SysML

Historical Perspective
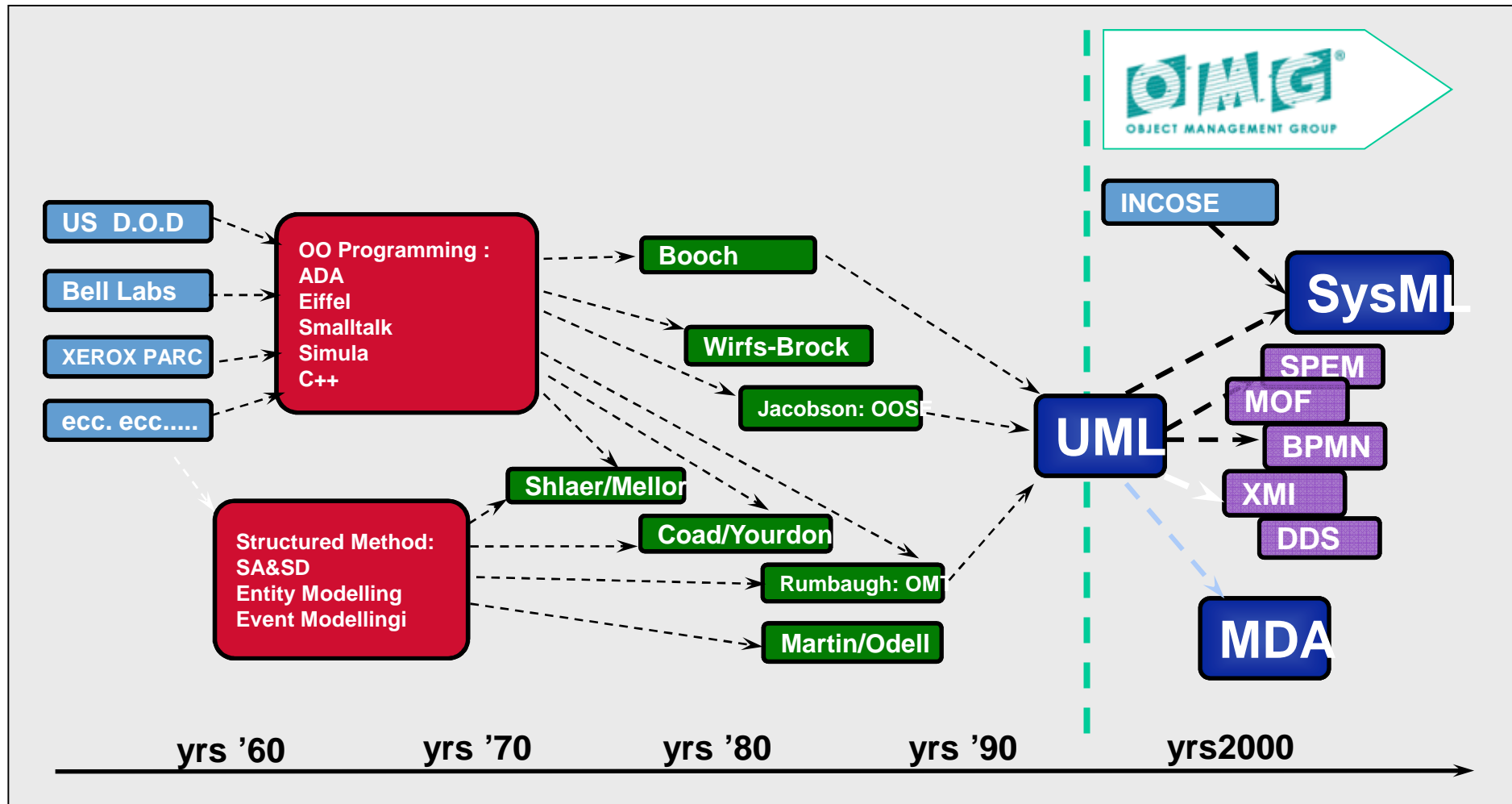
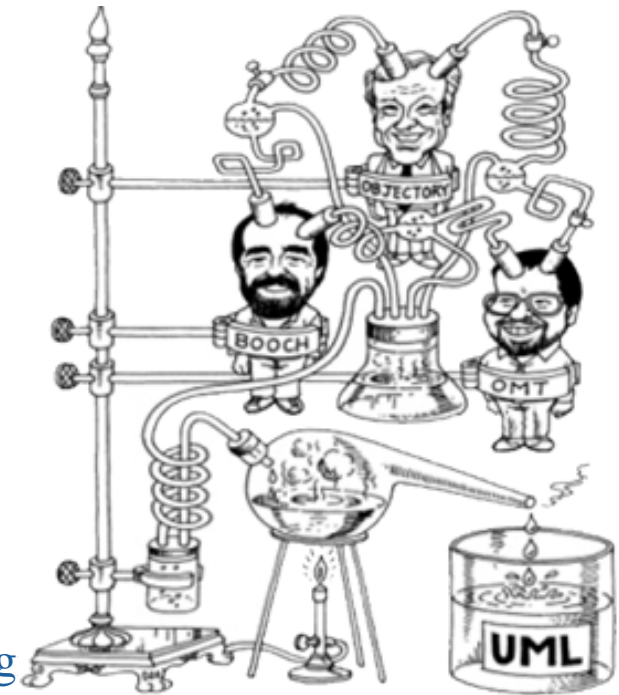# From document centric to Model centric
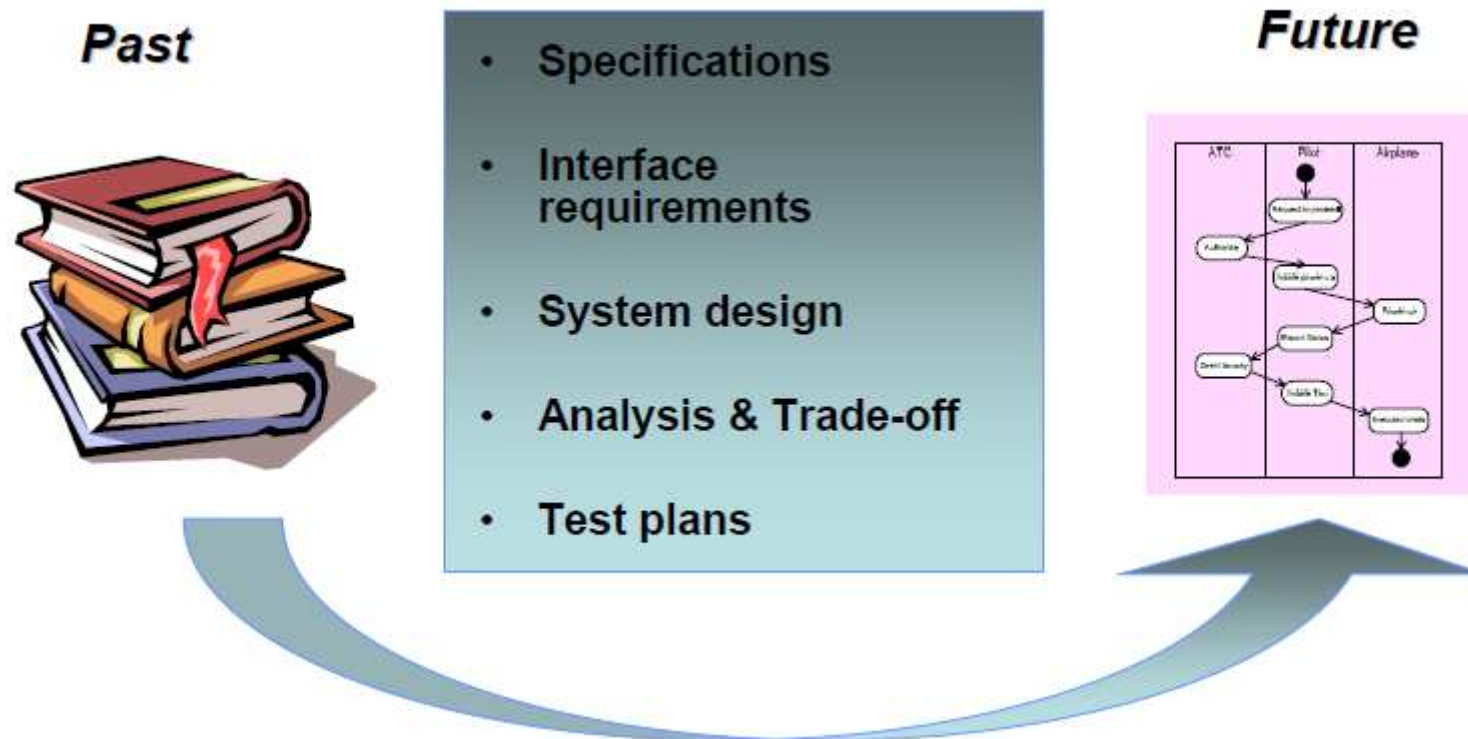
What is UML

➢ UML (Unified Modeling Language)
  ❑ A consensus on best practices in software eng
    • Now a de facto standard
  ❑ A graphical modeling language supported by tools

From UML to SysML (3/3)

➢ Why not UML for systems ?

  ❑ Objects are for computer-literates, not systems engineers

  ❑ Requirements are described outside the model using, e.g., DOORS

  ❑ Allocation relations are not explicitly supported

➢ Nevertheless SysML is a UML 2 profile

  ❑ In response to the UML for Systems Engineering RFP developed by the OMG and INCOSE

➢ SysML standard

  ❑ http://www.omgsysml.org/

Who are the main contributors

➢ Industry & Government

❑ American Systems, BAE SYSTEMS, Boeing, Deere & Company, EADS-Astrium, Eurostep, Lockheed Martin, Motorola, NIST, Northrop Grumman, oose.de, Raytheon, THALES

➢ Vendors

❑ Artisan, EmbeddedPlus, Gentleware, IBM, I-Logix, Mentor Graphics, PivotPoint Technology, Sparx Systems, Telelogic, Vitech Corp

➢ Academia

❑ Georgia Institute of Technology

➢ Liaison Organizations

❑ INCOSE, ISO AP233 Working Group

What is SysML?

➢ **SysML has been designed with systems in mind**
  ❑ A graphical modelling language that supports the specification, analysis, design, verification, and validation of systems

➢ SysML is a notation, not a method

• **The SysML standard is a reference language for system engineering**

➢ SysML is supported by tools
  ❑ SysML plugins for UML tools
  ❑ Proprietary tools: Entreprise Architect, Rhapsody, Modelio, …
  ❑ Free software tools: TOPCASED, TTool, …

Relationship between UML and SysML
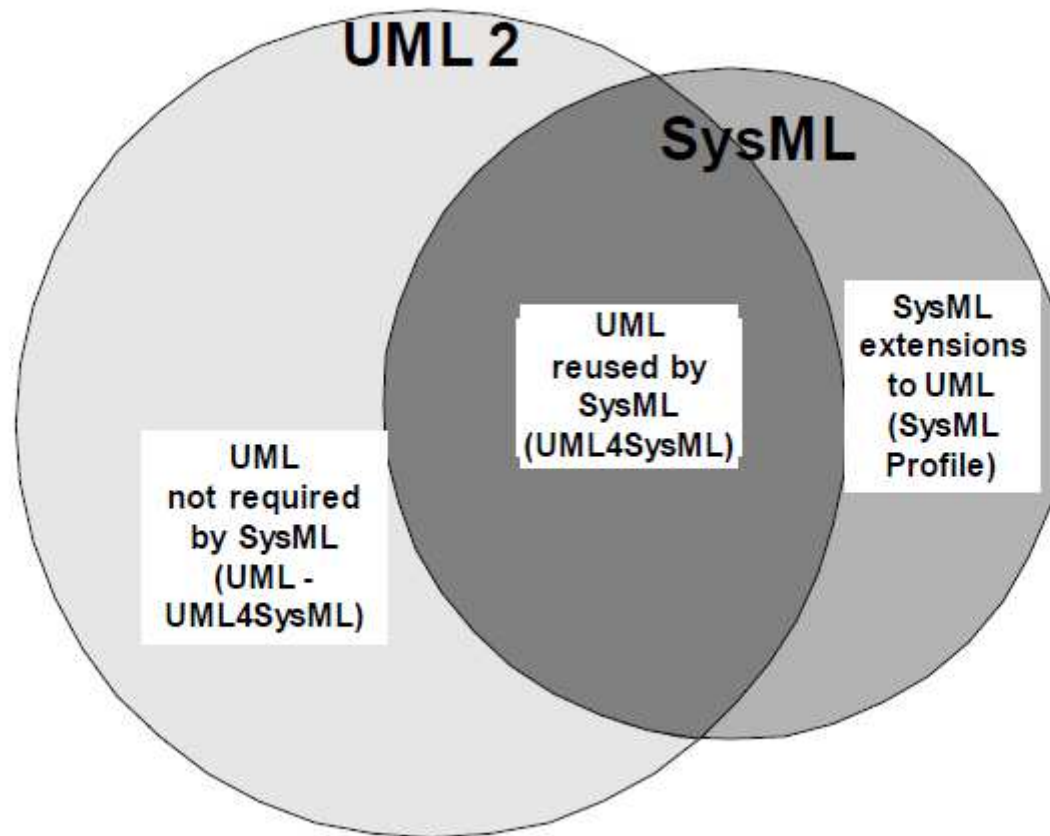
SysML Diagrams Taxonomy

```
                          ┌──────────────────┐
                          │  SysML Diagram   │
                          └──────────────────┘
                                    △
        ┌───────────────────────────┼───────────────────────────┐
┌──────────────┐          ┌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌┐          ┌──────────────┐
│  Behavior    │          ┊  Requirement     ┊          │  Structure   │
│  Diagram     │          ┊  Diagram         ┊          │  Diagram     │
└──────────────┘          └╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌╌┘          └──────────────┘
        △                                                      △
```
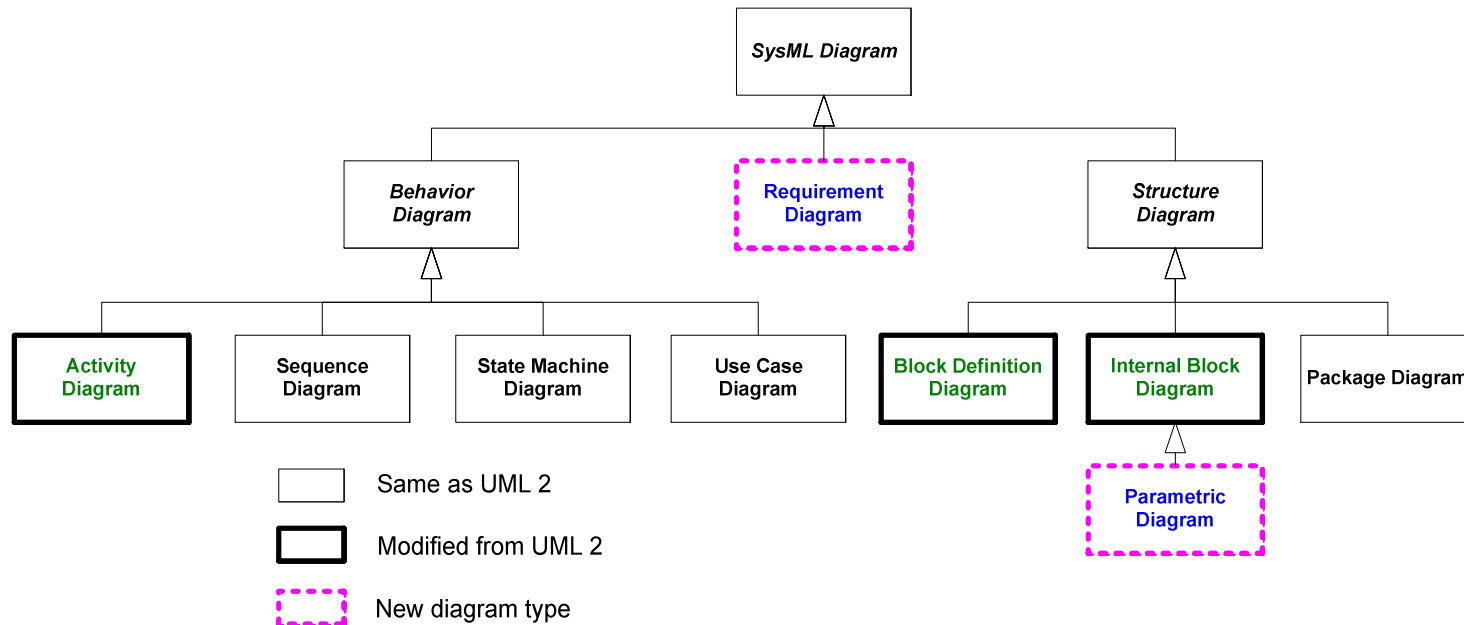
| Activity Diagram | Sequence Diagram | State Machine Diagram | Use Case Diagram | | Block Definition Diagram | Internal Block Diagram | Package Diagram |

```
                                                                      △
                                                           ┌╌╌╌╌╌╌╌╌╌╌╌╌╌┐
                                                           ┊ Parametric  ┊
                                                           ┊ Diagram     ┊
                                                           └╌╌╌╌╌╌╌╌╌╌╌╌╌┘
```

☐	Same as UML 2
☐ (bold)	Modified from UML 2
☐ (magenta dashed)	New diagram type

SysML diagrams (1/3)

Diagram name	Description
Requirement Diagram	Shows system requirements and their relationships with other elements.
Use-case Diagram	Shows system services and functions with respect to its external environment.
Sequence Diagram	Show interactions betweens blocks inside the system, as well as interactions between the system and actors.
Activity Diagram	Shows system behavior as function flows and data flows

SysML diagrams (2/3)

Diagram name	Description
Package Diagram	Show how a model is organized into packages, views and viewpoints.
Block Diagram	Show system structure as components along with their properties, operations and relationships.
Internal Block Diagram	Show the internal structures of components, including their parts and connectors.

SysML diagrams (3/3)

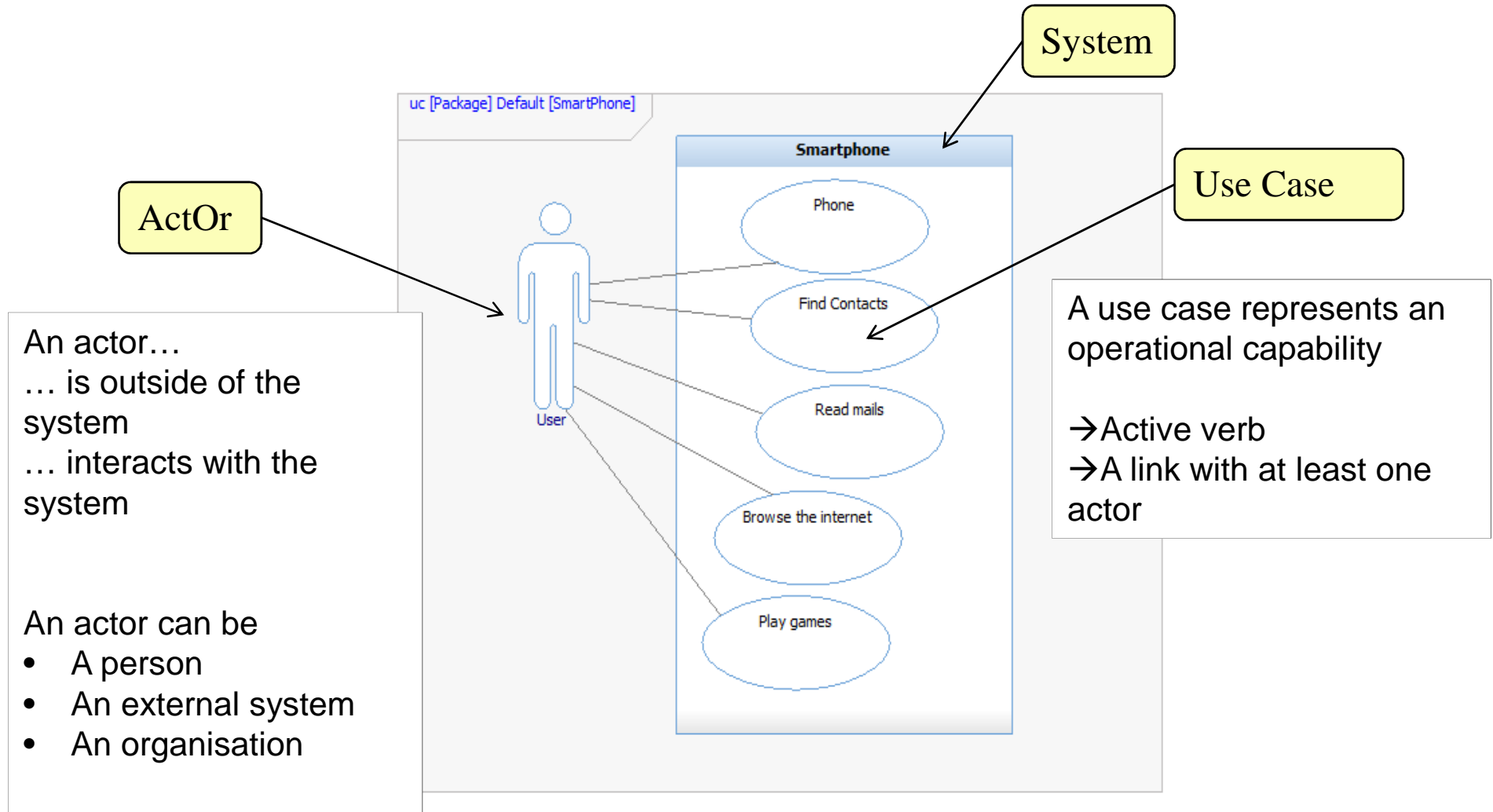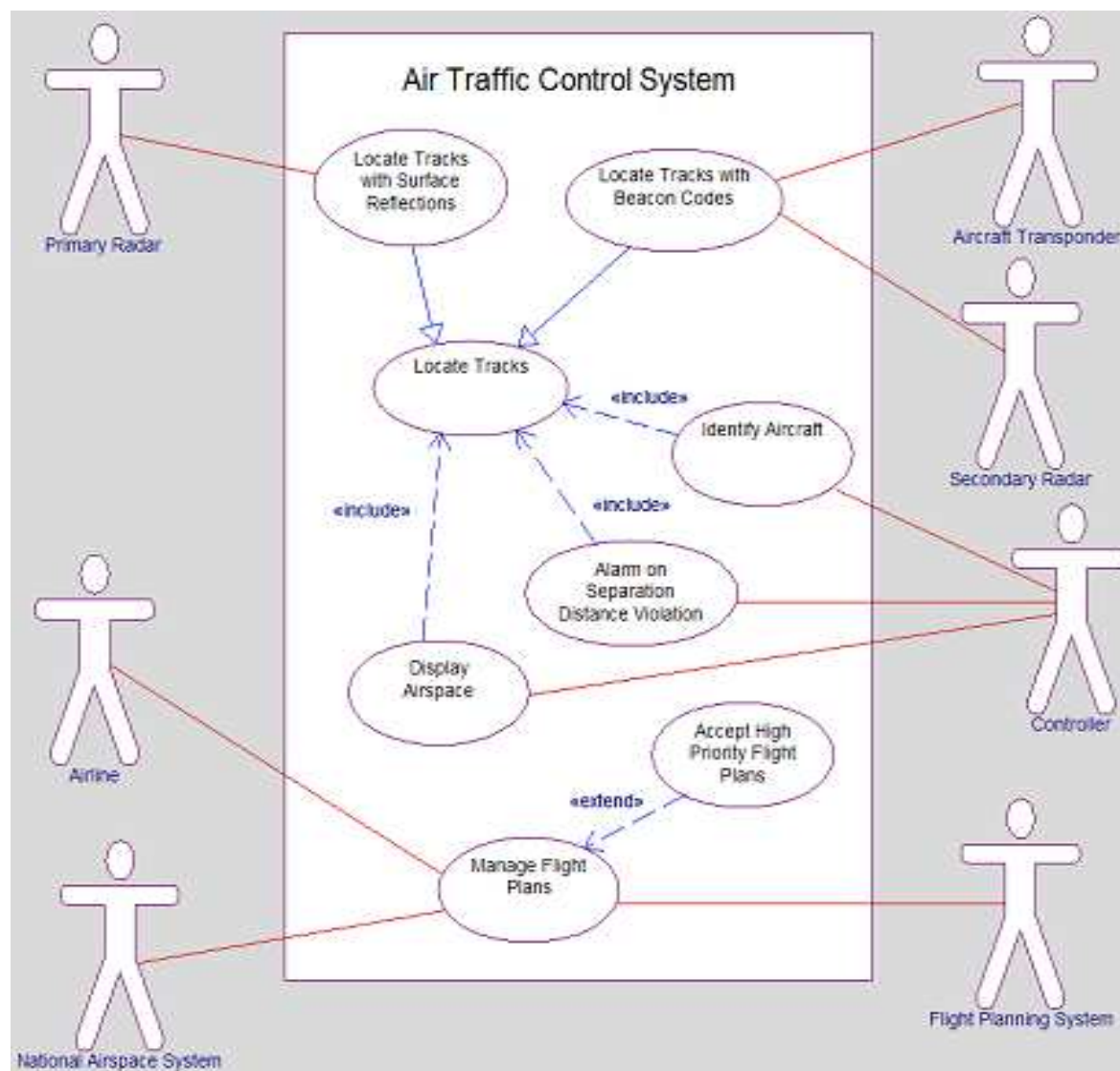| Diagram name	Description
State Machine Diagram	Shows system internal behavior as states and transitions between states.
Parametric Diagram	Shows parametric constraints between structural elements.

Use Case diagram

Why use case modeling?

➢ Use case modeling is all about describing WHAT the system will do:
- ❑ At a high level – Analysis NOT Design
- ❑ From the perspective of the User

➢ Use cases scope the system:
- ❑ The suppliers have to develop the functionality described by the use cases.
- ❑ Users should not expect more/less/different functionality than is described by the use cases.

➢ Use cases organize requirements:
- ❑ Typically 8-20 pages of requirements per use case.
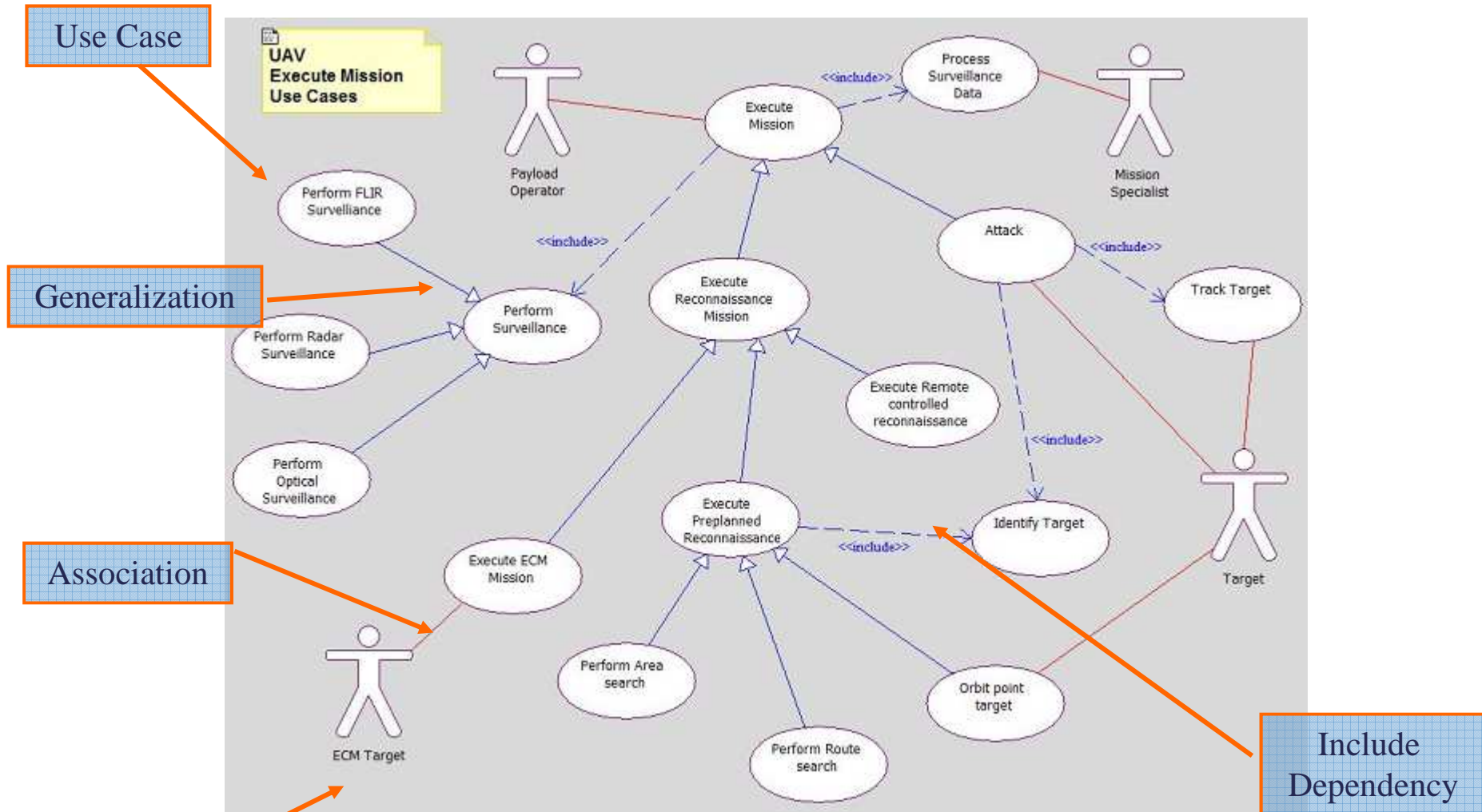
Example of Use Case Diagram

System

uc [Package] Default [SmartPhone]

Smartphone

ActOr

Use Case

Phone

Find Contacts

Read mails

Browse the internet

Play games

User

An actor…
… is outside of the system
… interacts with the system

An actor can be
- A person
- An external system
- An organisation

A use case represents an operational capability

→Active verb
→A link with at least one actor

Elaborated Use case

Air Traffic Control System

- Primary Radar
- Aircraft Transponder
- Secondary Radar
- Controller
- Airline
- National Airspace System
- Flight Planning System

Use cases:
- Locate Tracks with Surface Reflections
- Locate Tracks with Beacon Codes
- Locate Tracks
- Identify Aircraft «include»
- Alarm on Separation Distance Violation «include»
- Display Airspace «include»
- Accept High Priority Flight Plans
- Manage Flight Plans «extend»

Complete use case syntax

Use Case

Generalization

Association

Actor

Include Dependency

UAV Execute Mission Use Cases

Payload Operator

Execute Mission

Process Surveillance Data

<<include>>

Mission Specialist

Perform FLIR Survelliance

<<include>>

Attack

<<include>>

Track Target

Perform Surveillance

Execute Reconnaissance Mission

Perform Radar Surveillance

Execute Remote controlled reconnaissance

Perform Optical Surveillance

<<include>>

Identify Target

Execute Preplanned Reconnaissance

<<include>>

Target

Execute ECM Mission

Perform Area search

Orbit point target

ECM Target

Perform Route search

Sequence Diagram

SysML tool: sequence diagram

- Objective

    - Representation of a sequence of operations, progressing in time

    - Description of the interactions between the system of interest and external actors

Sequence Diagram: example

System Lifeline

:User

SmartPhone

Ref

Start_phone

Ask_for_contacts_list()

Display_Contact_List()

Select_contact()

Start_call()

Time

Interaction

:User

SmartPhone

Start_Phone()

Welcome()
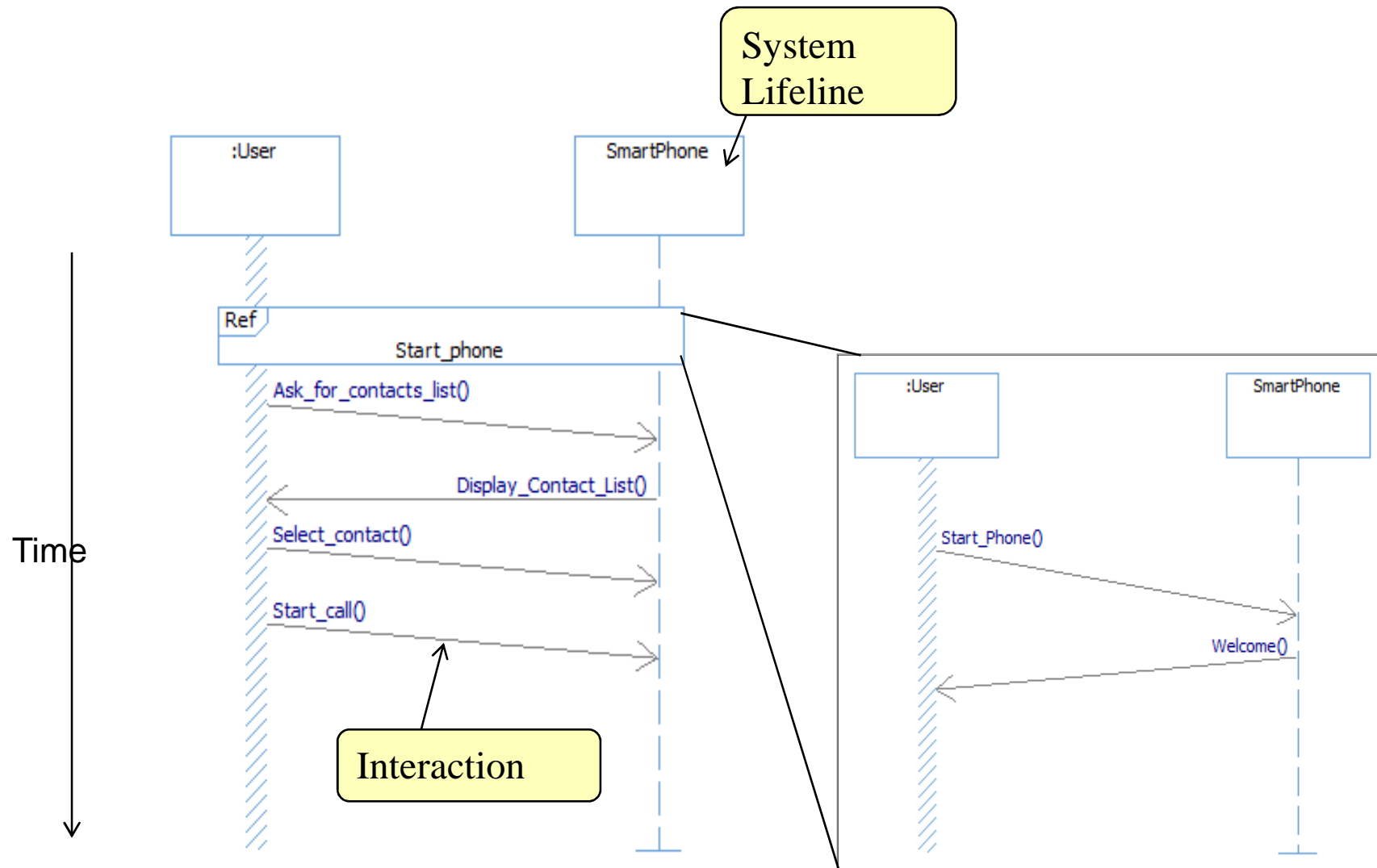
Distiller sequence diagram
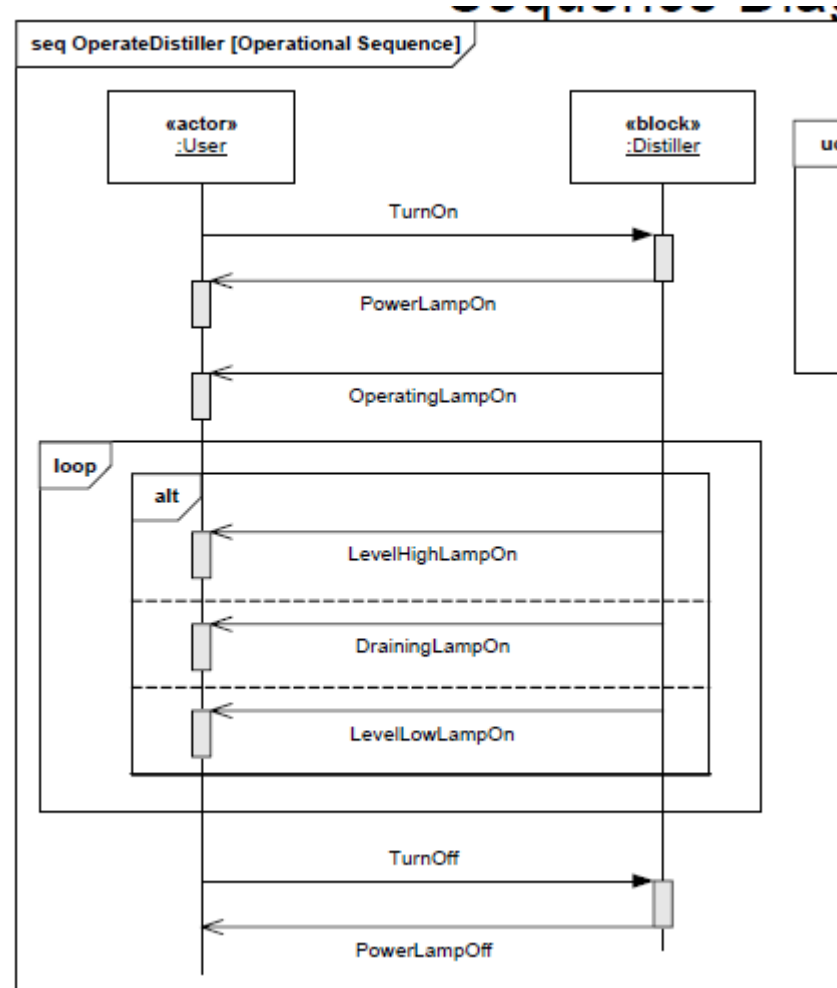
State Machine Diagram

State Machine Diagram

- ➢ Typically used to represent the life cycle of a block
- ➢ Support event-based behavior (generally asynchronous)
  - ❑ Transition with trigger, guard, action
  - ❑ State with entry, exit, and do-activity
  - ❑ Can include nested sequential or concurrent states
  - ❑ Can send/receive signals to communicate between blocks during state transitions, etc.

Statecharts (2)

▪ What is a state?

> A state is a condition of existence of an object that persists for a significant period of time

▪ What is a transition?

> A transition is a response to an event, moving the object from a state to a state.

State Machine Diagram example

Default transition

stm HSUVOperationalStates

Off — keyOff/ →

start[in neutral]/start engine

shutOff/stop engine

Nominal states only

Operate

Idle

accelerate/

when (speed = 0)

releaseBrake/

Accelerating/ Cruising

Braking

engageBrake/
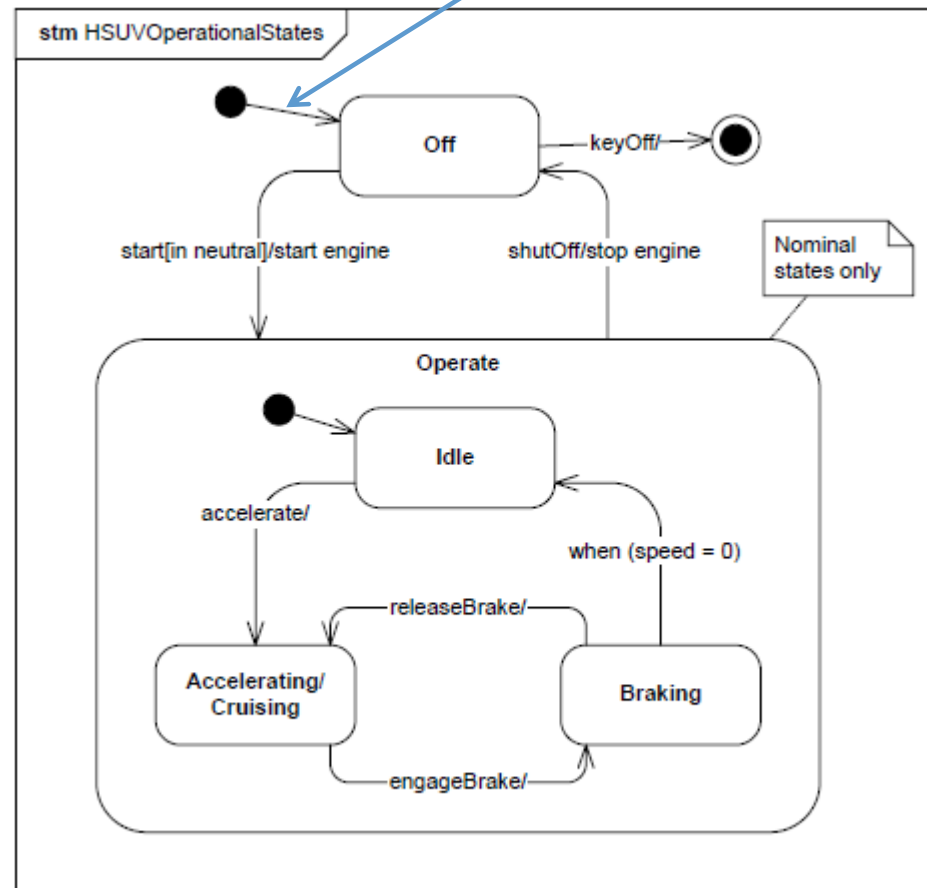
State Machine example
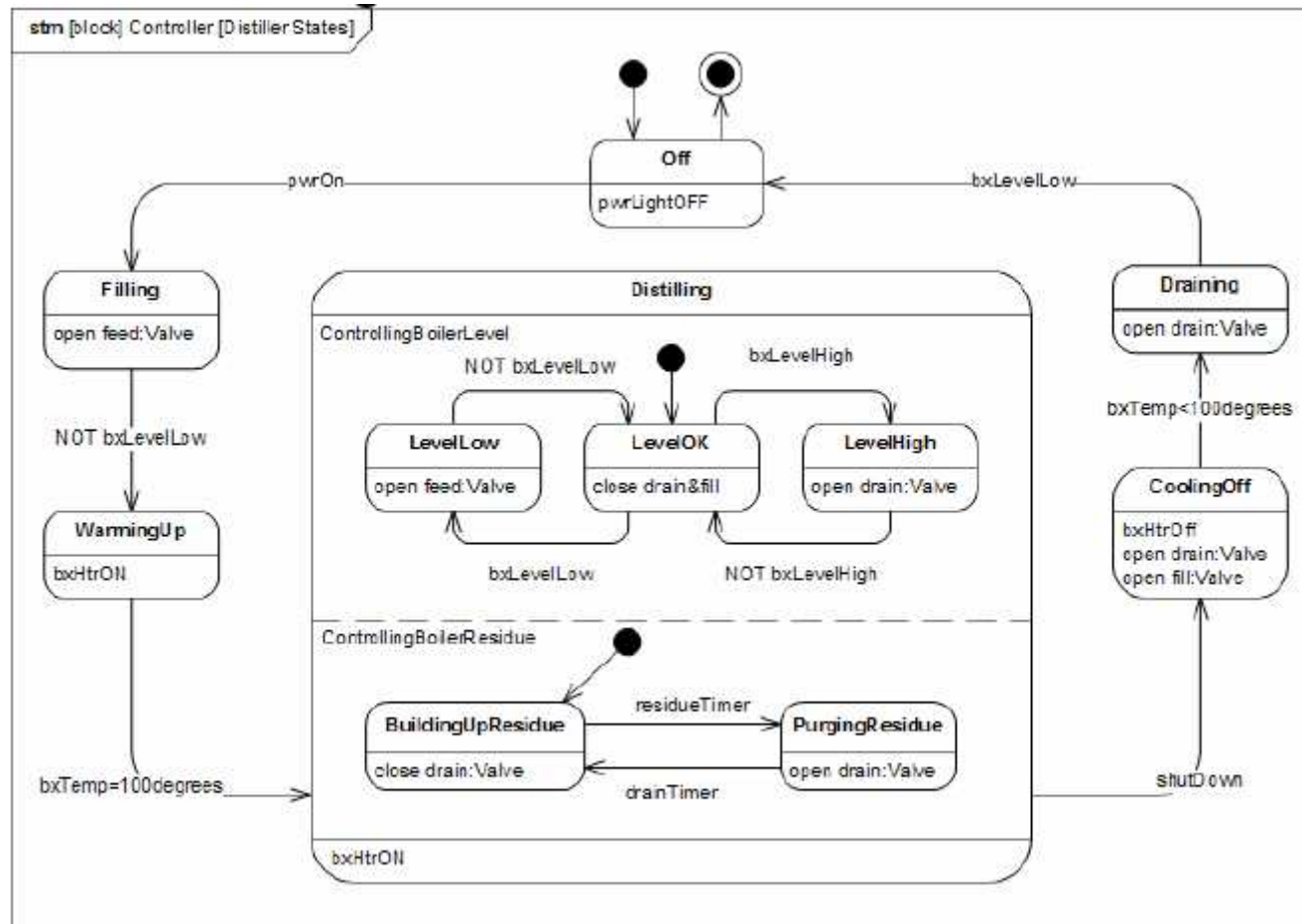
Diagrams for structure

Basic Structuring Entity: Block

- Provides a unifying concept to describe the structure of an element or system
  - Hardware
  - Software
  - Data
  - Procedure
  - Facility
  - Person

«block» BrakeModulator
allocatedFrom «activity»Modulate BrakingForce
values DutyCycle: Percentage

- Multiple compartments can describe the block characteristics
  - Properties (parts, references, values)
  - Operations
  - Constraints
  - Allocations to the block (e.g. activities)
  - Requirements the block satisfies

Using blocks

➢ Blocks are used to specify hierarchies and interconnection

➢ Block definition diagram describes the relationship among blocks (e.g., composition, association, classification)
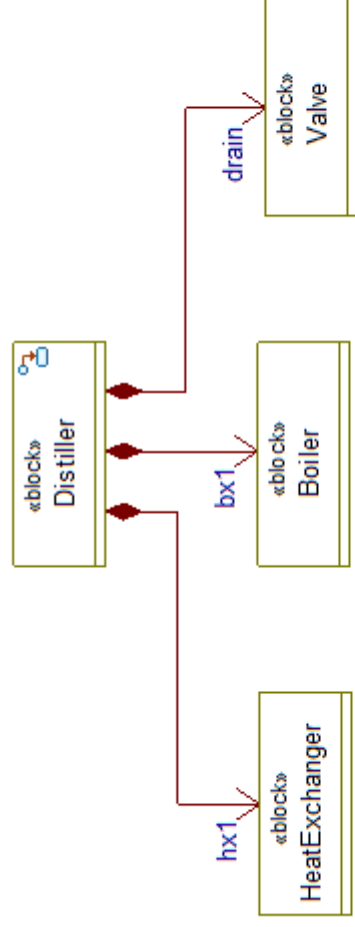
❑ Behavior can be allocated to blocks

Block Definition Diagram

➤ Type Definitions
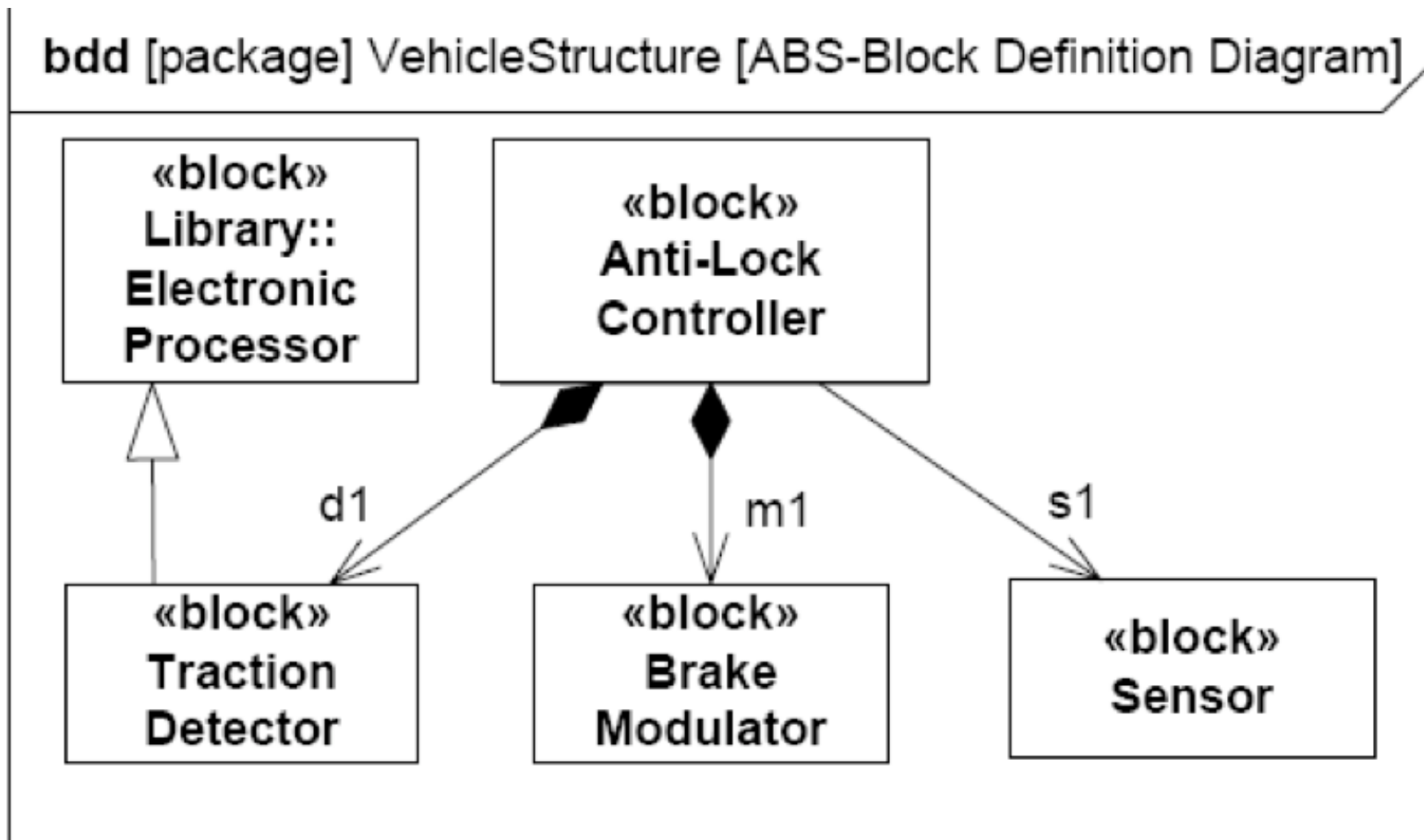
bdd [Package] Structure [bdd 1 Structural Breakdown]

«block»
Distiller

«block»
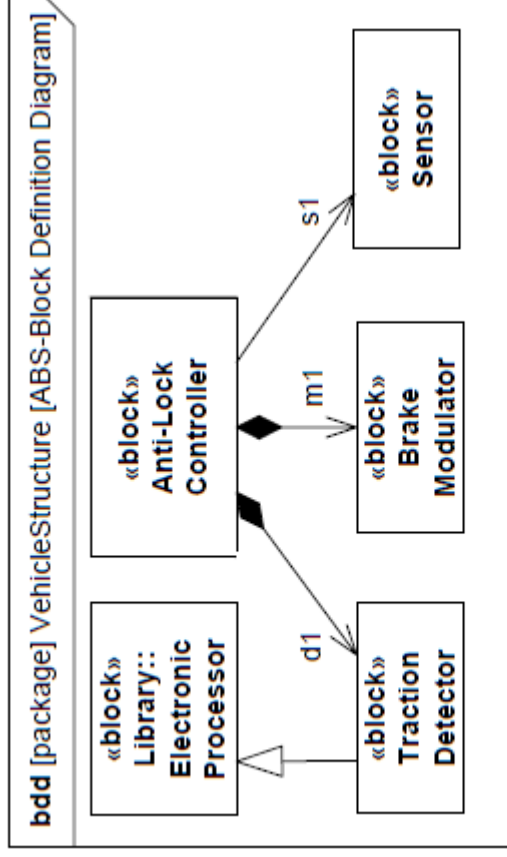HeatExchanger

«block»
Boiler

«block»
Valve

hx1

bx1

drain

Internal block diagram

➤ Internal block diagram describes the internal structure of a block in terms of its properties and connectors

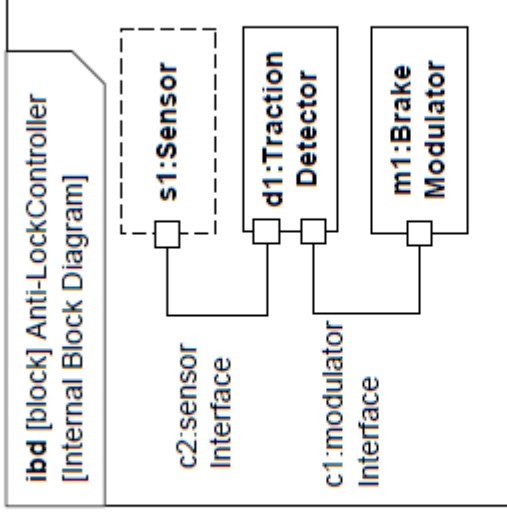# Block Definition Diagram

bdd [package] VehicleStructure [ABS-Block Definition Diagram]

«block»
Library::
Electronic
Processor

«block»
Anti-Lock
Controller

d1

m1

s1

«block»
Traction
Detector

«block»
Brake
Modulator

«block»
Sensor

Definition

- Block is a definition/type
- Captures properties, etc.
- Reused in multiple contexts

Internal Block Diagram

ibd [block] Anti-LockController
[Internal Block Diagram]

s1:Sensor

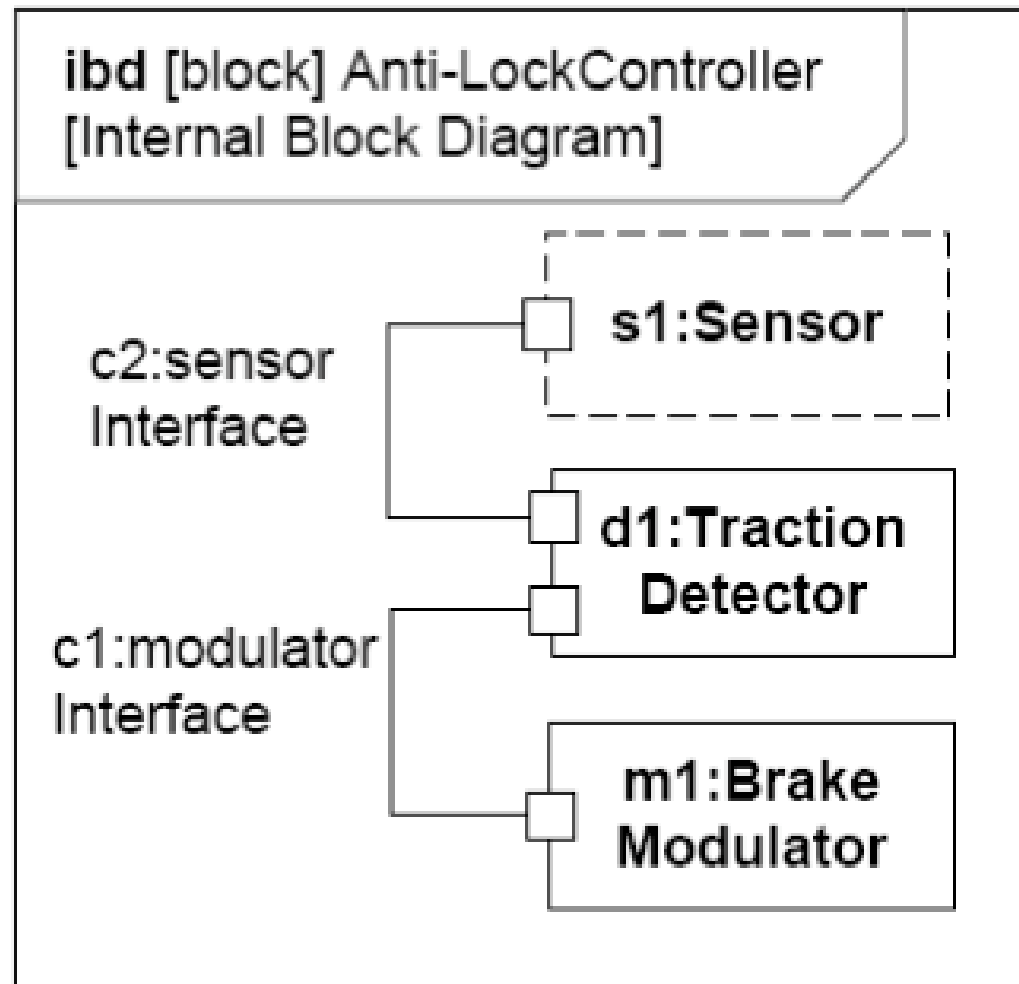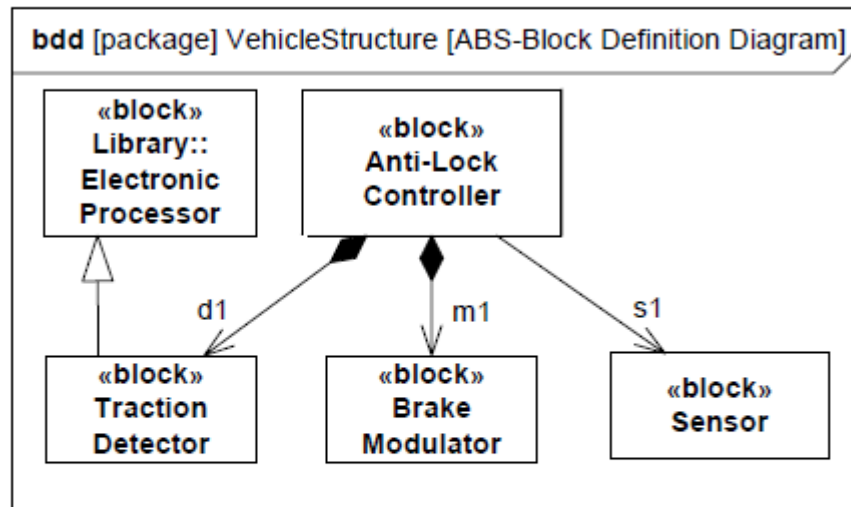d1:Traction
Detector

m1:Brake
Modulator

c2:sensor
Interface

c1:modulator
Interface

Usage

- Part is the usage in a particular context
- Typed by a block
- Also known as a role

Internal Block Diagram

ibd [block] Anti-LockController
[Internal Block Diagram]

c2:sensor Interface

s1:Sensor

c1:modulator Interface

d1:Traction Detector

m1:Brake Modulator

Relation between BDD and IBD

Block Definition Diagram

bdd [package] VehicleStructure [ABS-Block Definition Diagram]

«block»
Library::
Electronic
Processor

«block»
Anti-Lock
Controller
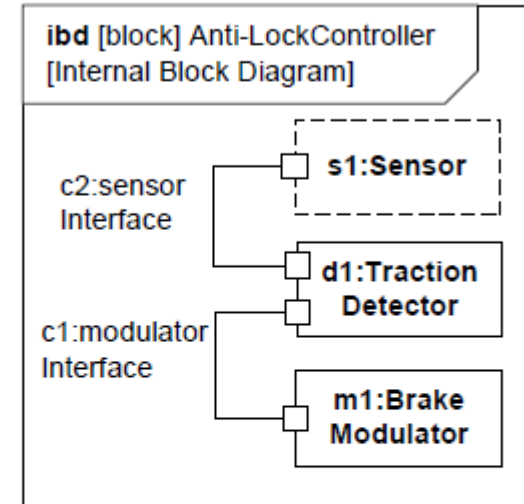
d1

m1

s1

«block»
Traction
Detector

«block»
Brake
Modulator

«block»
Sensor

Definition

– Block is a definition/type
– Captures properties, etc.
– Reused in multiple contexts

Internal Block Diagram

ibd [block] Anti-LockController
[Internal Block Diagram]

c2:sensor
Interface

s1:Sensor

d1:Traction
Detector

c1:modulator
Interface

m1:Brake
Modulator

Usage

– Part is the usage in a particular context
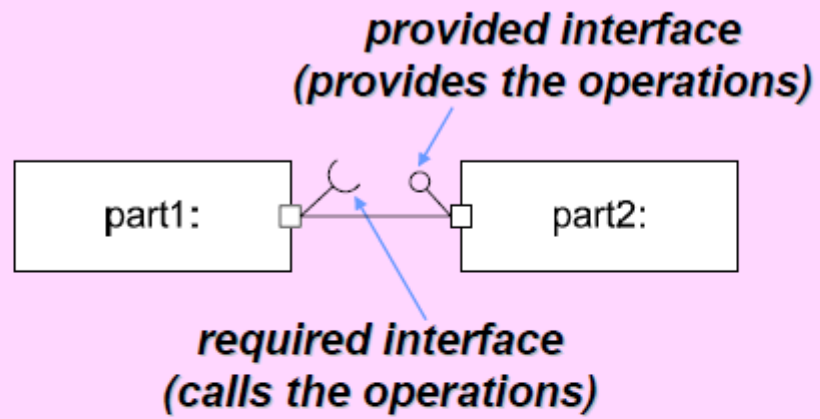– Typed by a block
– Also known as a role

Ports

- Specifies interaction points on blocks and parts
  - Supports integration of behavior and structure
- Port types
  - Standard (UML) Port
    - Specifies a set of operations and/or signals
    - Typed by a UML interface
  - Flow Port
    - Specifies what can flow in or out of block/part
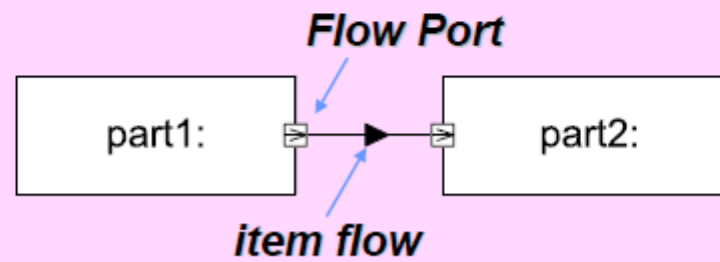   - Typed by a flow specification

Ports

Delegation through ports

- Delegation can be used to preserve encapsulation of block
- Interactions at outer ports of Block1 are delegated to ports of child parts
- Ports must match (same kind, types, direction etc.)
- (Deep-nested) Connectors can break encapsulation if required (e.g. in physical system modeling)

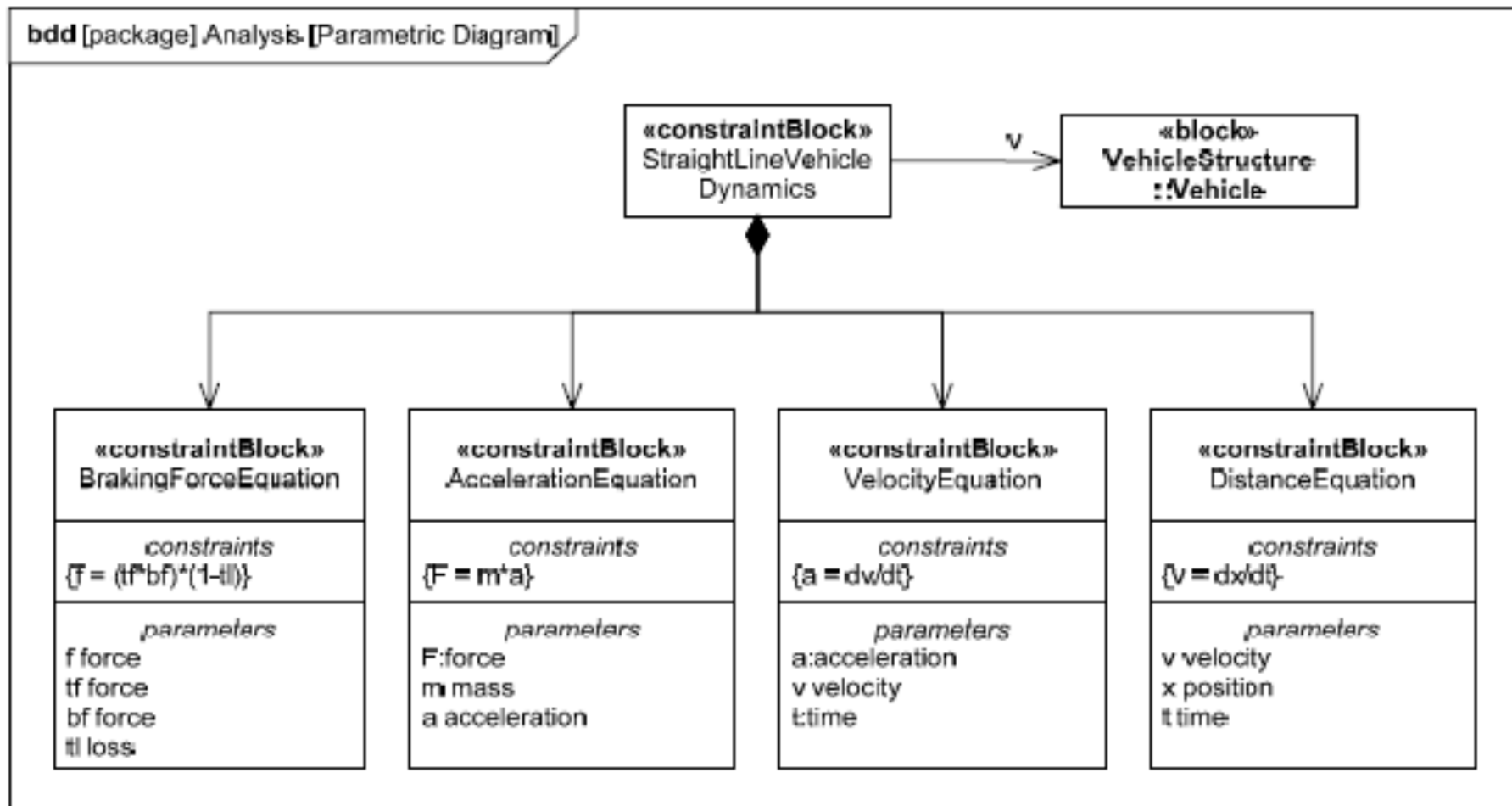ibd [block]Block1[delegation]

Child1:

Child2:
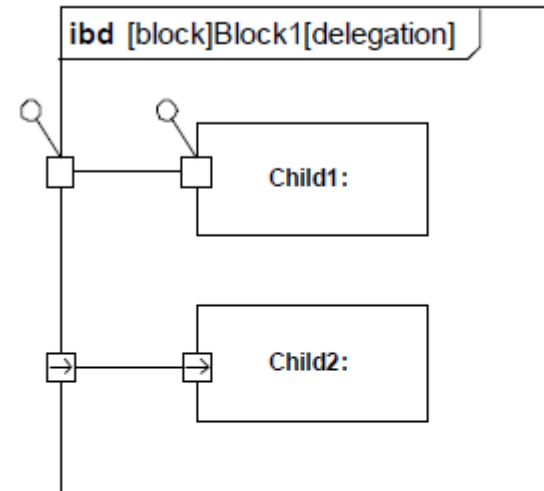
Parametric diagrams

Parametric Diagrams

➤ Used to express constraints (equations) between value properties
  ❑ Provides support for engineering analysis (e.g., performance, reliability)
➤ Constraint block captures equations
  ❑ Expression language can be formal (e.g., MathML, OCL) or informal
  ❑ Computational engine is defined by applicable analysis tool and not by SysML
➤ Parametric diagram represents the usage of the constraints in an analysis context
  ❑ Binding of constraint usage to value properties of blocks (e.g., vehicle mass bound to $F = m \times a$)
➤ Parametrics Enable Integration of Engineering Analysis with Design Models

Defining Vehicle Dynamics

Parametric Diagram

par [constraintBlock] StraightLineVehicleDynamics [Parametric Diagram]

v.chassis.tire.Friction:

v.brake.abs.m1.DutyCycle:

v.brake.rotor.BrakingForce:

v.Weight:

tf: tl: bf: m:

:BrakingForce Equation
$\{f = (tf*bf)*(1-tl)\}$

f:

:Accelleration Equation
$\{F = m*a\}$

F:

a:

a:

:DistanceEquation
$\{v = dx/dt\}$

v:

:VelocityEquation
$\{a = dv/dt\}$

v:

x:

v.Position: