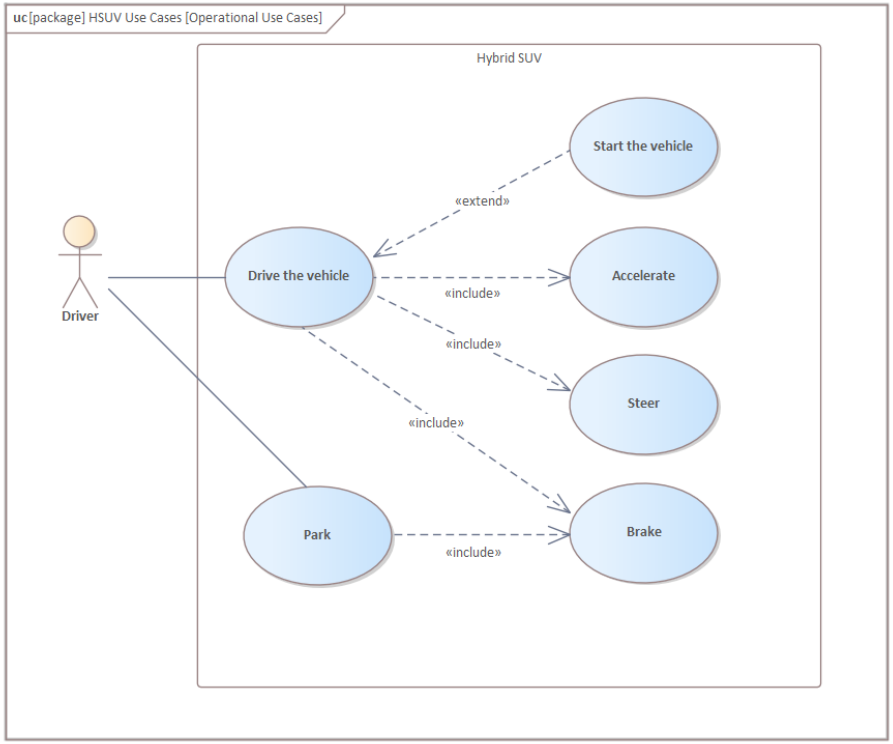


Imperial College London

Use Case diagram

- A SysML Use Case diagram is used to define and view Use Cases and the Actors that derive value from the system. The Use Case diagram describes the relationship between the Actors and the Use Cases. Enclosing the Use Case within a Boundary defines the border of the system; the Actors by definition lie outside the boundary. (All elements are internal to the SysML diagram frame.)

- While the Use Case diagram can appear simplistic, it is a powerful communication device that describes the value or goals that external roles achieve from interacting with the system. Each Use Case can be detailed, with descriptions, constraints and any number of scenarios that contain sets of steps performed alternately by Actor and system to achieve the desired goal.



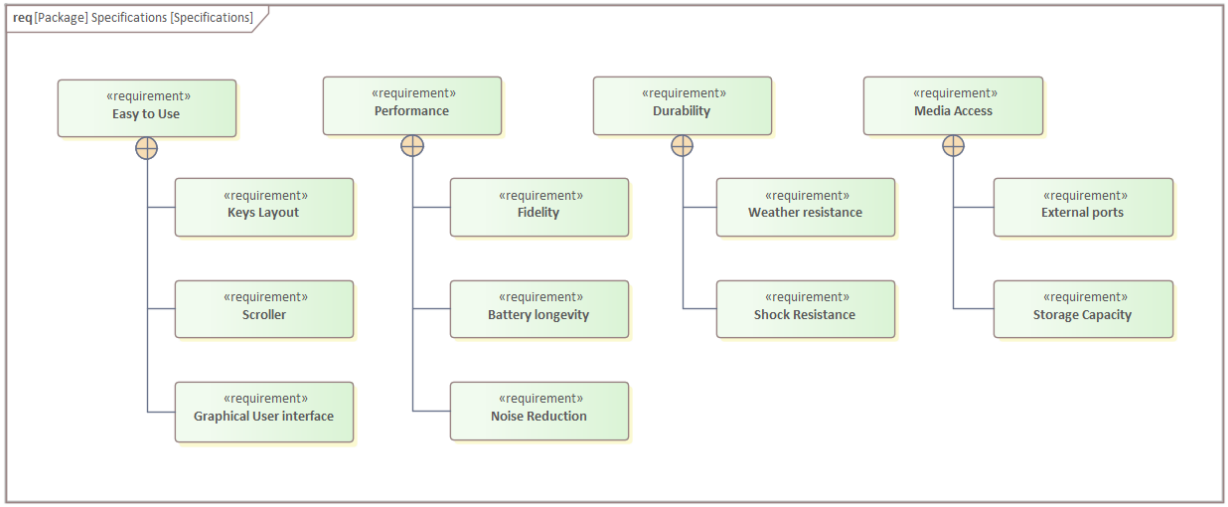
Imperial College London

Requirements diagrams

Imperial College London

Requirement diagram

- The SysML Requirements Model provides the system requirements, the expected abstract behaviour and the operating constraints that the designed system must conform to. This diagram shows an example Requirements model for a Portable Audio Player.



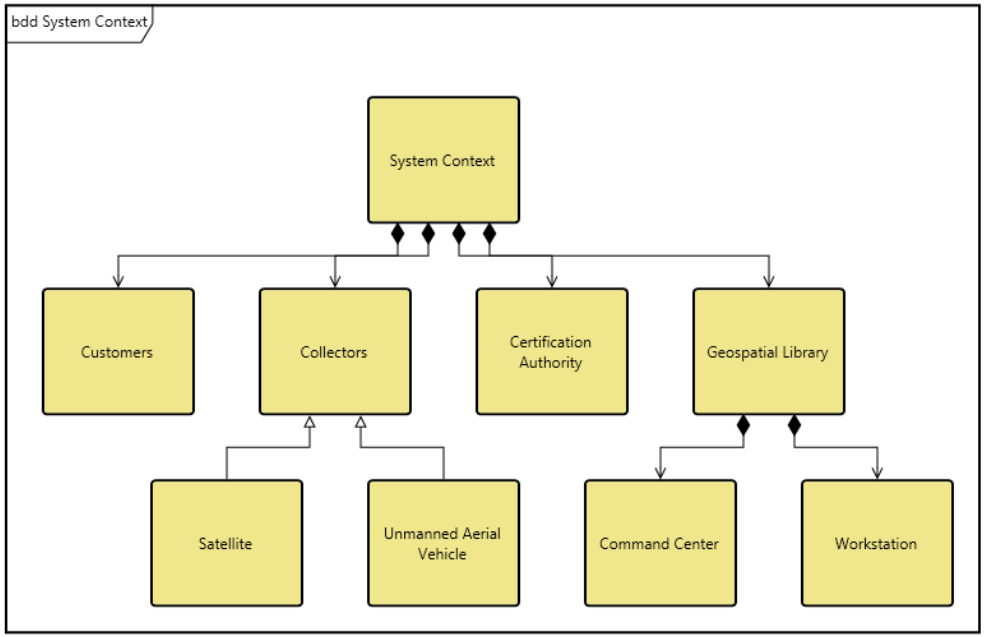
Imperial College London

Structural diagrams

Imperial College London

Block Definition Diagram

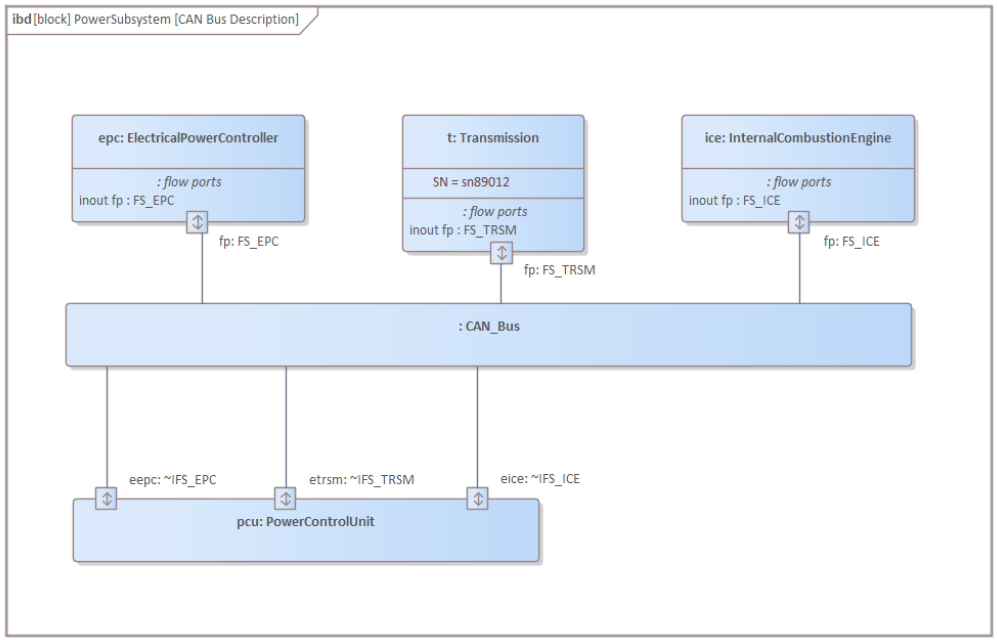
- A Block defines a collection of features used to describe a system, subsystem, component or other engineering object of interest. These features can include both structural and behavioral features, such as properties, operations and receptions, that represent the state of the system and the behavior that the system might exhibit.



Imperial College London

Internal Block Diagrams

- An Internal Block diagram (IBD) captures the internal structure of a Block element, in terms of its properties (Ports and Parts) and the connections between those properties. The IBD is an instance of the Block element, and the Block is the classifier for the IBD.



Imperial College London

Parametric Diagrams

- SysML Parametric models support the engineering analysis of critical system parameters, including the evaluation of key metrics such as performance, reliability and other physical characteristics. These models combine Requirement models with system design models, by capturing executable constraints based on complex mathematical relationships, which can be defined and calculated using the integrated facilities of Math Simulation tools such as MATLAB Simulink, OpenModelica and Octave, under the SysML Extension for Physical Interaction and Signal Flow Simulation (SysPhS) standard. The Mathematical Simulation tools are discussed in the *Mathematical Simulations* chapter.
- Parametric diagrams are specialized Internal Block diagrams that help you, the modeler, to combine behavior and structure models with engineering analysis models such as performance, reliability, and mass property models.
- SysML Parametric diagrams are dependant on Block definitions being created in the model. The parametric definitions apply equations as constraints on the properties of these Blocks. The equations have parameters that are bound to the properties of the system. Parametric diagrams use ConstraintBlocks to define these constraints. These can be derived from the Block Definition or Internal Block model.