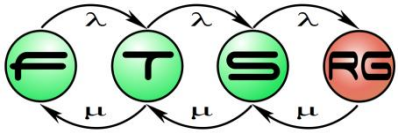
The System Modeling Language (SysML) and the SYSMOD modeling approach

Polgár Balázs

Ákos Horváth

Model Driven Software Development

Lecture 10



Acknowledgement

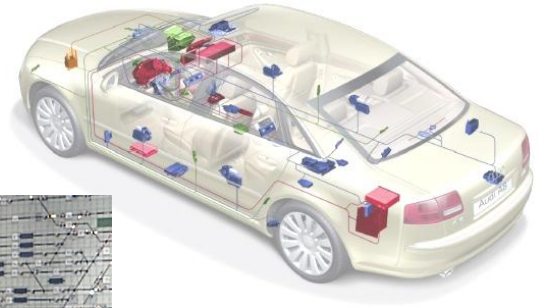
- Portions of this presentation are from
  - *Systems Engineering with SysML/UML*, by Tim Weilkiens, published by Morgan Kaufmann Publishers, Copyright 2007 Elsevier Inc. All rights reserved.
  - *A Practical Guide to SysML*, by Sanford Friedenthal, Alan Moore, and Rick Steiner, published by Morgan Kaufmann Publishers, Copyright 2009 Elsevier Inc. All rights reserved.
  - IBM course, Requirements management

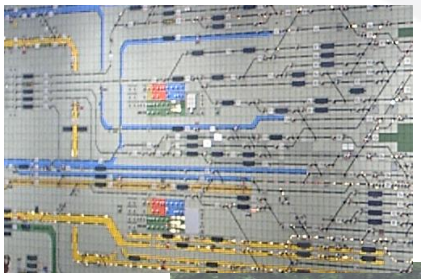
Overview

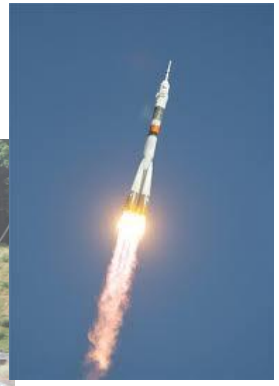
- Context
- SysML Overview
- SysML details + the SYSMOD Systems Engineering Methodology

Systems Engineering

- Systems Engineering is a multidisciplinary approach to develop balanced system solutions in response to diverse stakeholder needs
- ~ Integration Engineering
  - Software engineering
  - Hardware engineering
  - Mechanical engineering
  - Safety engineering
  - Security engineering
  - …
- ~ Process Engineering
- System
  - Military, airplane, car, aviation, railway interlocking, notebook, etc.

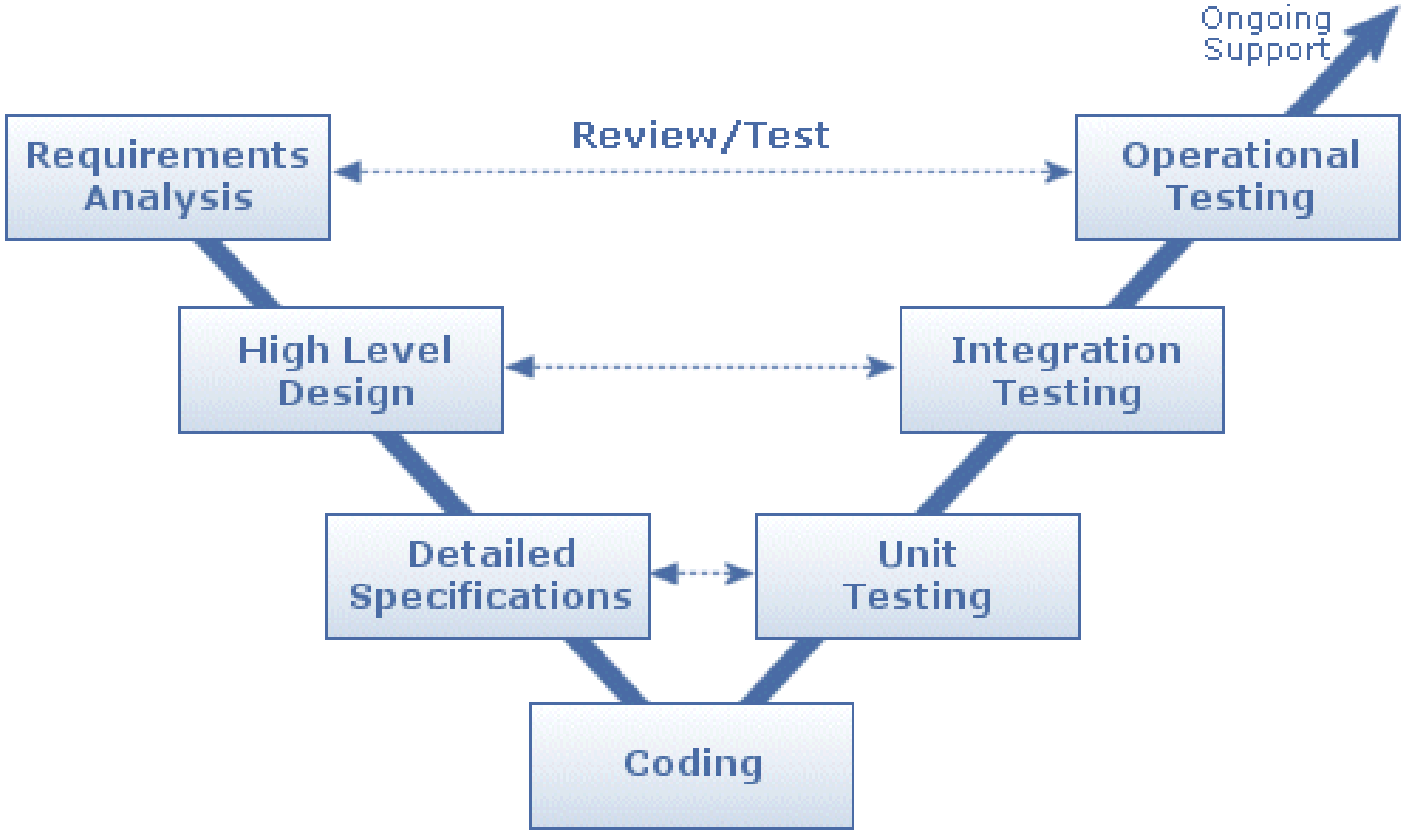




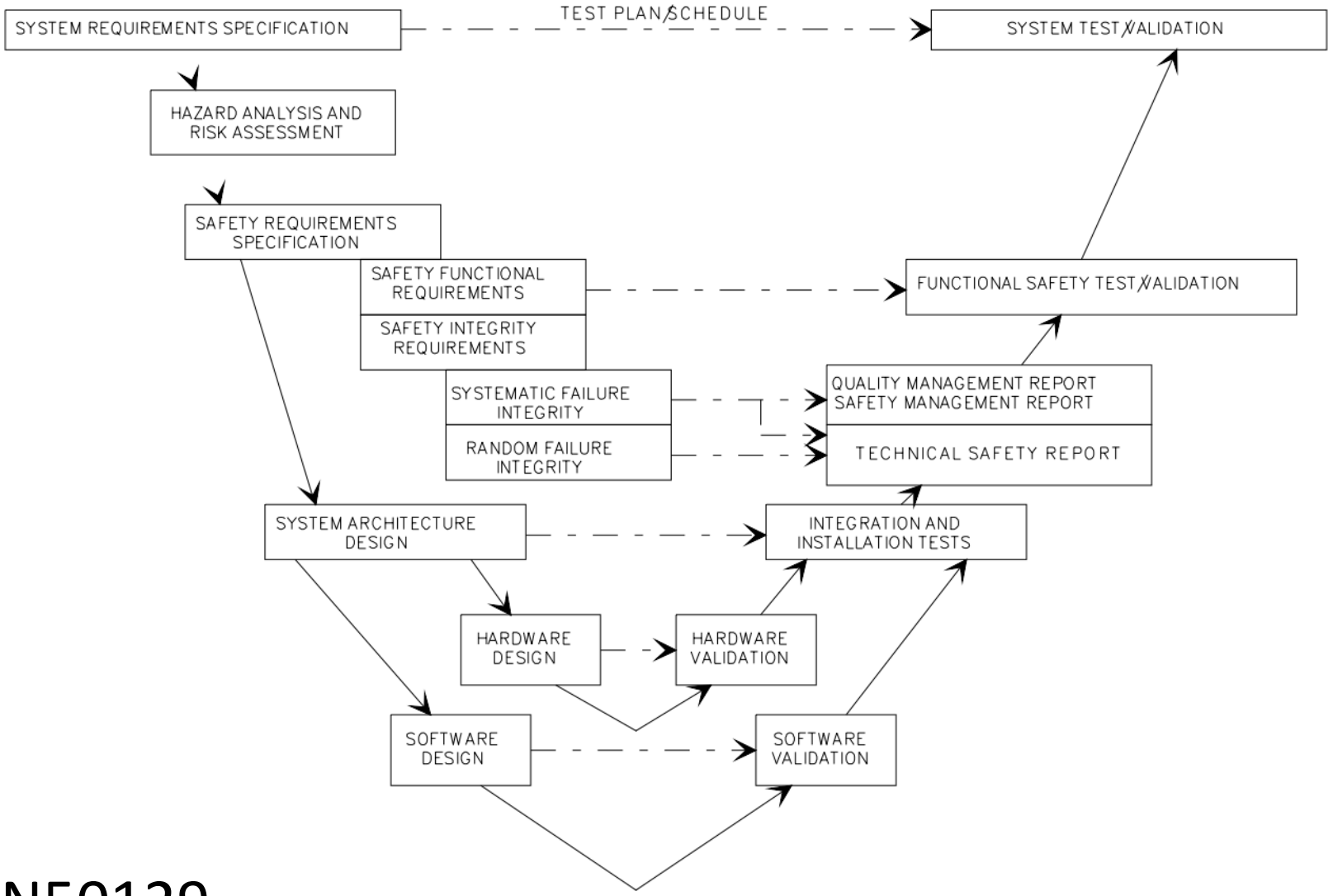


Systems Engineering Process

- V-model

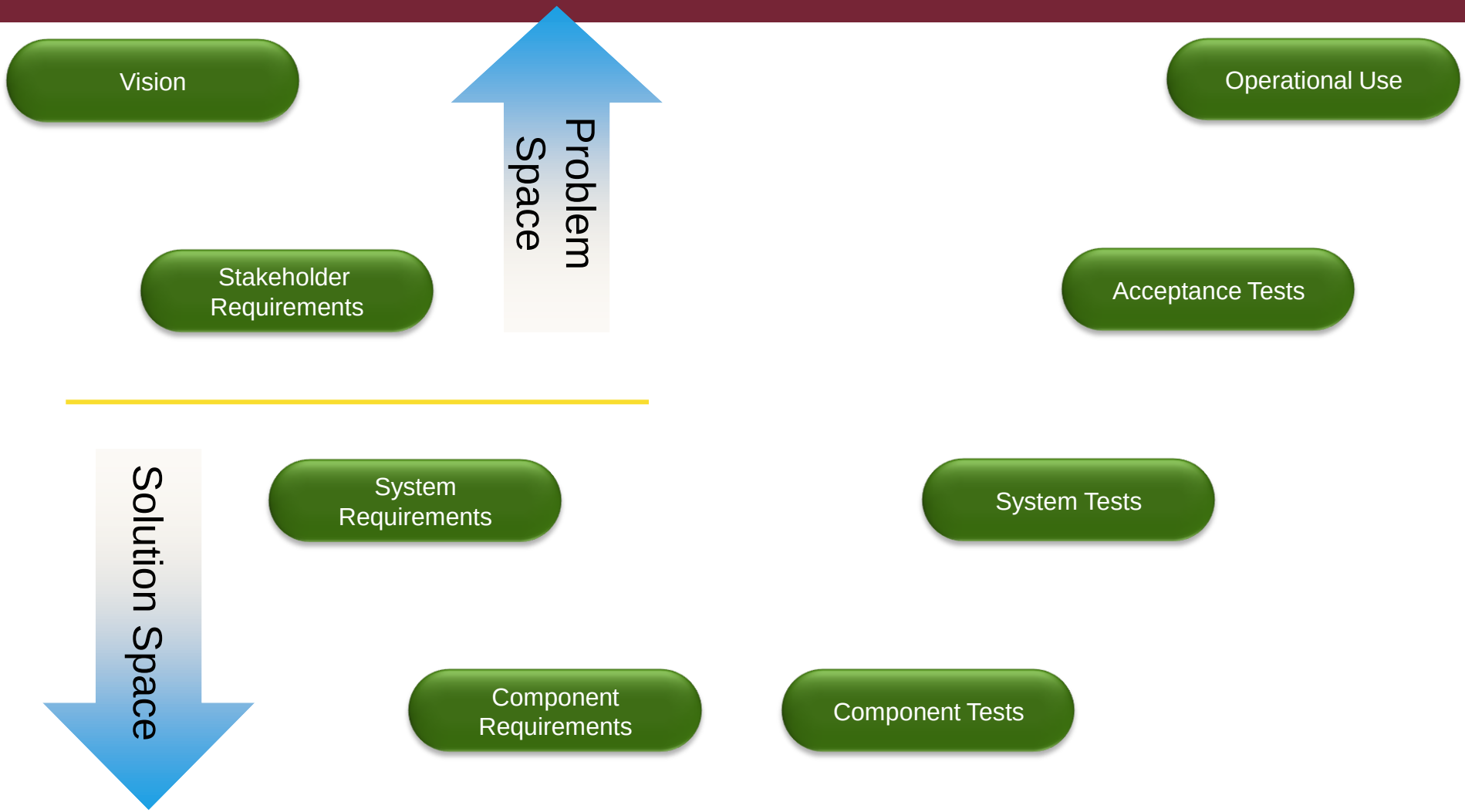


V-model – v2



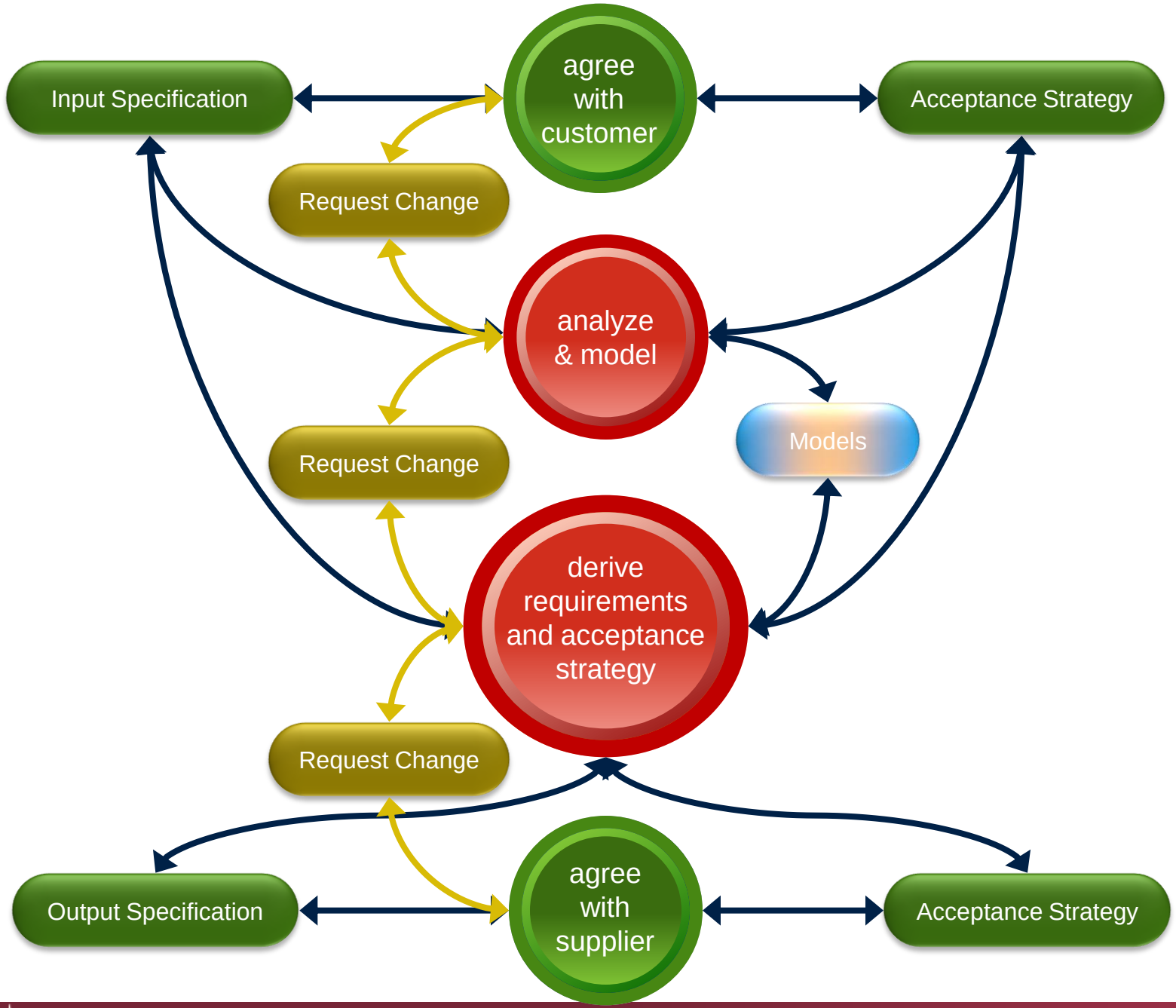
EN50129

V-model – v3


Vision
Problem Space
Operational Use
Stakeholder Requirements
Acceptance Tests
Solution Space
System Requirements
System Tests
Component Requirements
Component Tests

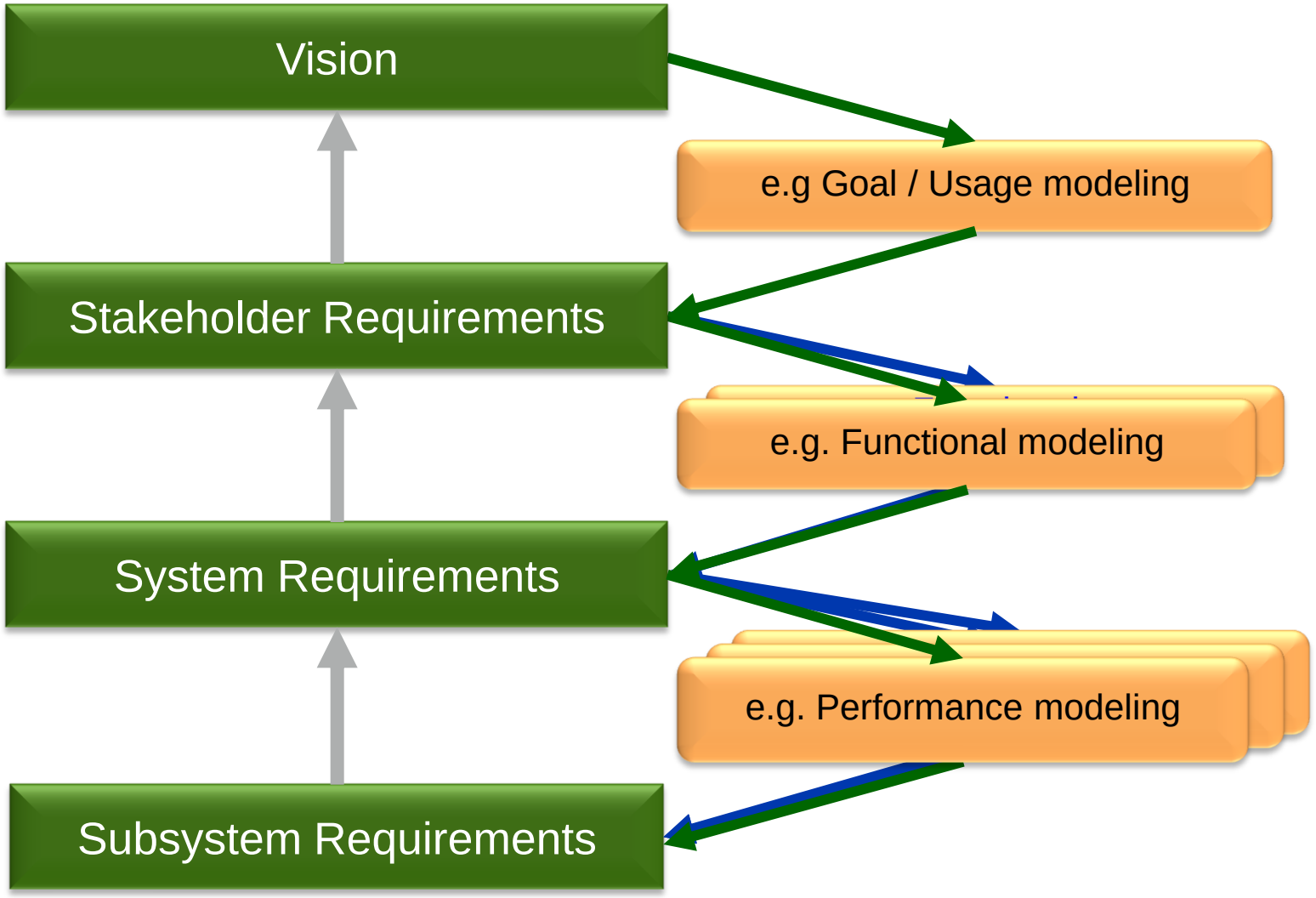
Differentiating Problem and Solution

Problem	*Solution*
Stakeholder requirements	*System requirements*
• A description of the problem and its context	• An *abstract* representation of the solution
• Describes what stakeholders want from the system	• Describes what the system will do
• Not the definition of the solution (except for environment)	• Not the definition of the design
• Quality of results	• How well it does it
• Created by stakeholders	• Created by systems engineers
"The user shall be able to ...."	*"The system shall do ...."*

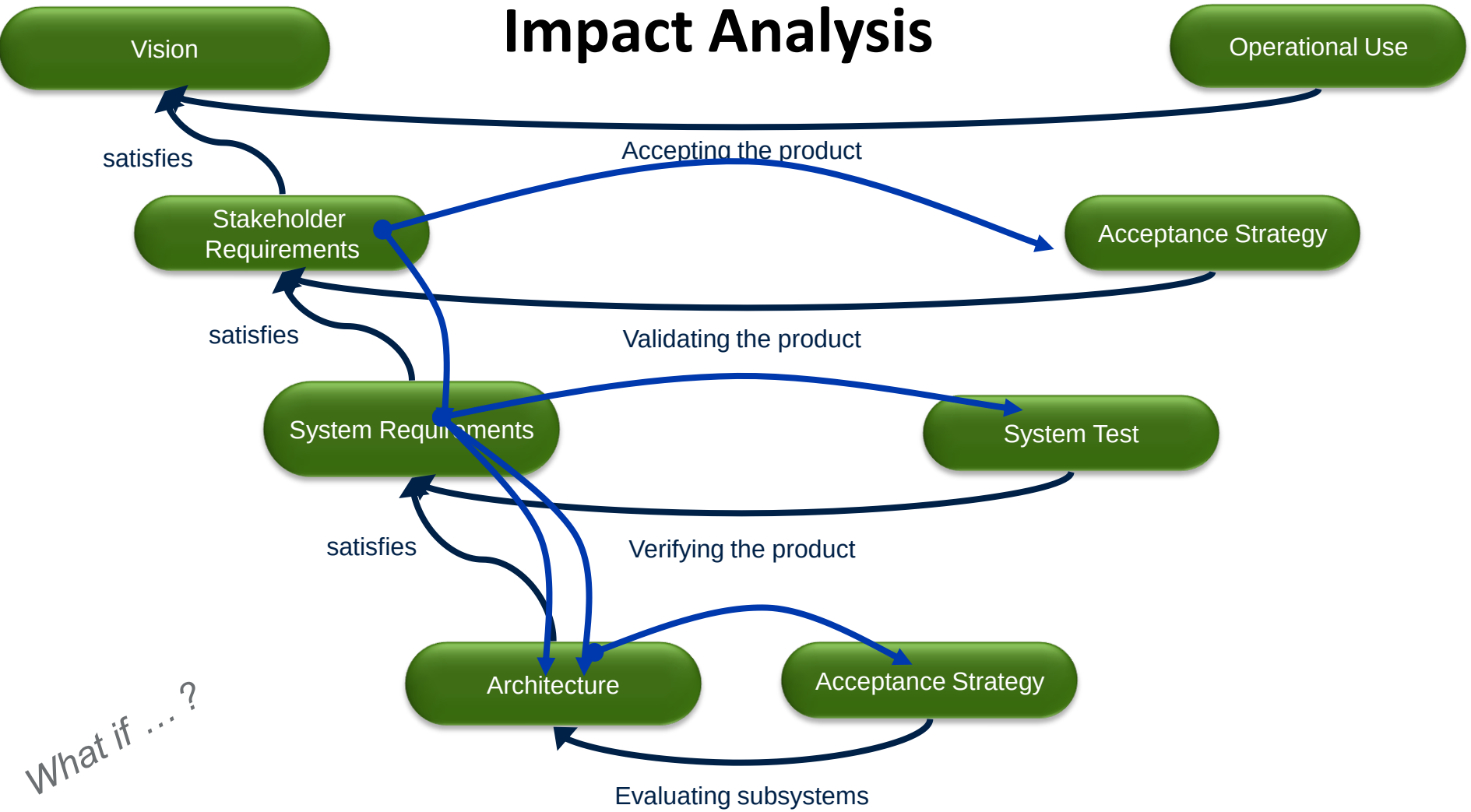

Input Specification
agree with customer
Acceptance Strategy
Request Change
analyze & model
Models
Request Change
derive requirements and acceptance strategy
Request Change
Output Specification
agree with supplier
Acceptance Strategy


agree
Vision
Operational Use
Request Change
Stakeholder
Requirements
Acceptance Tests
Models
System Requirements
System Tests
strategy
Component
Requirements
Component Tests
Output Specification
agree
with
supplier
Acceptance Strategy

Models Bridge Layers of Requirements

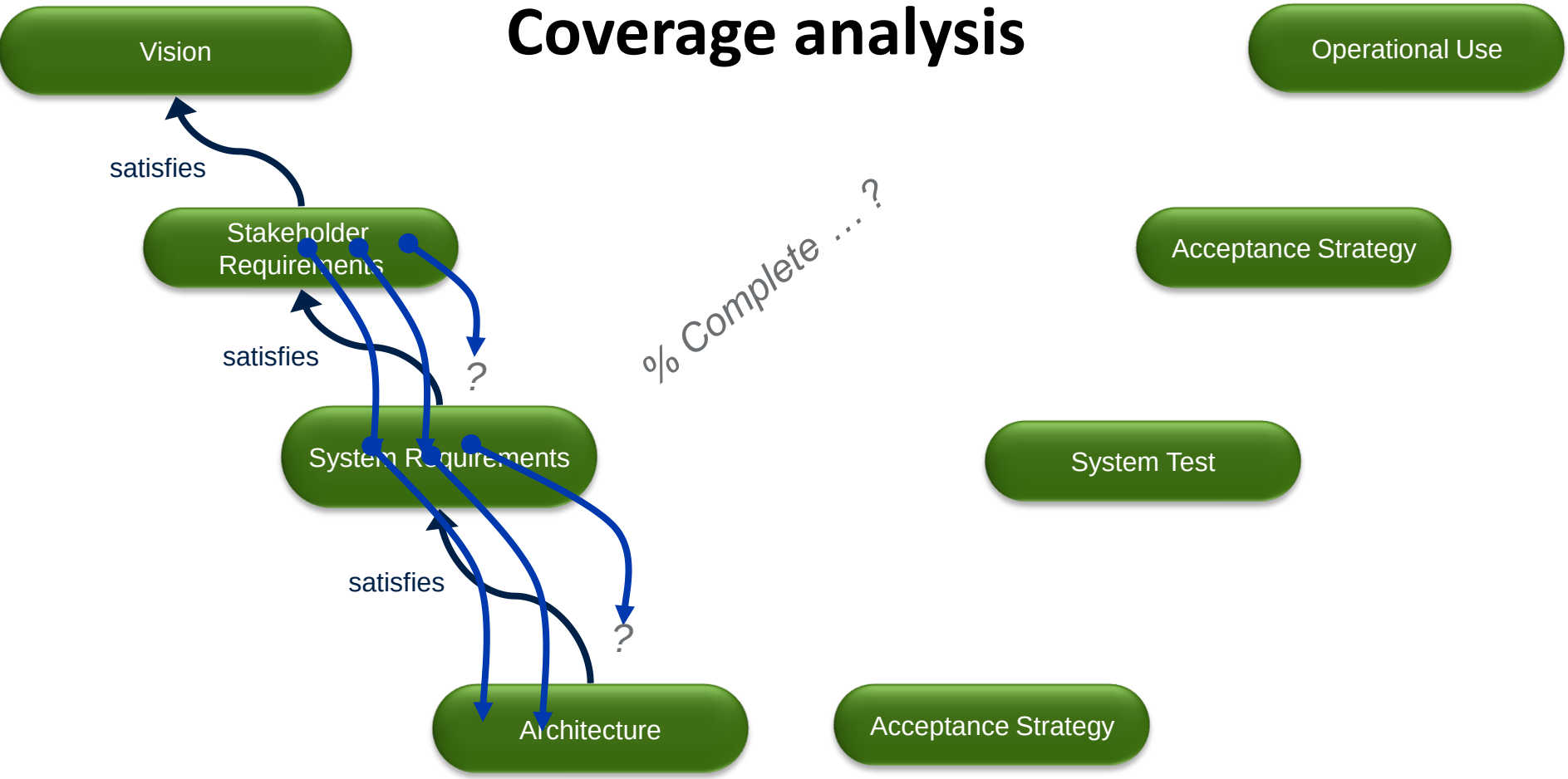


Importance of Traceability



Importance of Traceability

Coverage analysis



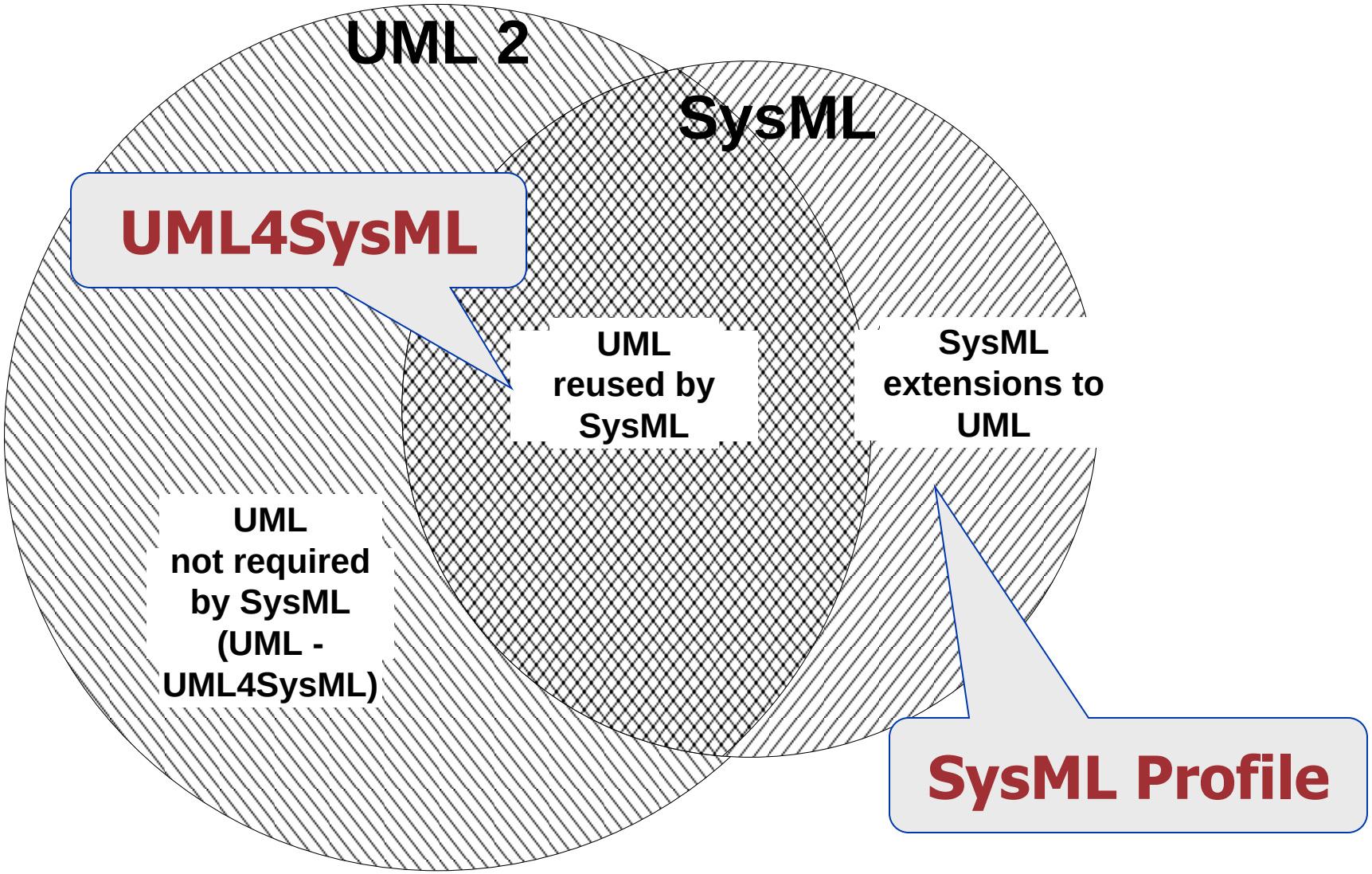
Overview

- Context
- **SysML Overview**
- SysML details + the SYSMOD Systems Engineering Methodology

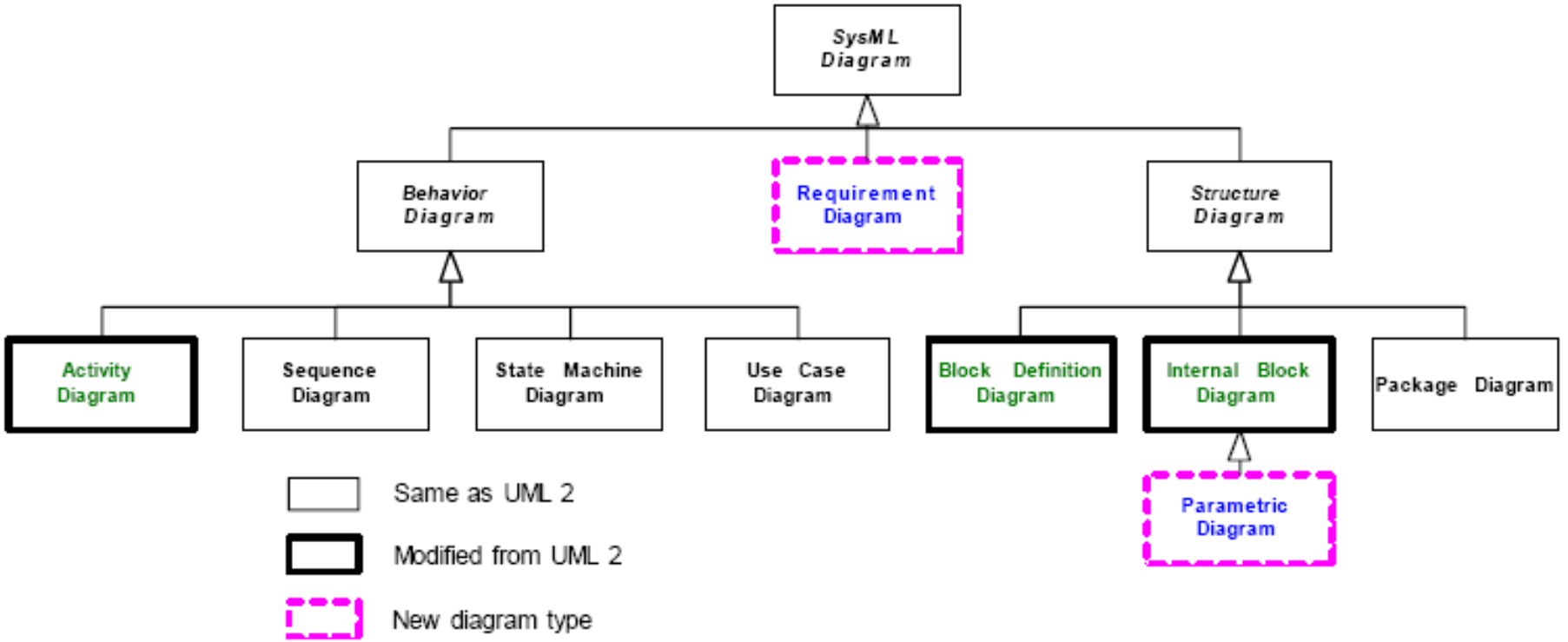
SysML overview

- „UML for Systems Engineering”
  - Supports the specification, analysis, design, verification and validation of systems that include hardware, software, data, personnel, procedures, and facilities
- Developed by OMG and International Council on Systems Engineering (INCOSE)
- OMG SysML™ (http://www.omgsysml.org)
  - RFP – March 2003
  - Version 1.0 – September 2007
  - Version 1.1 – November 2008
  - Version 1.2 – June 2010
  - Version 1.3 – June 2012

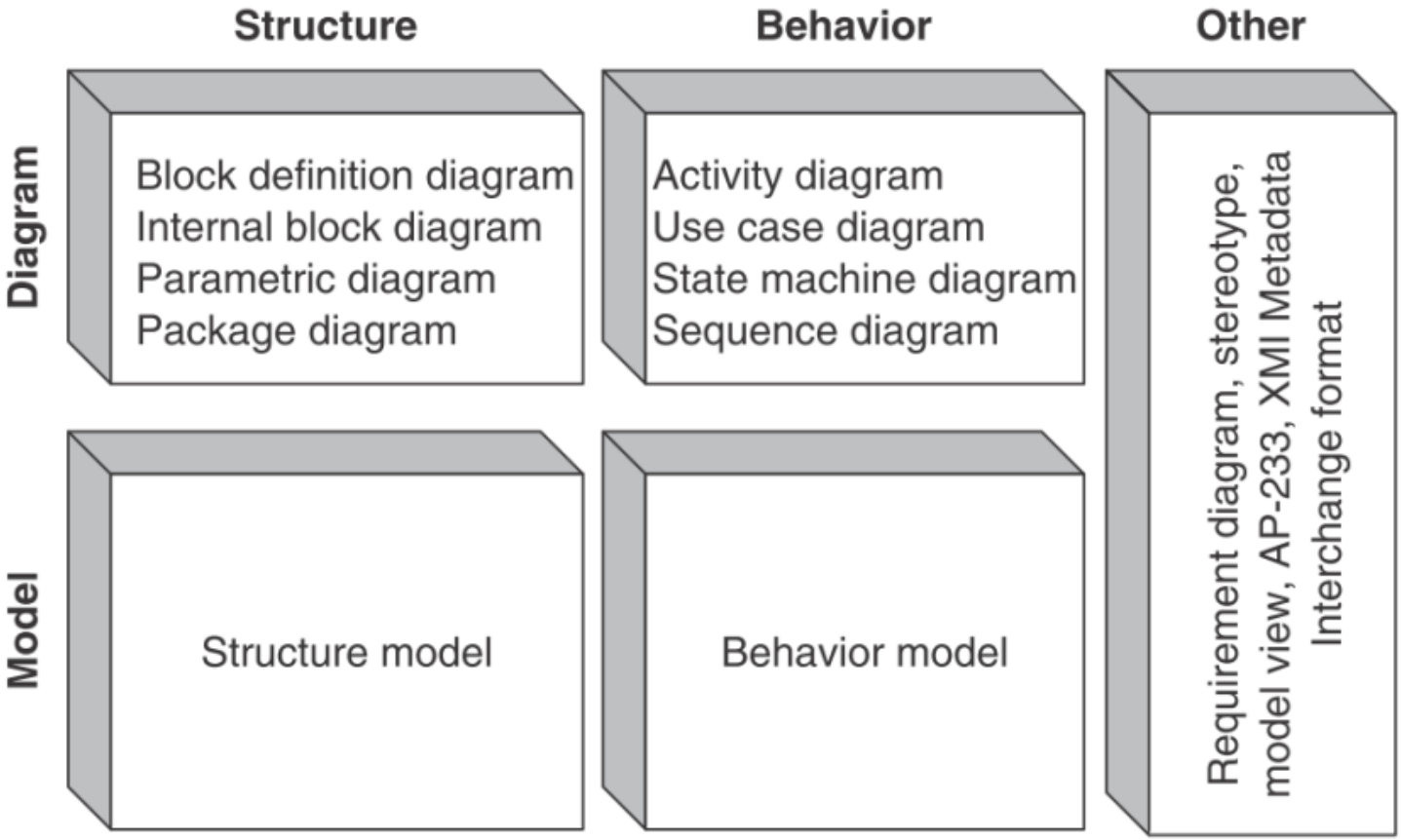
Relationship Between SysML and UML



SysML Diagram Taxonomy



Aspects of SysML



Overview

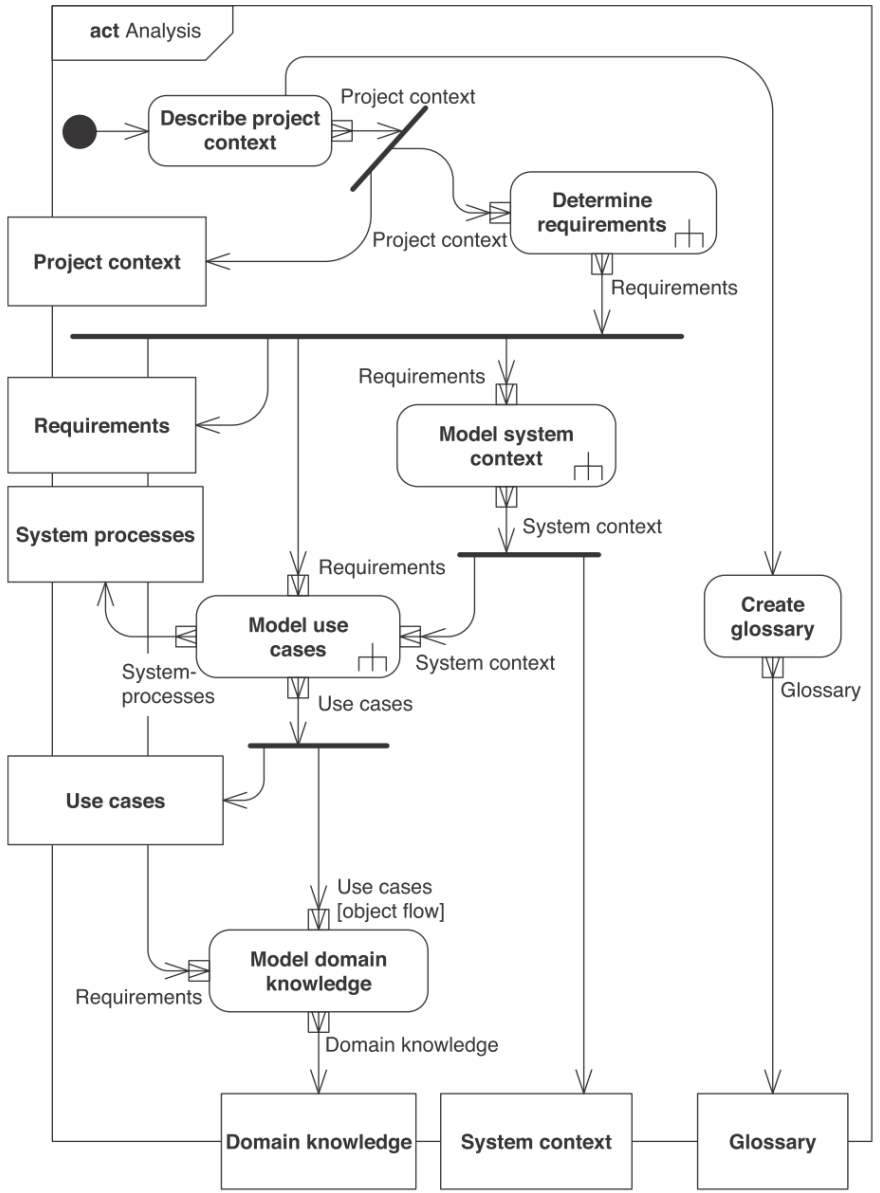
- Context
- SysML Overview
- **SysML details + the SYSMOD Systems Engineering Methodology**

Language vs. Methodology

- Modeling Language
  - Defines elements and their relationship
  - Defines syntax and semantics
  - *What type of elements can be used during modeling?*
  - E.g. SysML
- Development Methodology
  - Defines the steps of analyzing and designing the system
  - Defines the usage of the model elements and diagrams
  - *How shall the model be built?*
  - E.g. SYSMOD (SYStem MODeling) by Tim Weilkiens

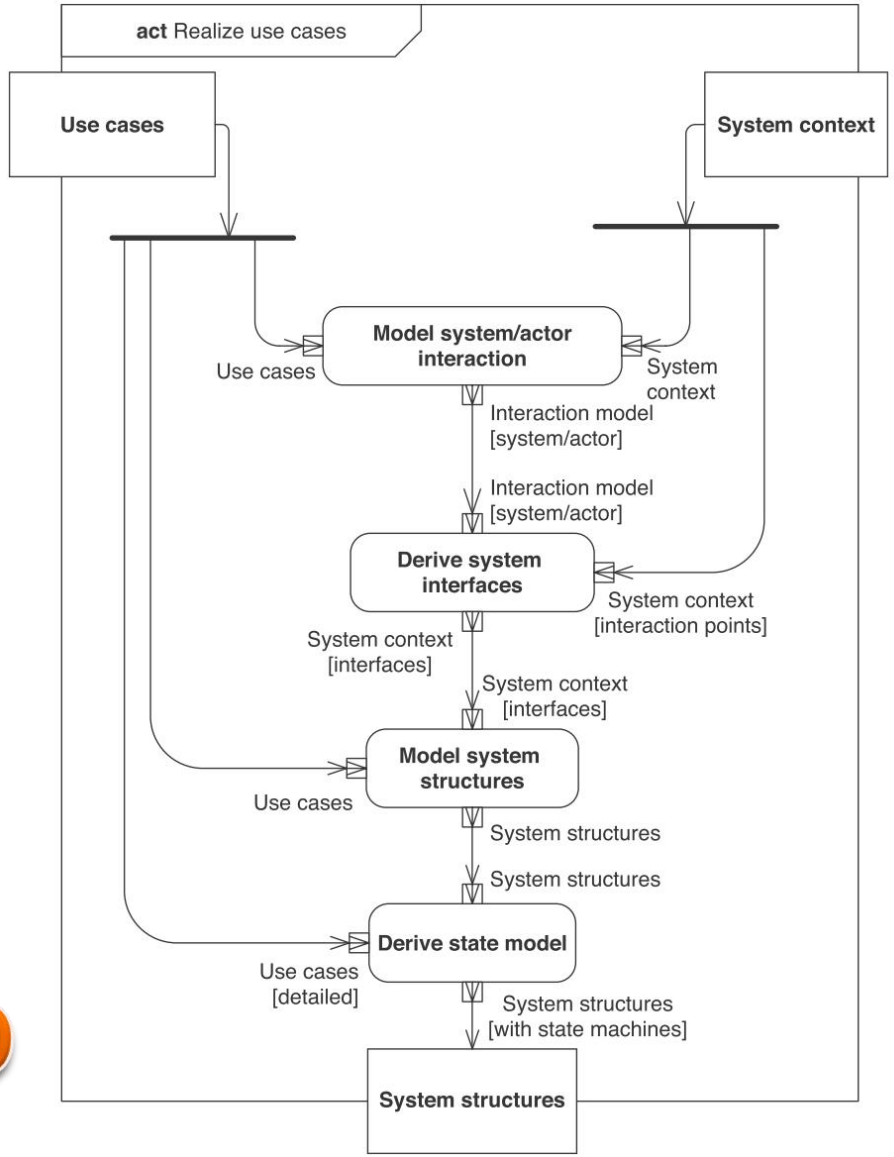
The SYSMOD approach

- Analyzing Requirements



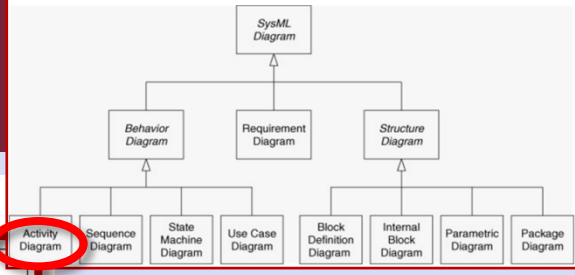
SYSMOD

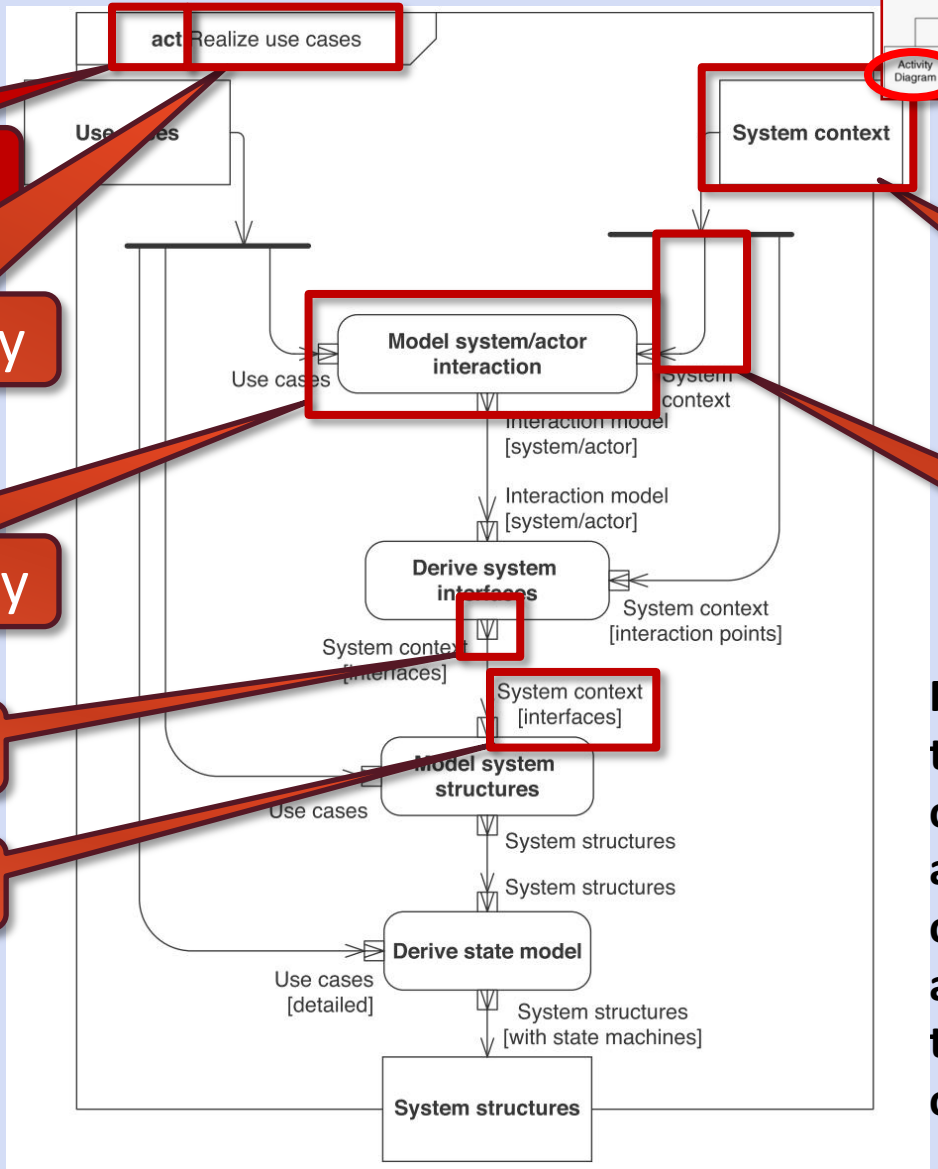
The SYSMOD approach for design



SYSMOD

SysML





Activity diagram

Name of activity

Action in activity

Pin

Object

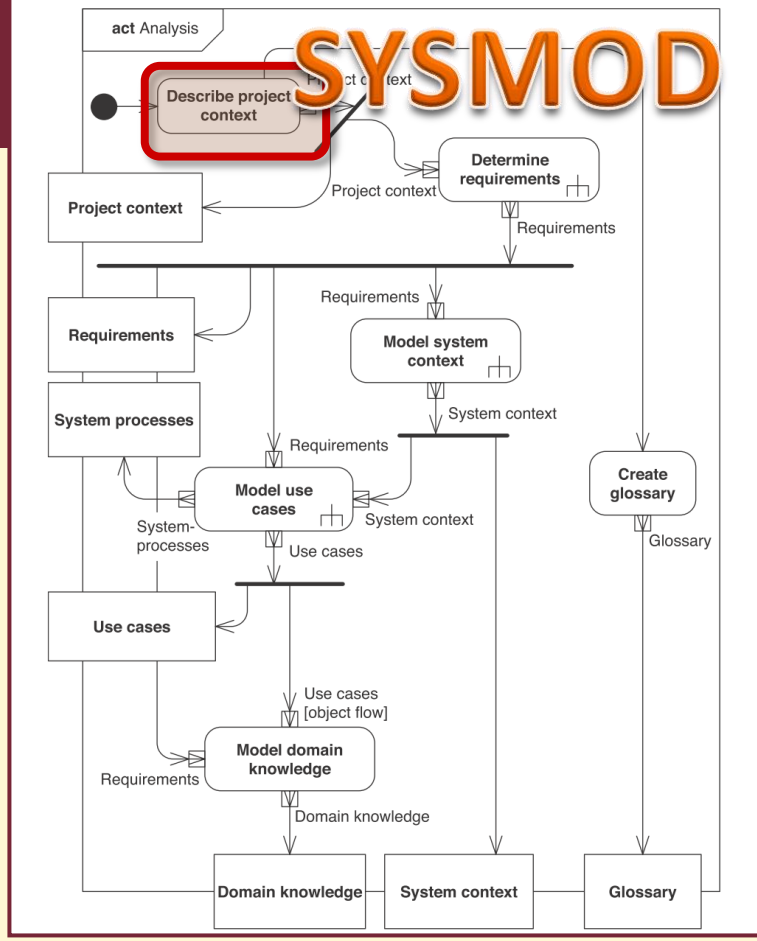
Parameter of activity

Object flow

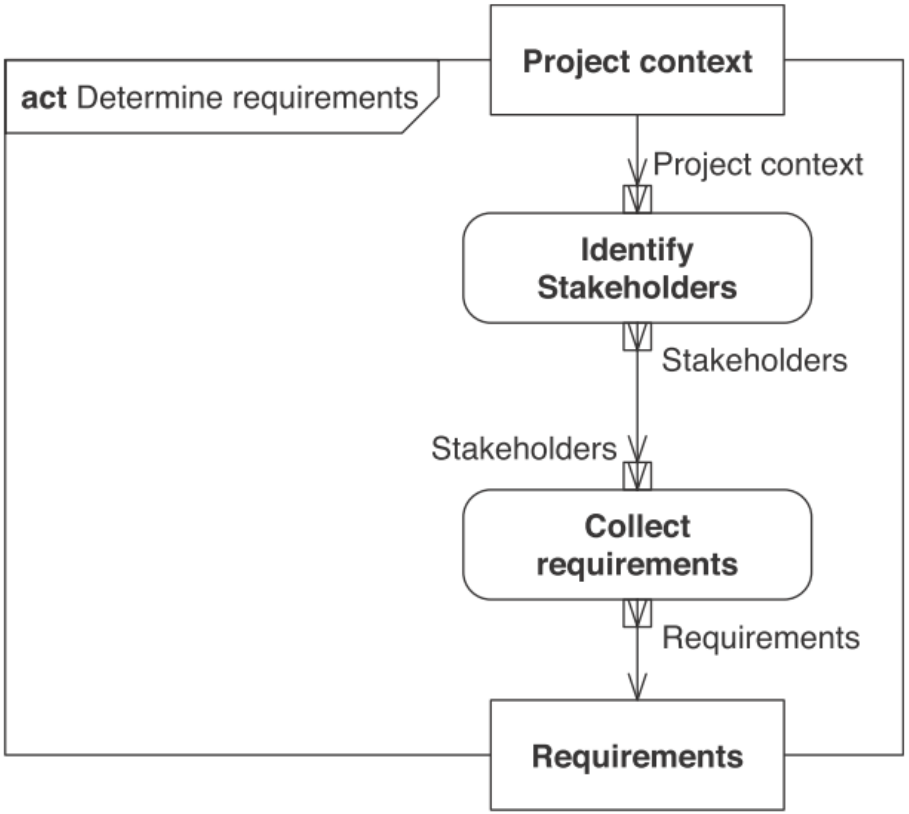
Represents behavior in terms of the ordering of actions based on the availability of inputs, outputs, and control, and how the actions transform the inputs to outputs

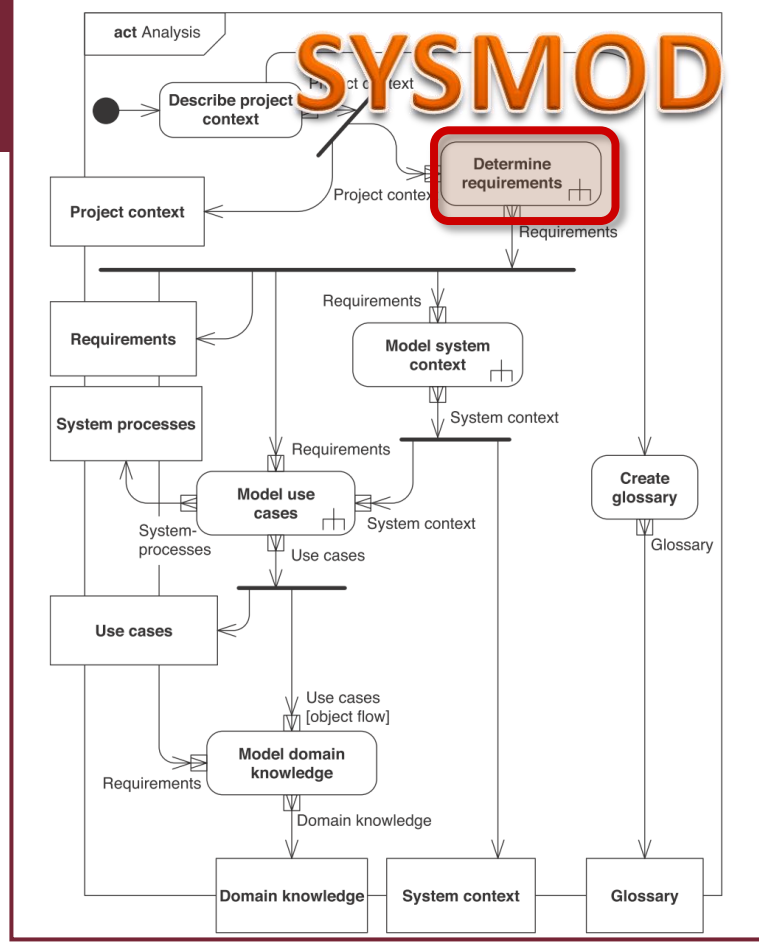
Describe Project Context

- Car rental system
  - Works without staff
    - Customer identification needed
  - Central computer in radio compartment
    - Communicates with central reservation system
    - Collects usage data
    - Comfort features
      - Navigation
      - Radio
      - Phone
      - …



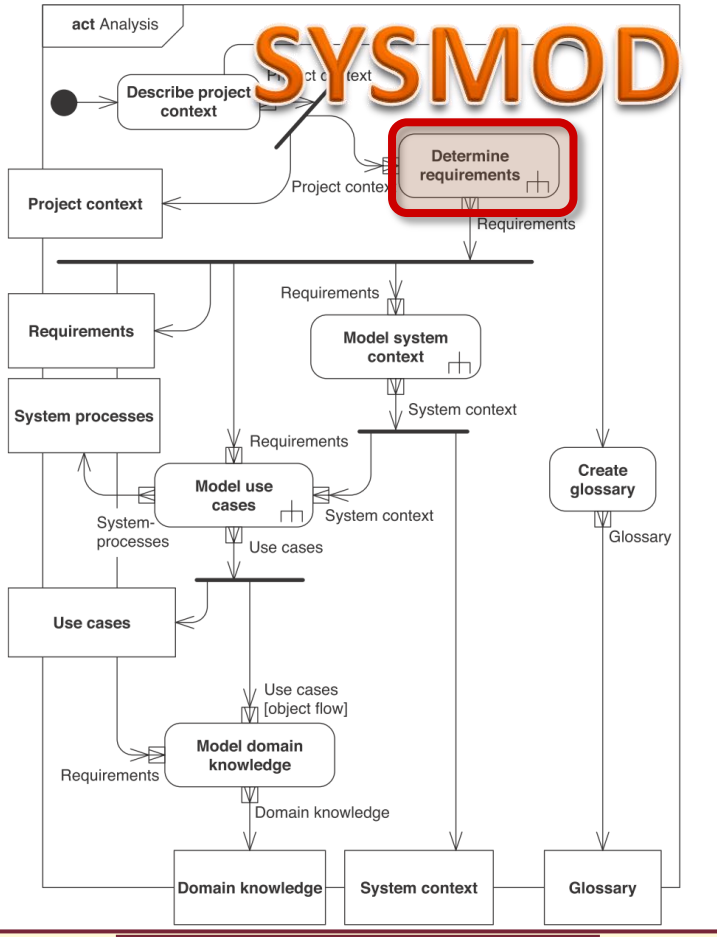
Determine requirements

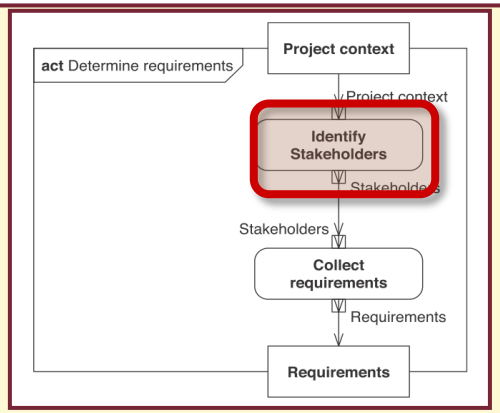




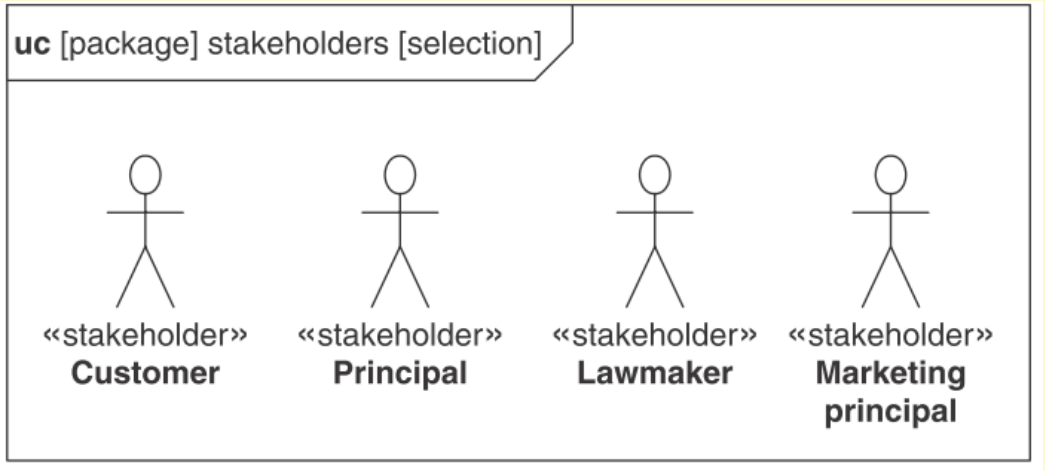
Identify stakeholders

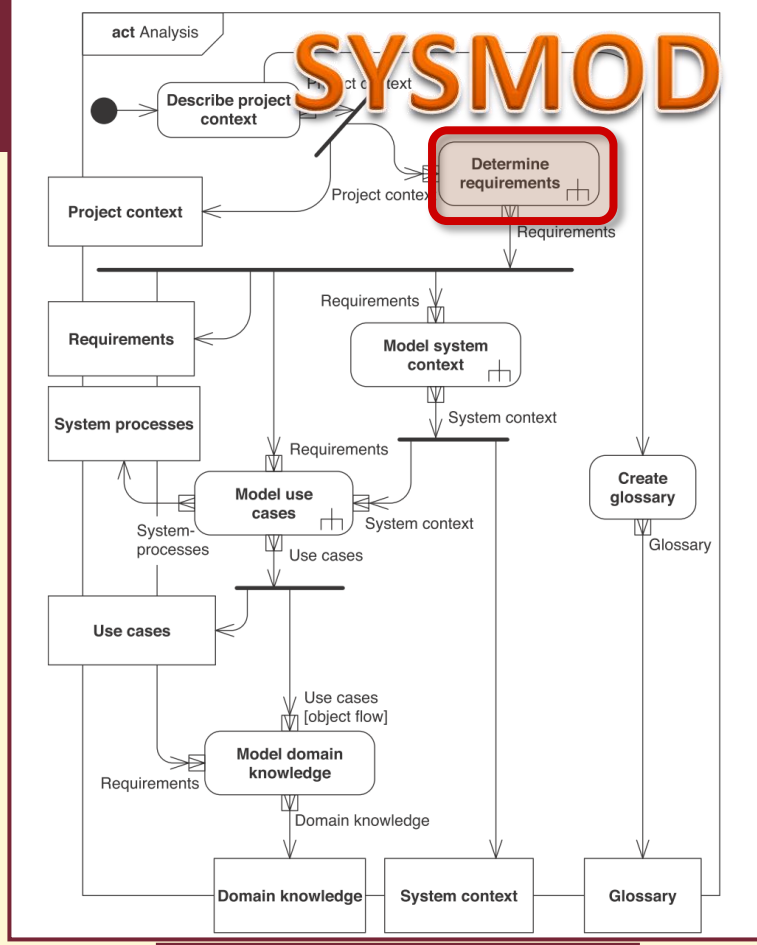
Stakeholder	Priority (1–4)	Comments/Interests
Customer	1	Wants easy and comfortable access to a car and low prices.
Reservation system	2	Requires interface to the on-board computer.
Car manufacturer	1	The on-board computer must control the central locking system and the drive-away protection, and collect mileage information.
Cellular communication vendor	1	The on-board computer and the reservation system will presumably communicate via SMS. Both speed and availability must be ensured.
Insurance company	1	Is break-in protection coverage for the on-board computer sufficient?
Car service	2	Installation, maintenance, and configuration of the on-board computer.
SpeedyCar call center	2	Handles customer enquiries with regard to the on-board computer's operation.
Navigation system manufacturer	4	SpeedyCar wants the on-board computer to have navigation system functionality.
Car radio manufacturer	2	The on-board computer should integrate car radio functionality since it will replace the regular radio.
Card reader manufacturer	1	The access device will be purchased from third party.
Legacy systems takeback law	3	What does the law say about the disposal of old devices? Who is responsible?
Lawmaker	1	What size/weight is permitted for the on-board computer? Other legal provisions have to be checked yet.

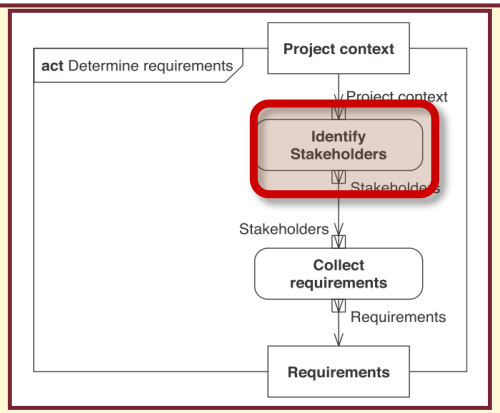




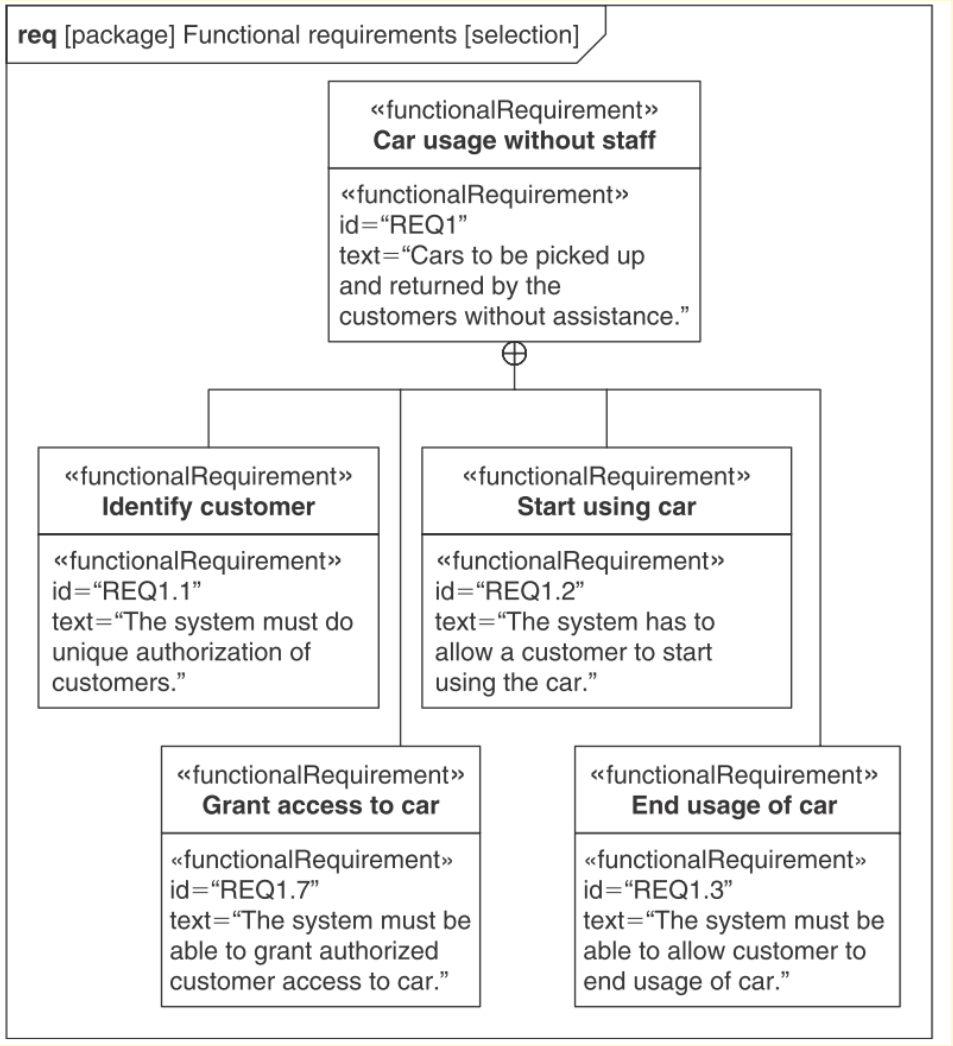
Identify stakeholders

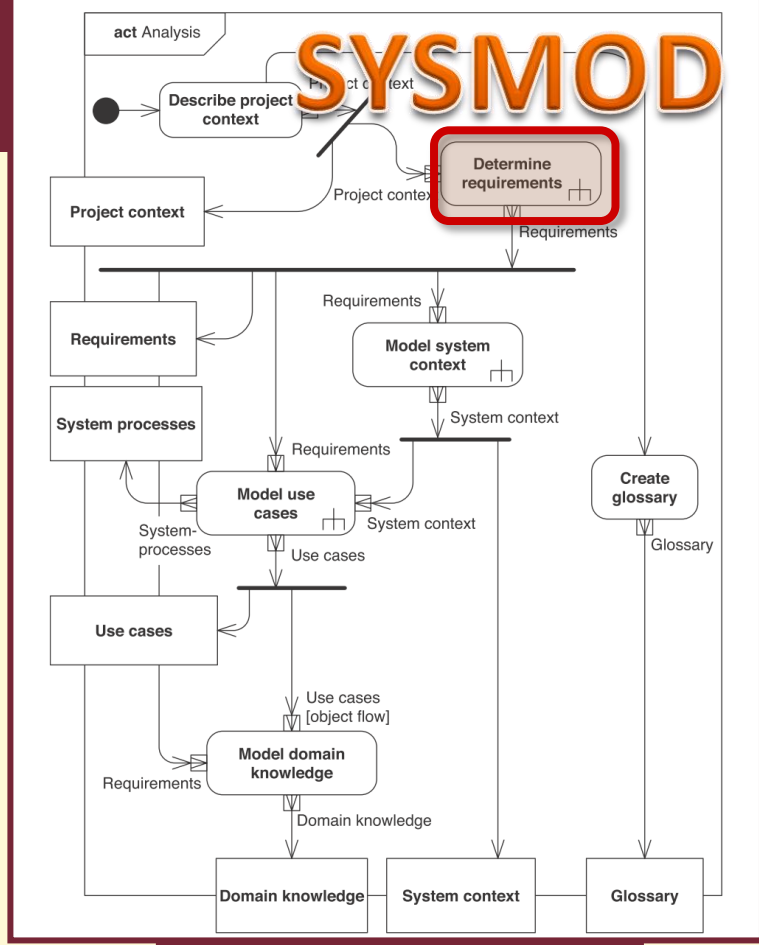


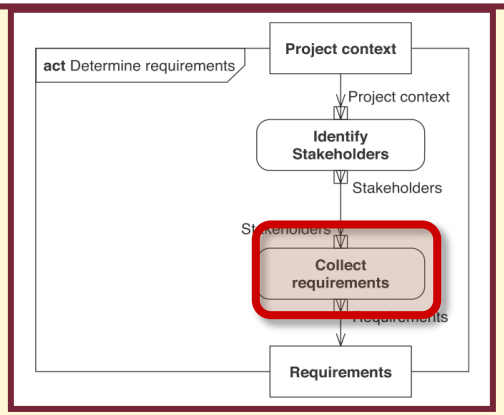




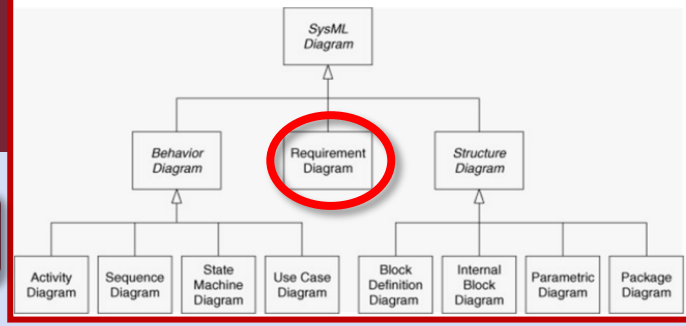
Collect requirements







SysML



Requirements diagram

Name of diagram

Name of context element

Requirement composition

Type of context element

Requirement

req [package] Functional requirements [selection]

«functionalRequirement»
Identify customer

«functionalRequirement»
id="REQ1.1"
text="The system must do unique authorization of customers."

«functionalRequirement»
Start using car

«functionalRequirement»
id="REQ1.2"
text="The system has to allow a customer to start using the car."

«functionalRequirement»
Grant access to car

«functionalRequirement»
id="REQ1.7"
text="The system must be able to grant authorized customer access to car."

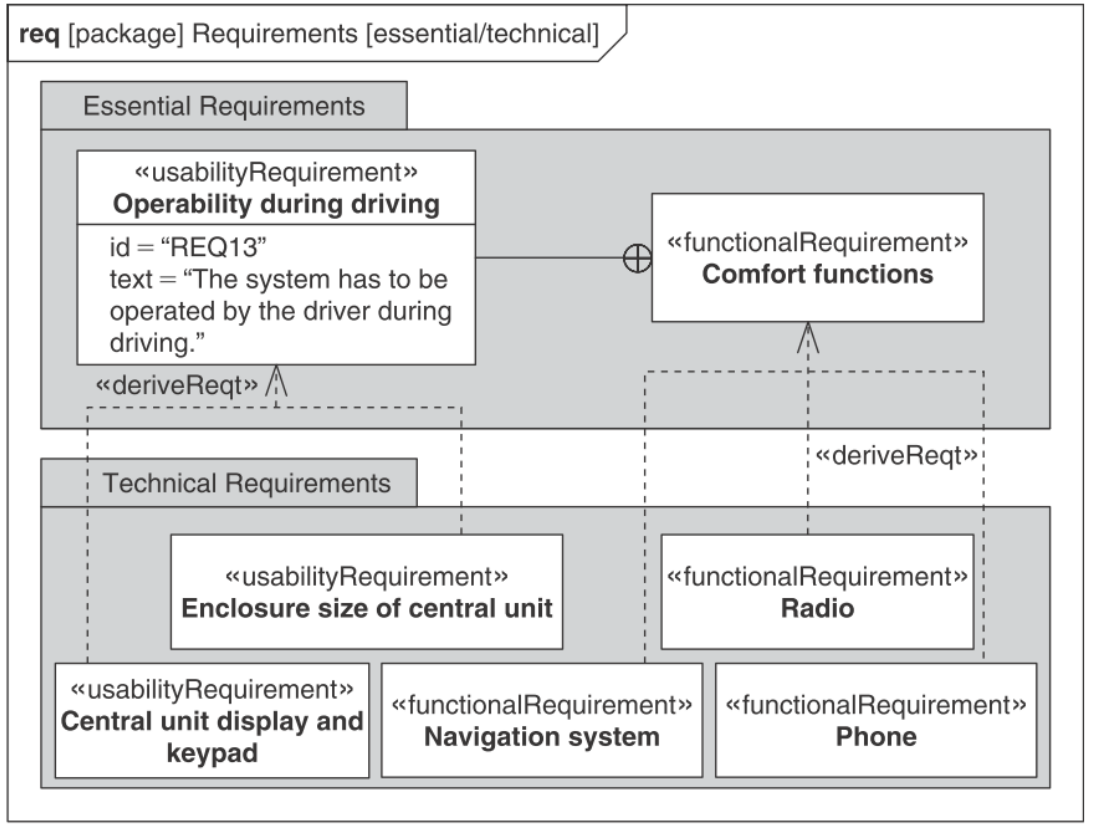
«functionalRequirement»
End usage of car

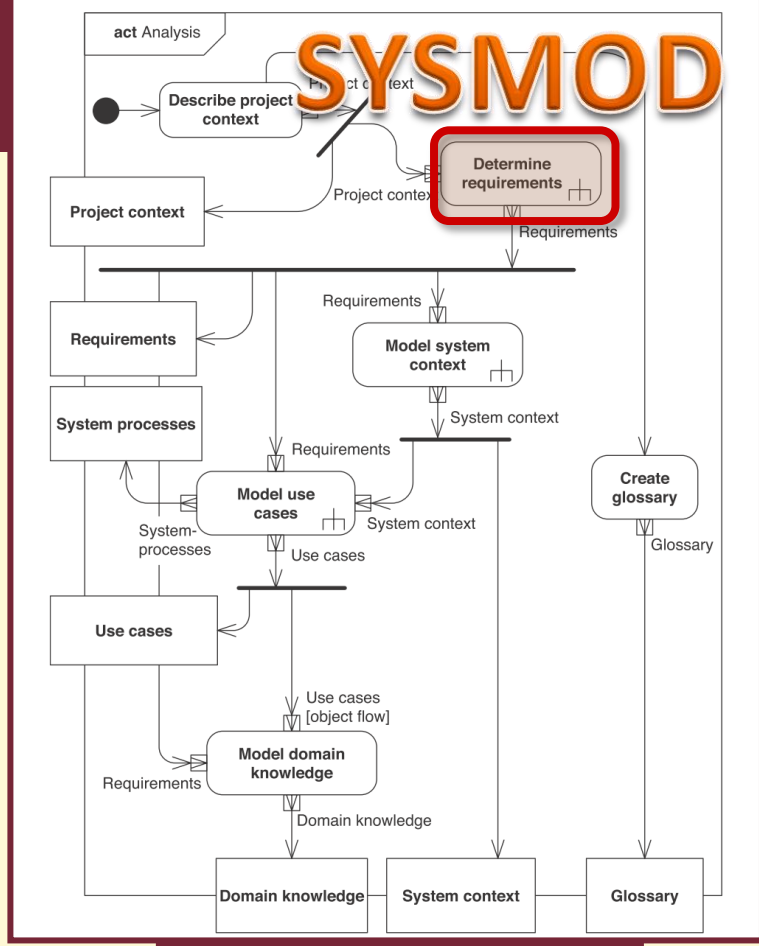
«functionalRequirement»
id="REQ1.3"
text="The system must be able to allow customer to end usage of car."

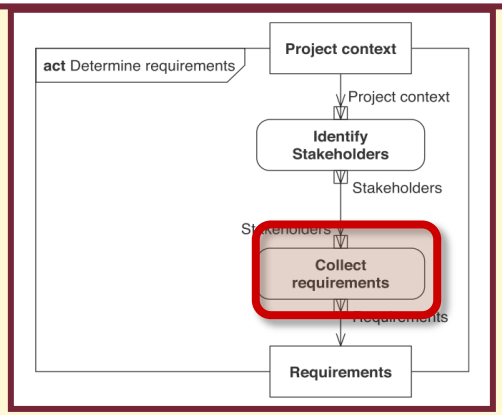
Requirements diagram

Represents text-based requirements and their relationship with other requirements, design elements, and test cases to support requirements traceability

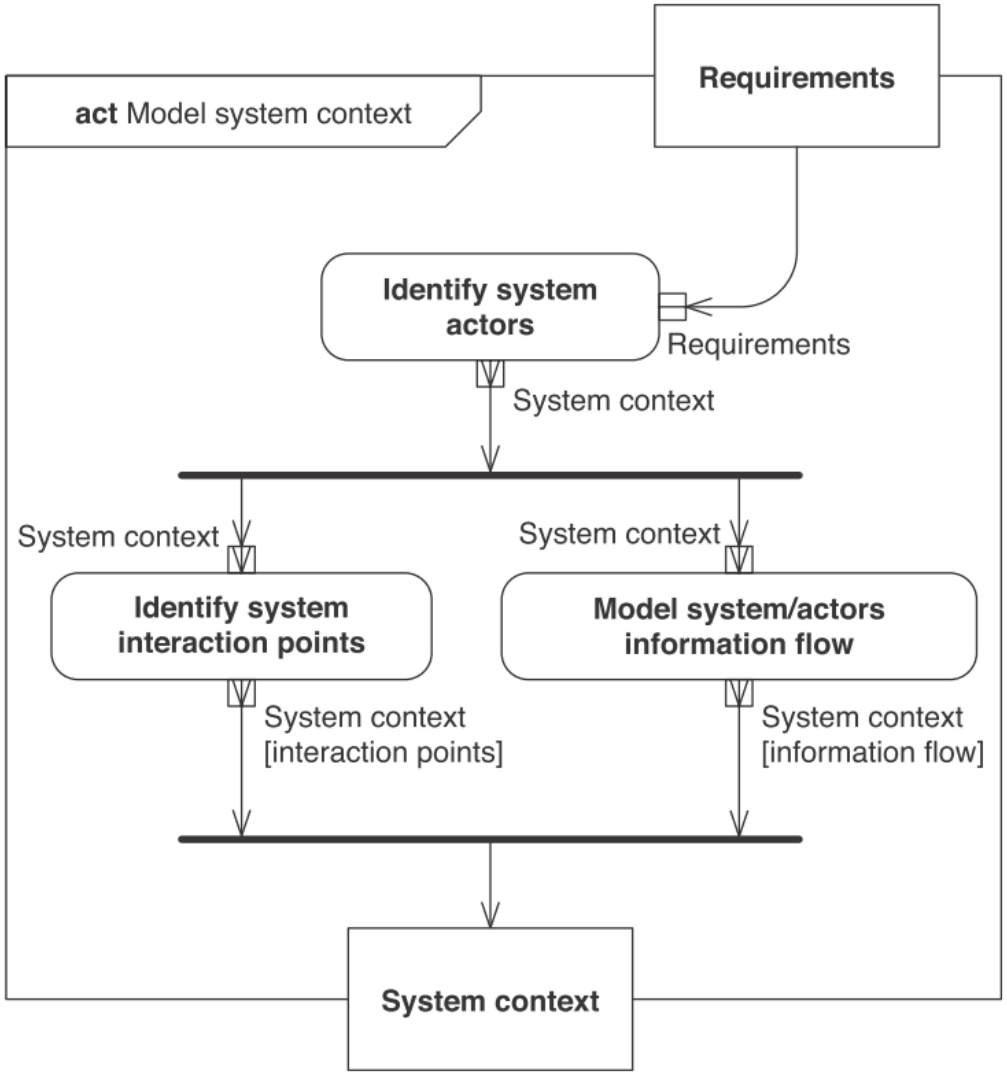
Collect requirements

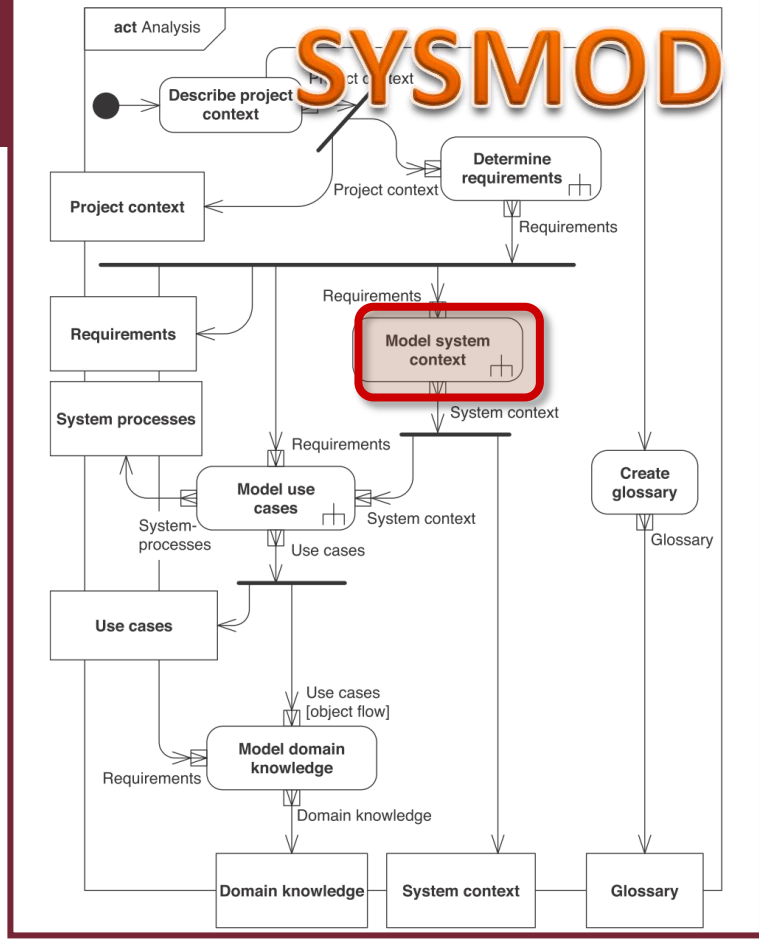




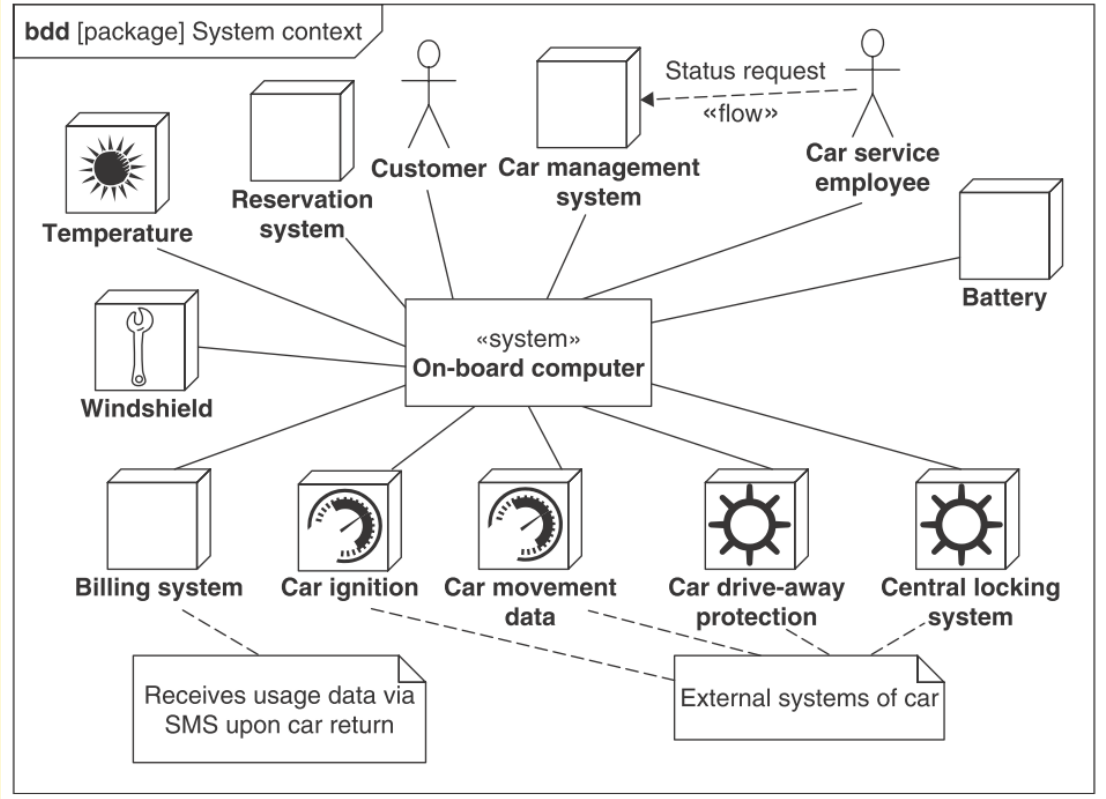


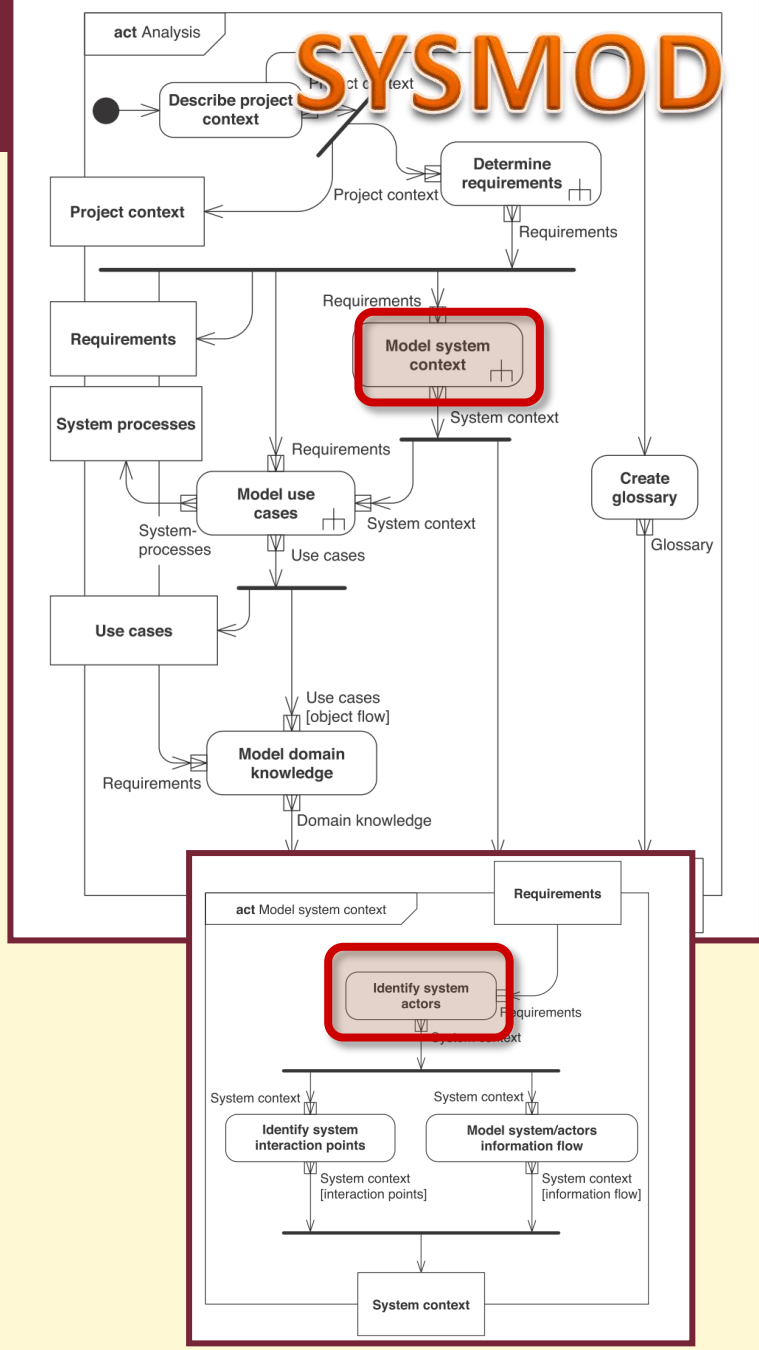
Model System Context



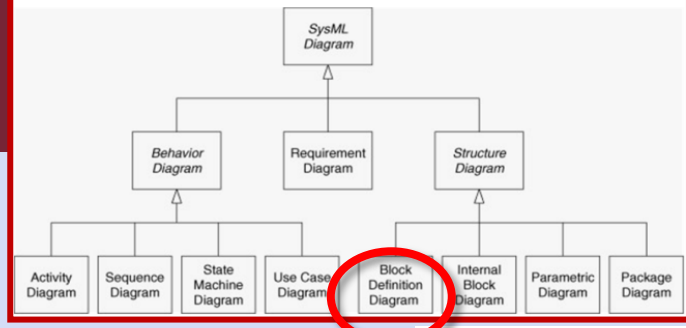


Identify System Actors

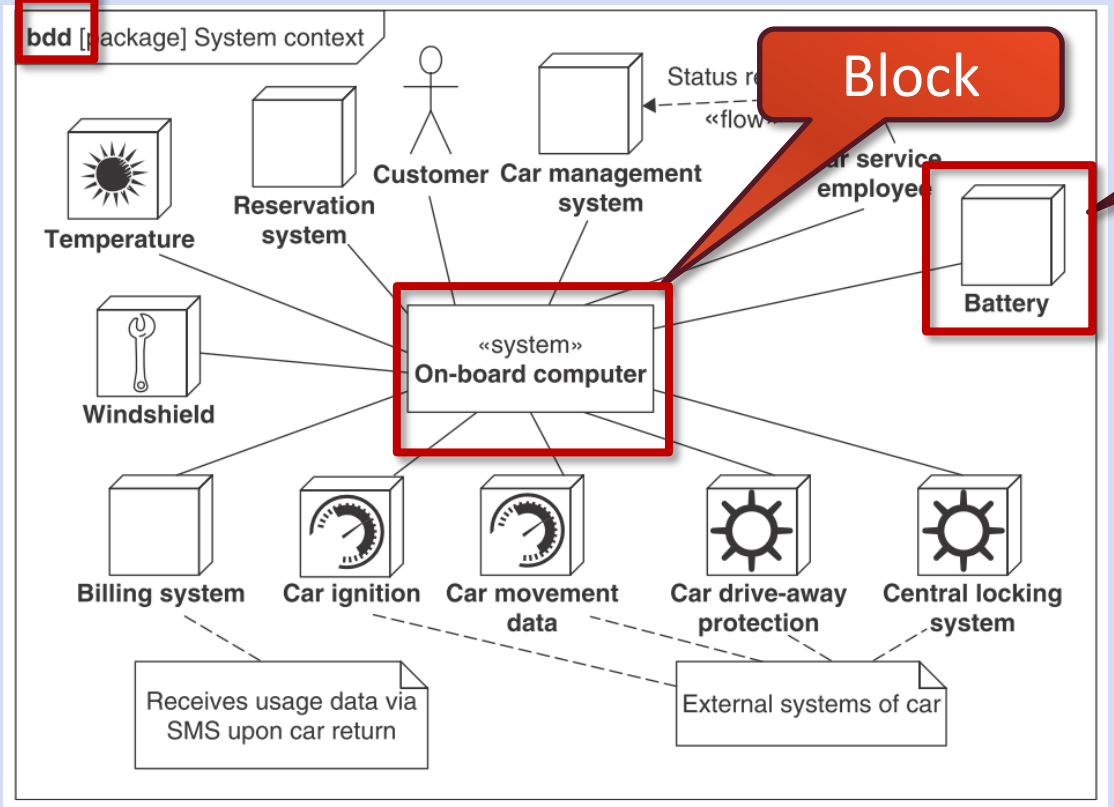




SysML



Block definition diagram



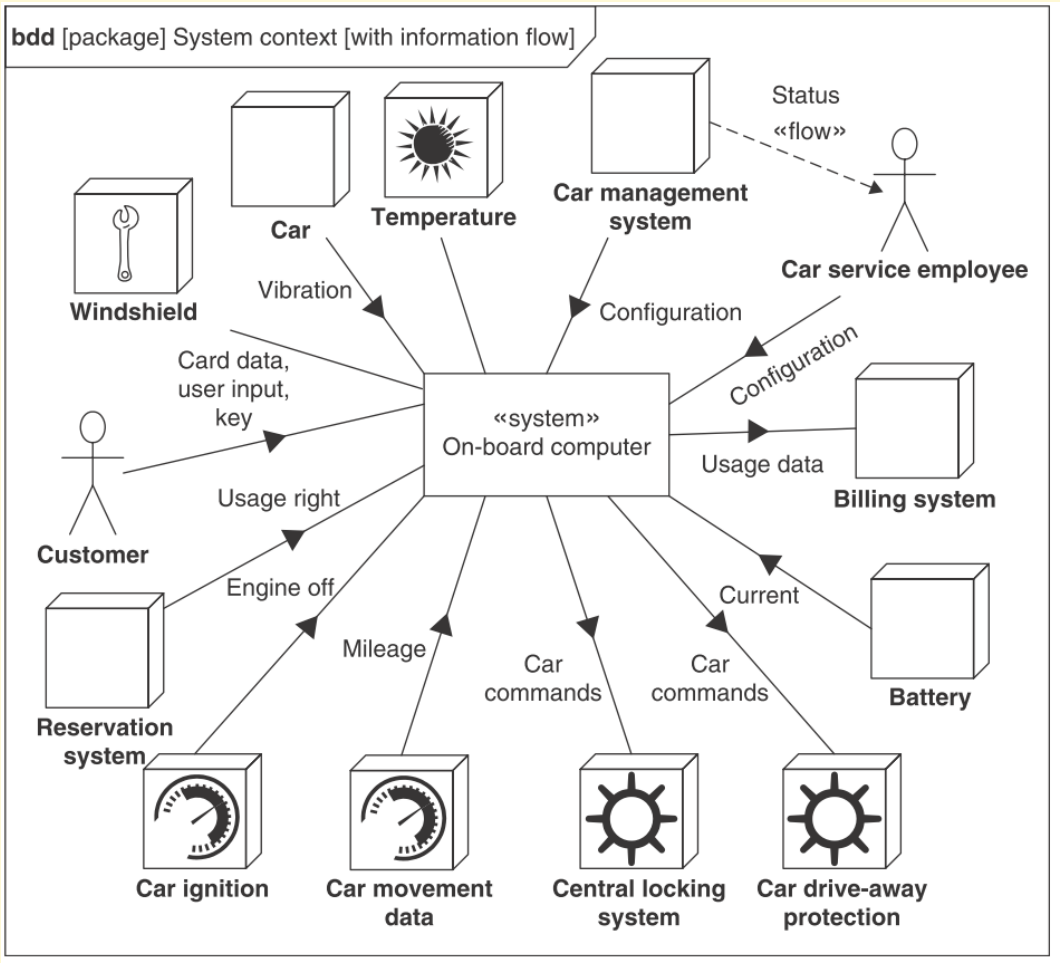
Block

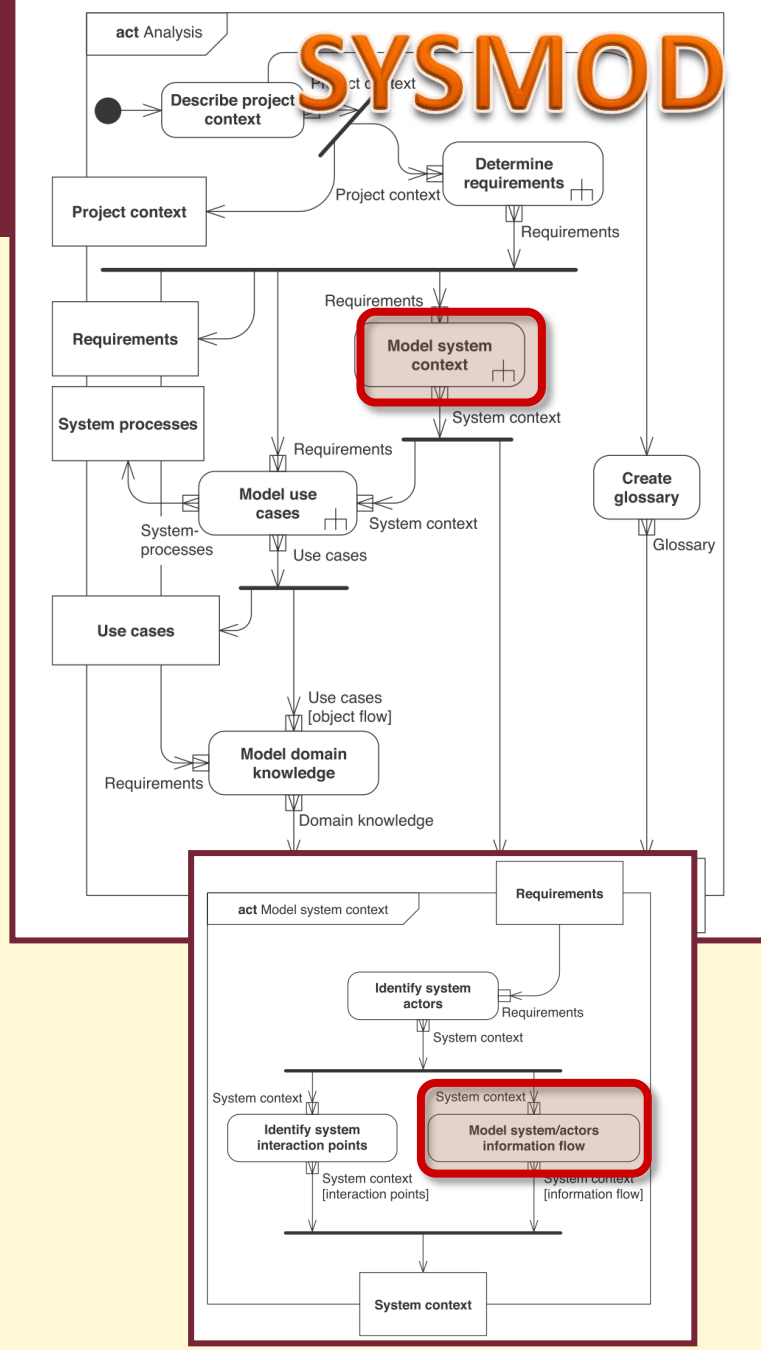
Actor

Block Definition Diagram

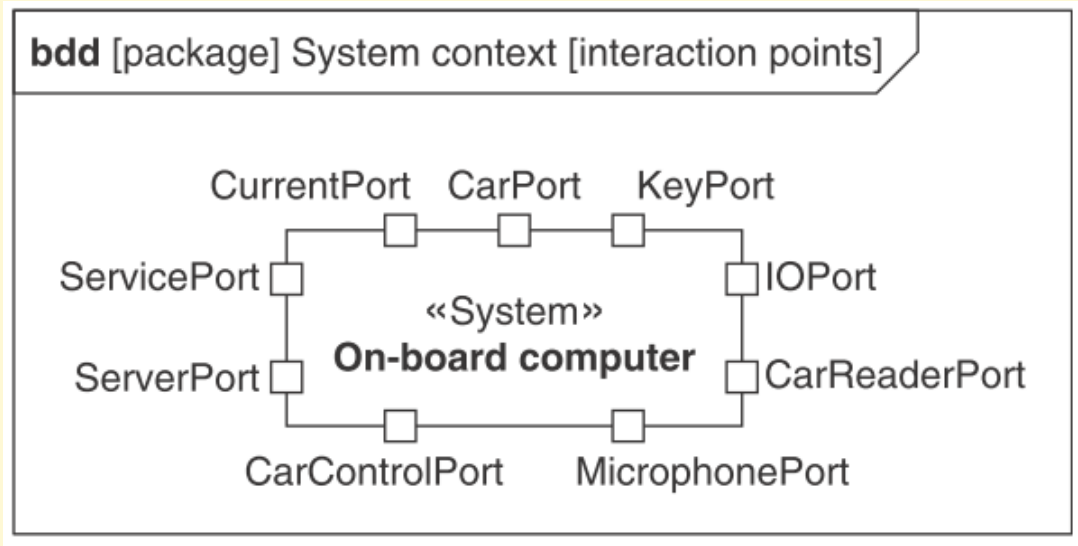
Represents structural elements called blocks, and their composition and classification

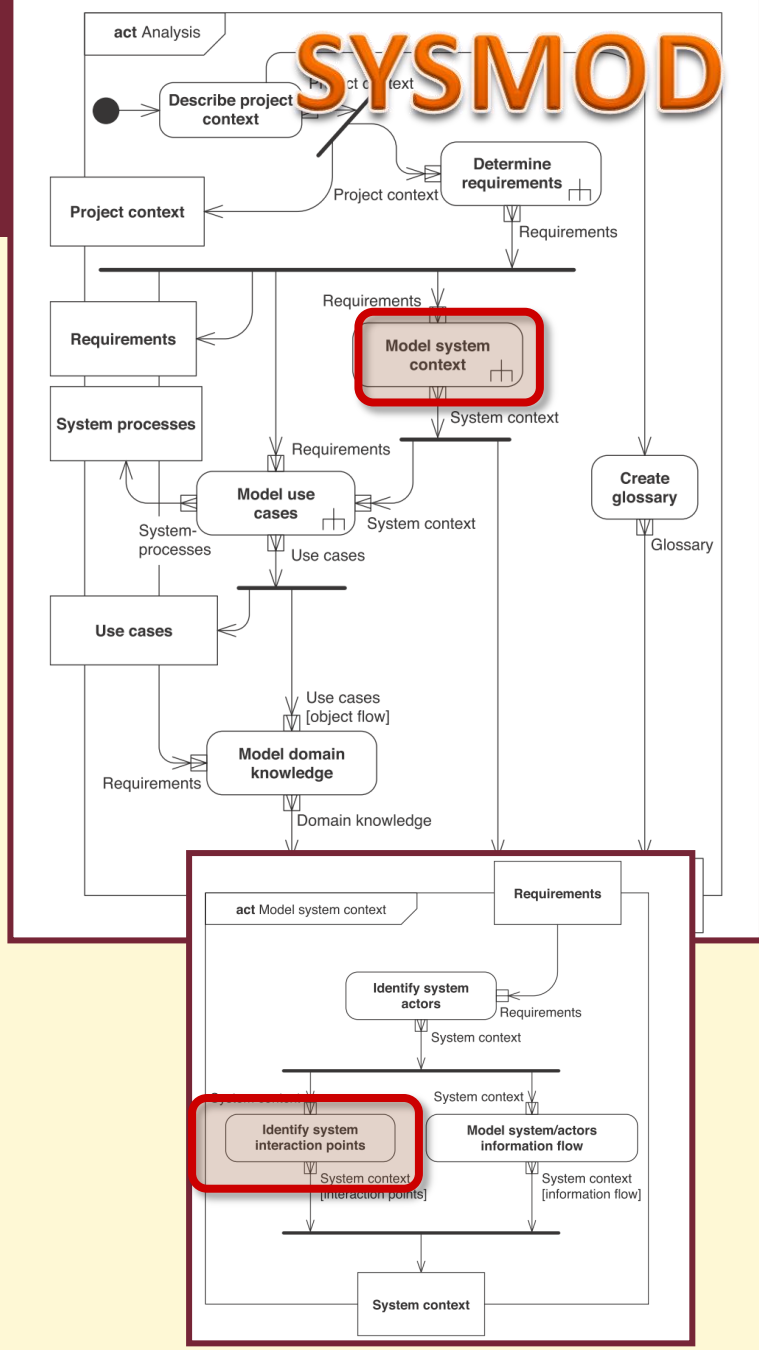
Model System-Actor Information Flow



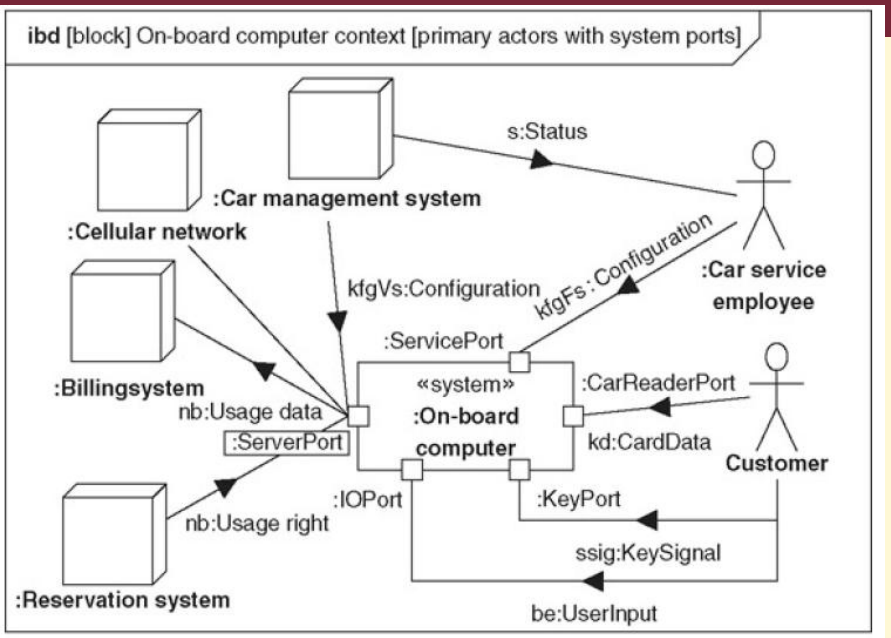


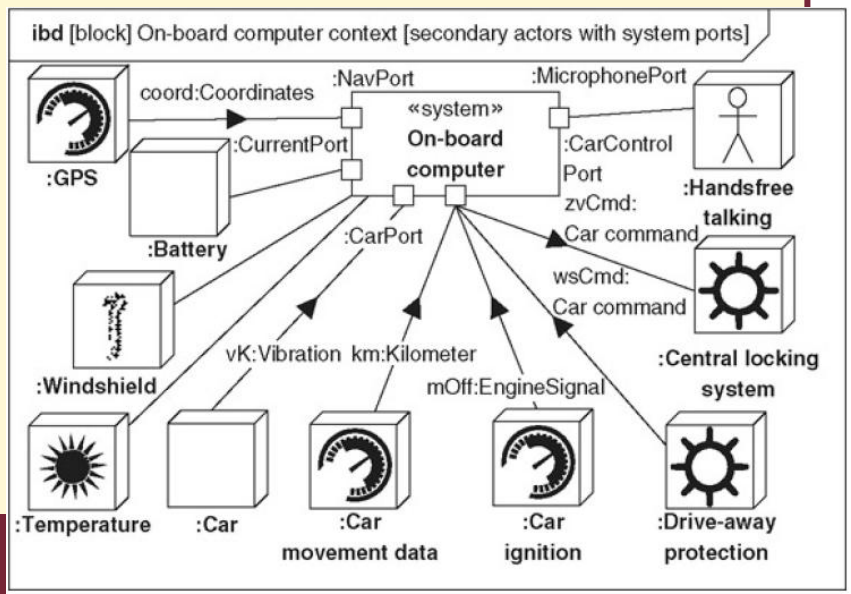
Identify System Interaction Points

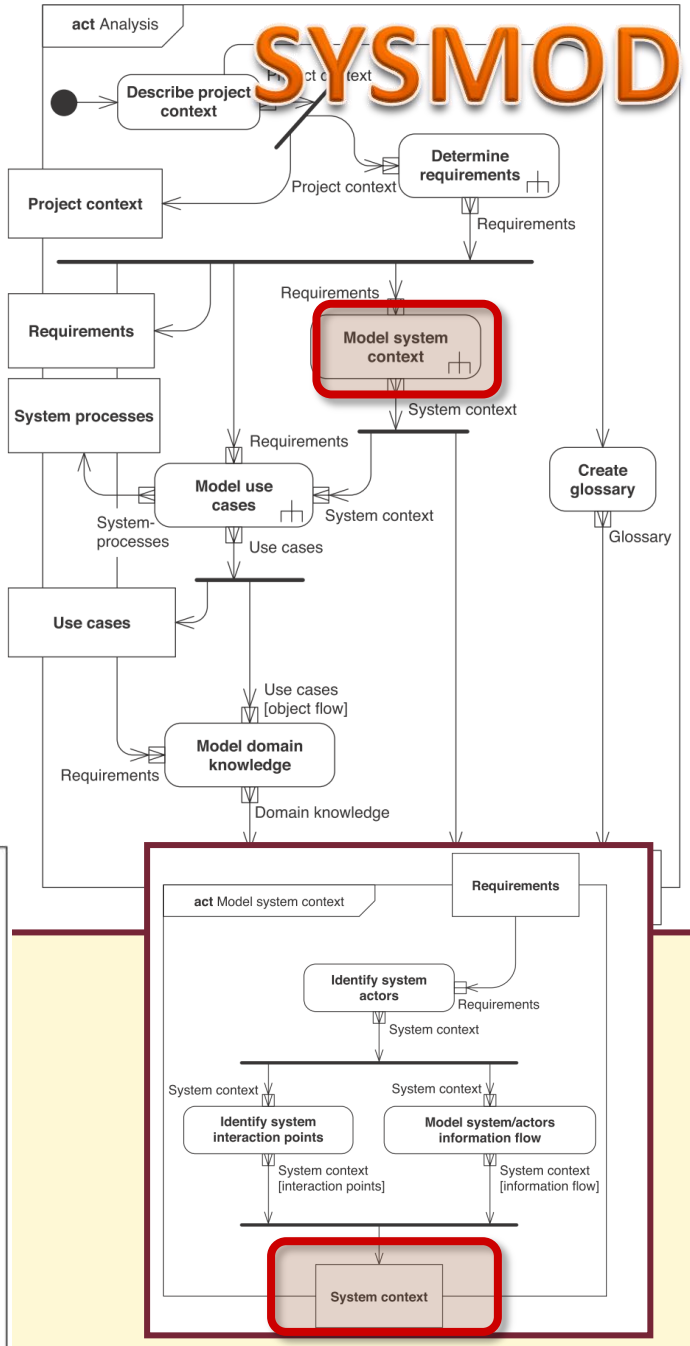




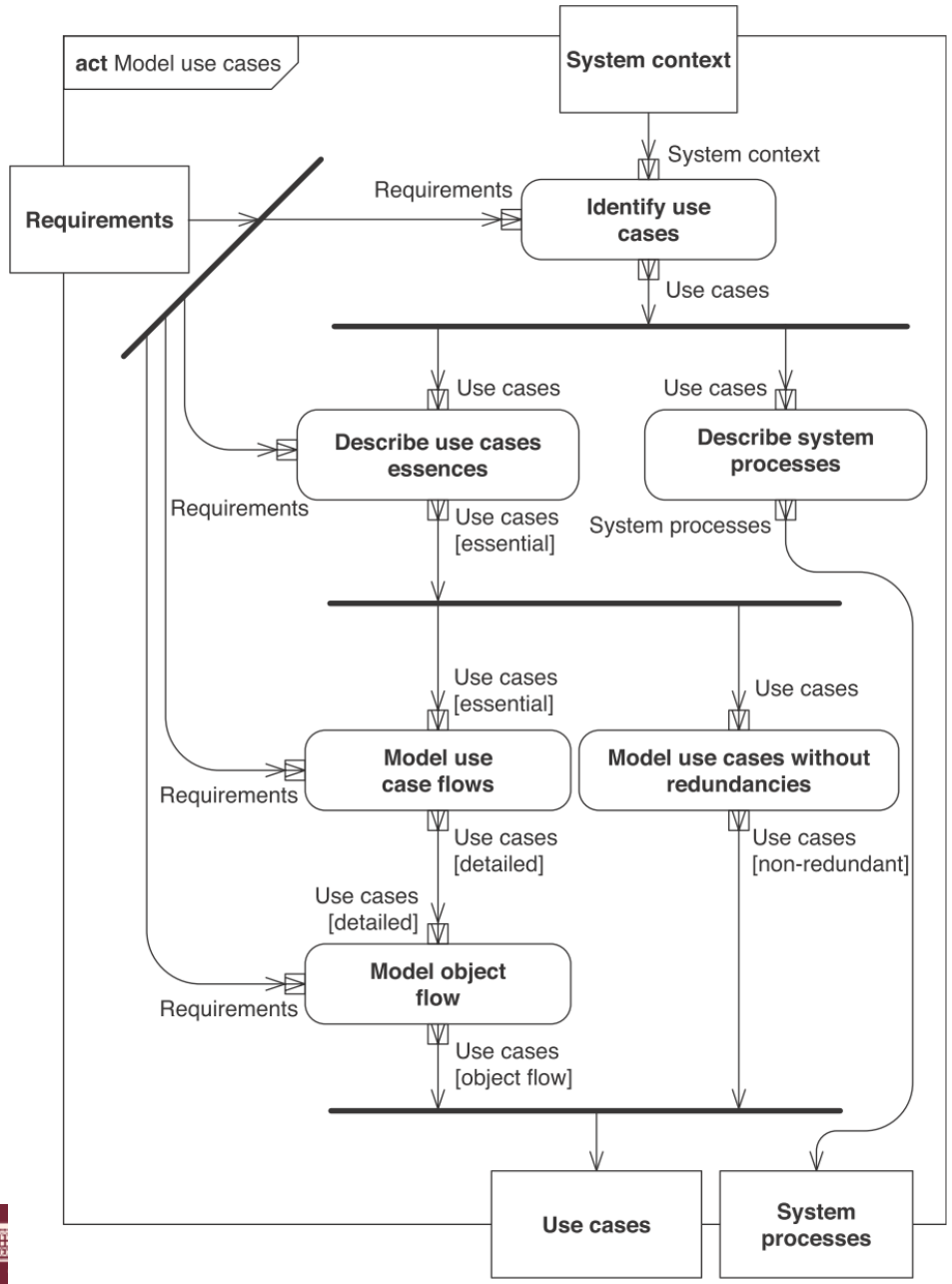
System Context

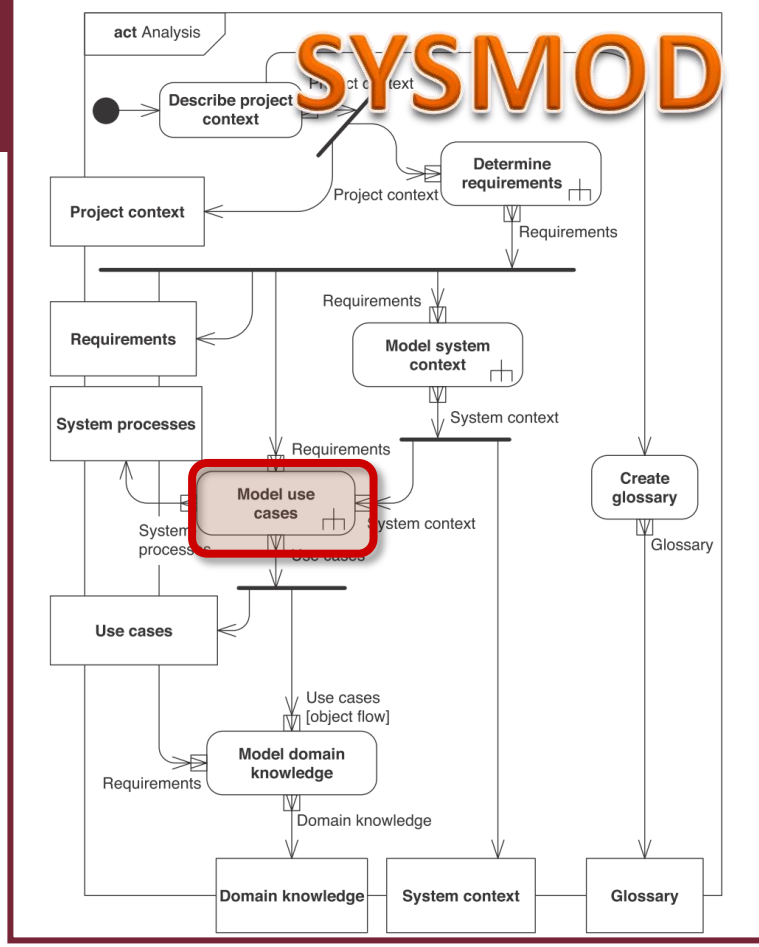




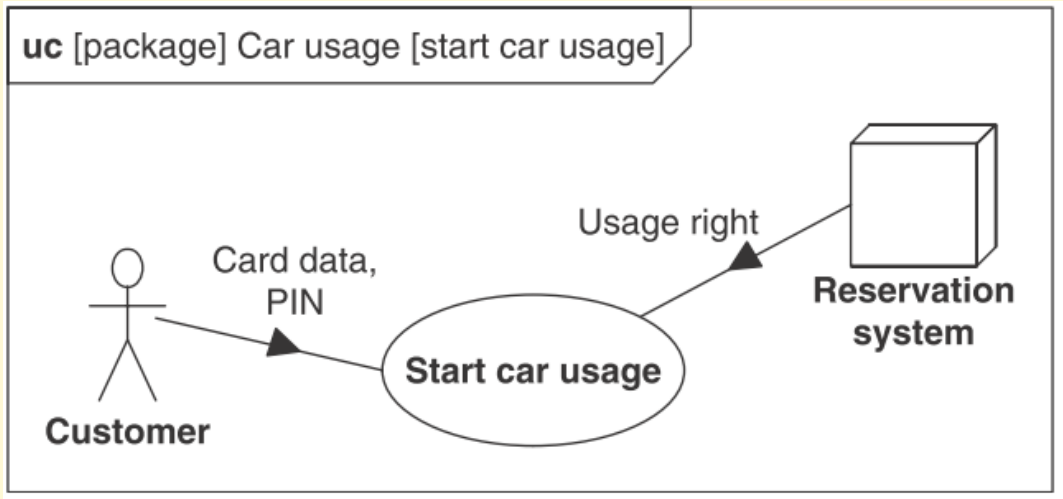


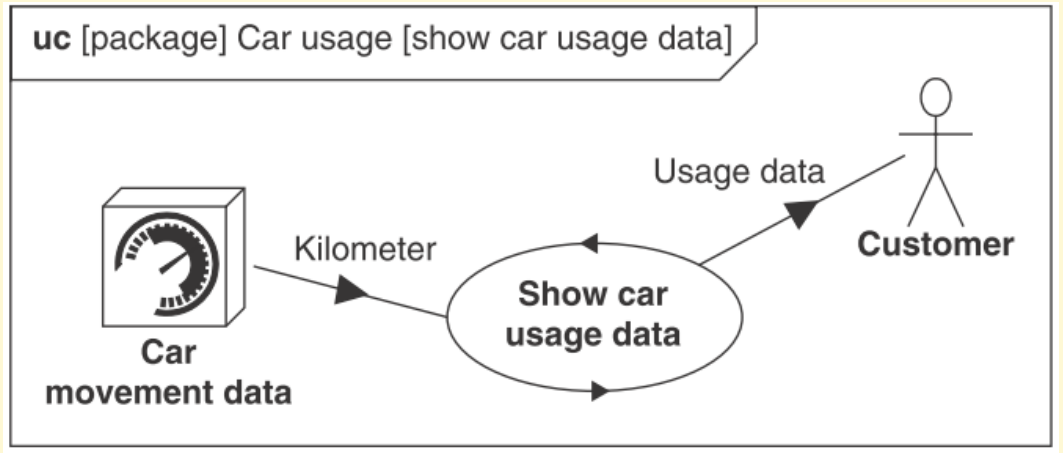
Model Use Cases

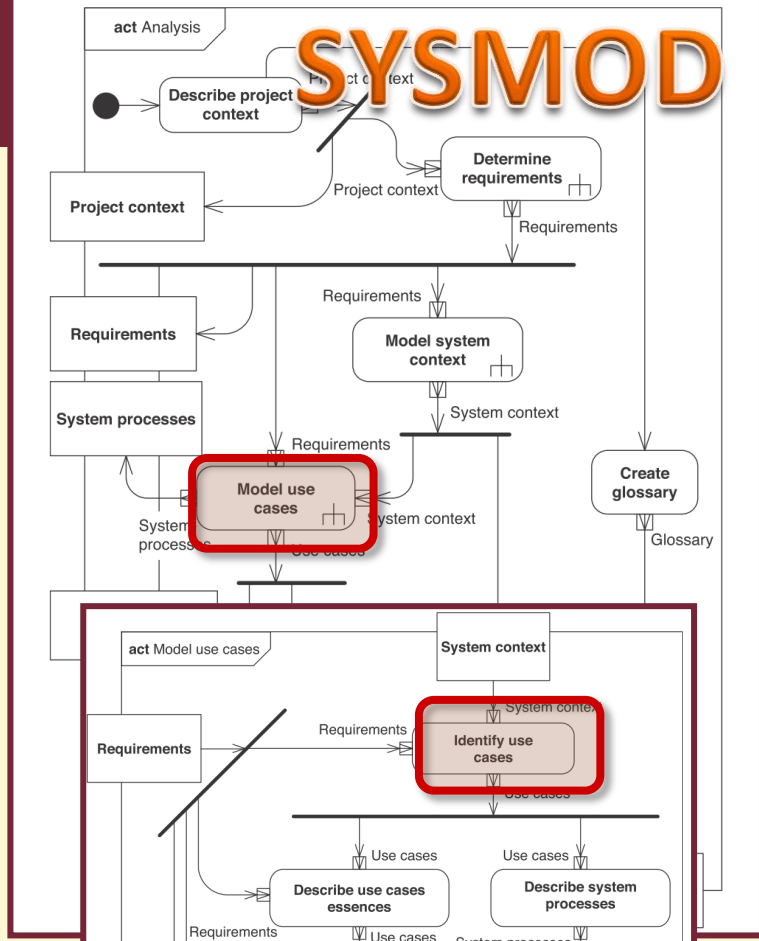


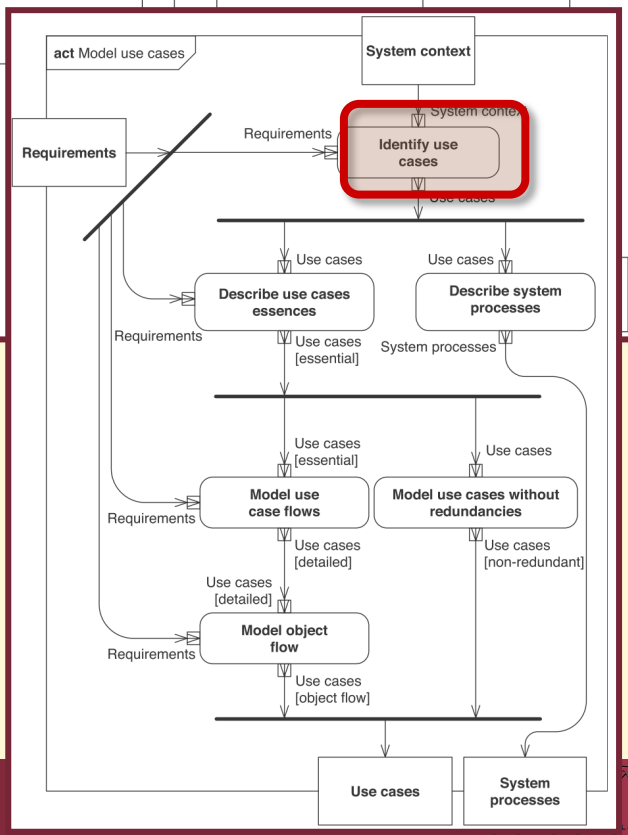


Model Use Cases

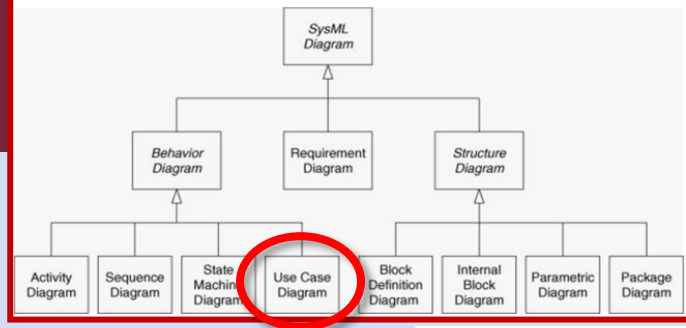


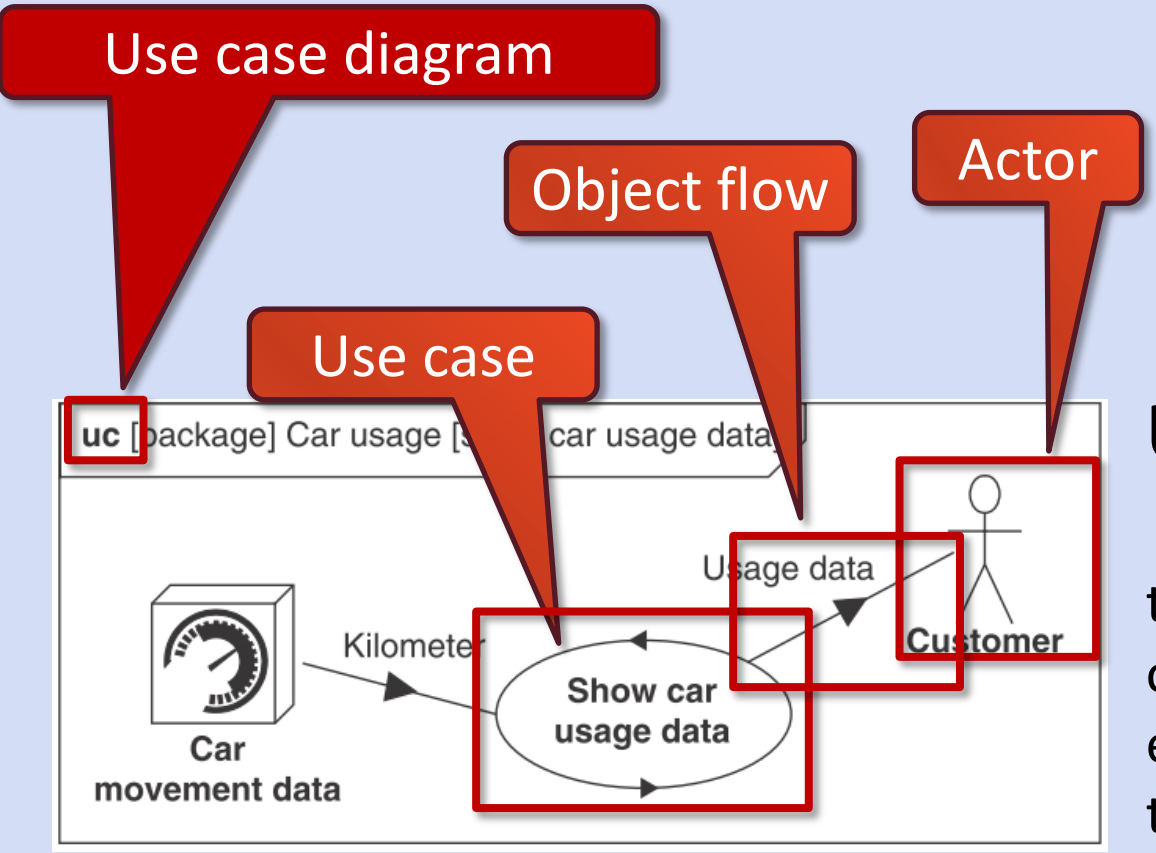






SysML



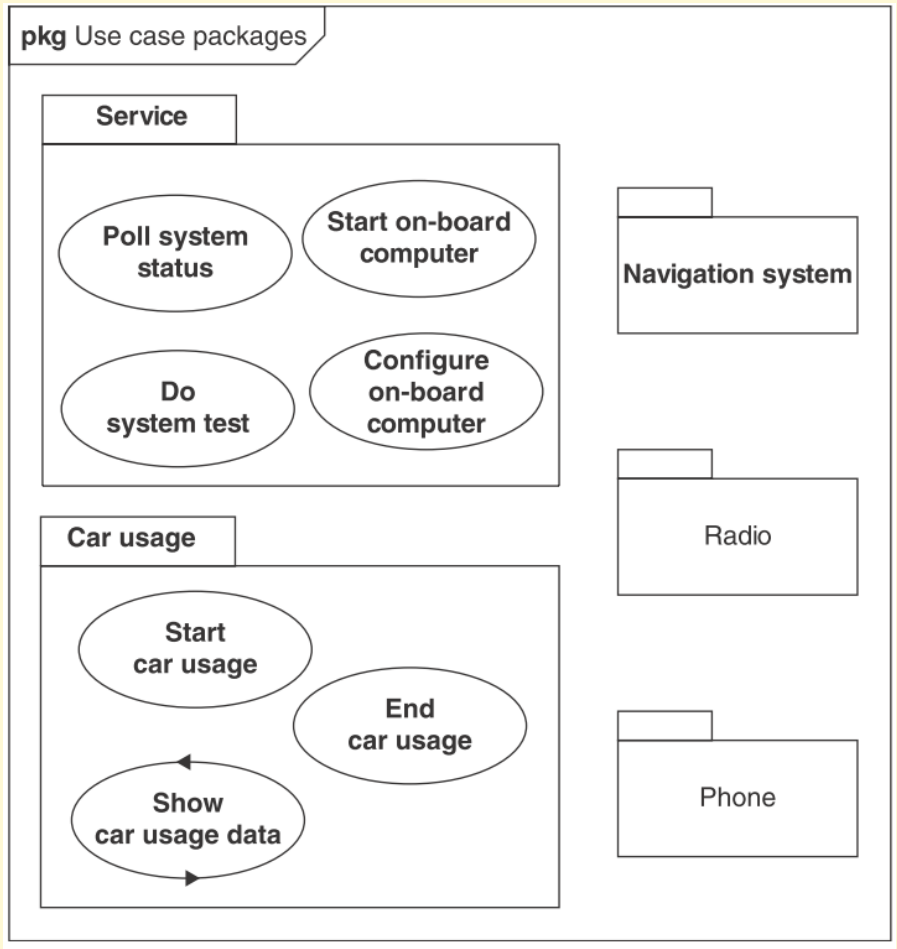


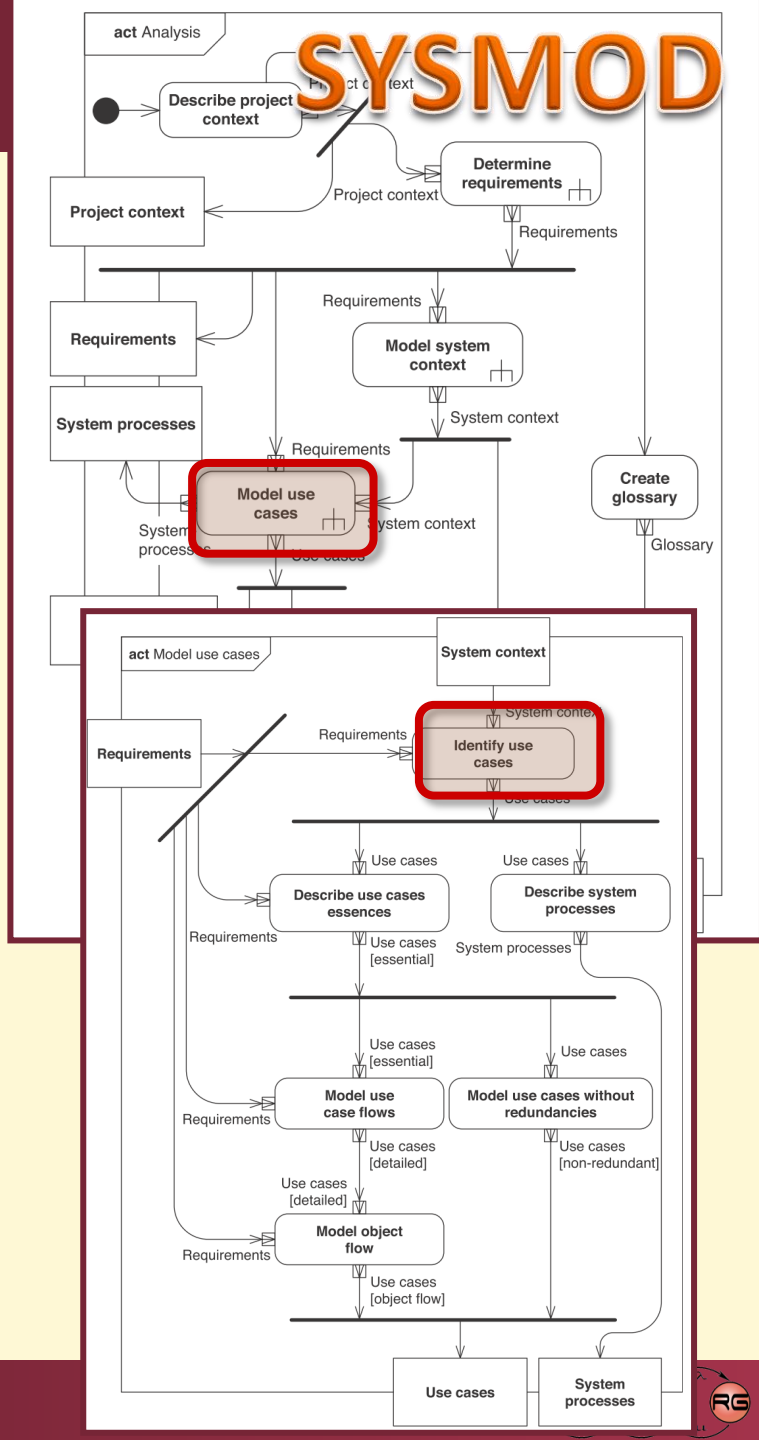
Use Case Diagram

Represents functionality in terms of how a system or other entity is used by external entities (i.e., actors) to accomplish a set of goals

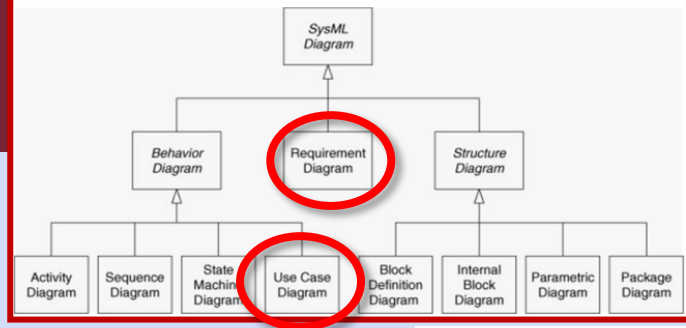
Model Use Cases

Organizing use cases into packages

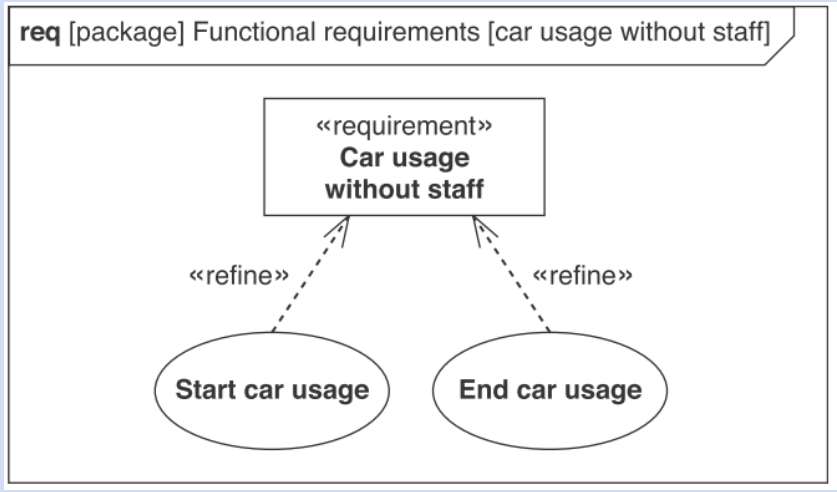


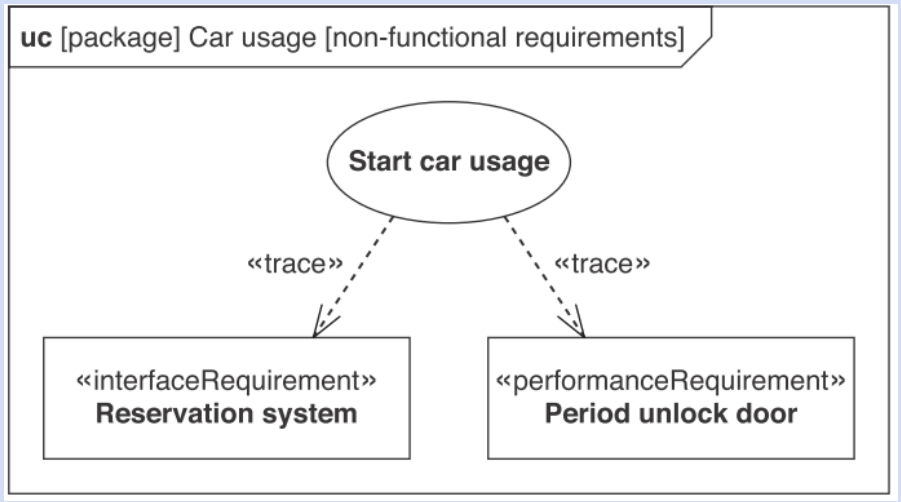


SysML



Requirements traceability

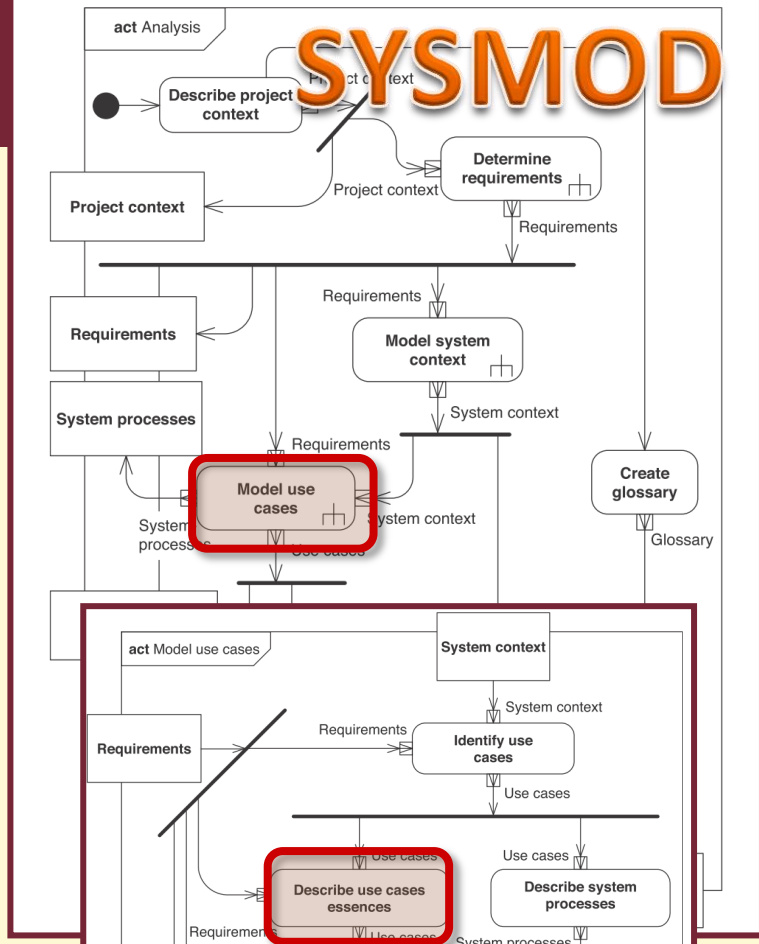


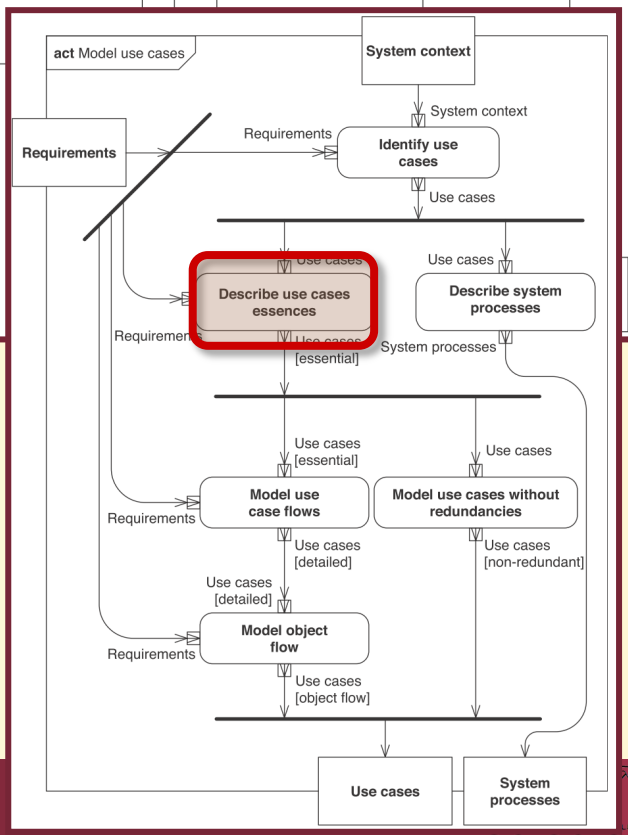


Describe Use Cases Essences

Pragmatic description

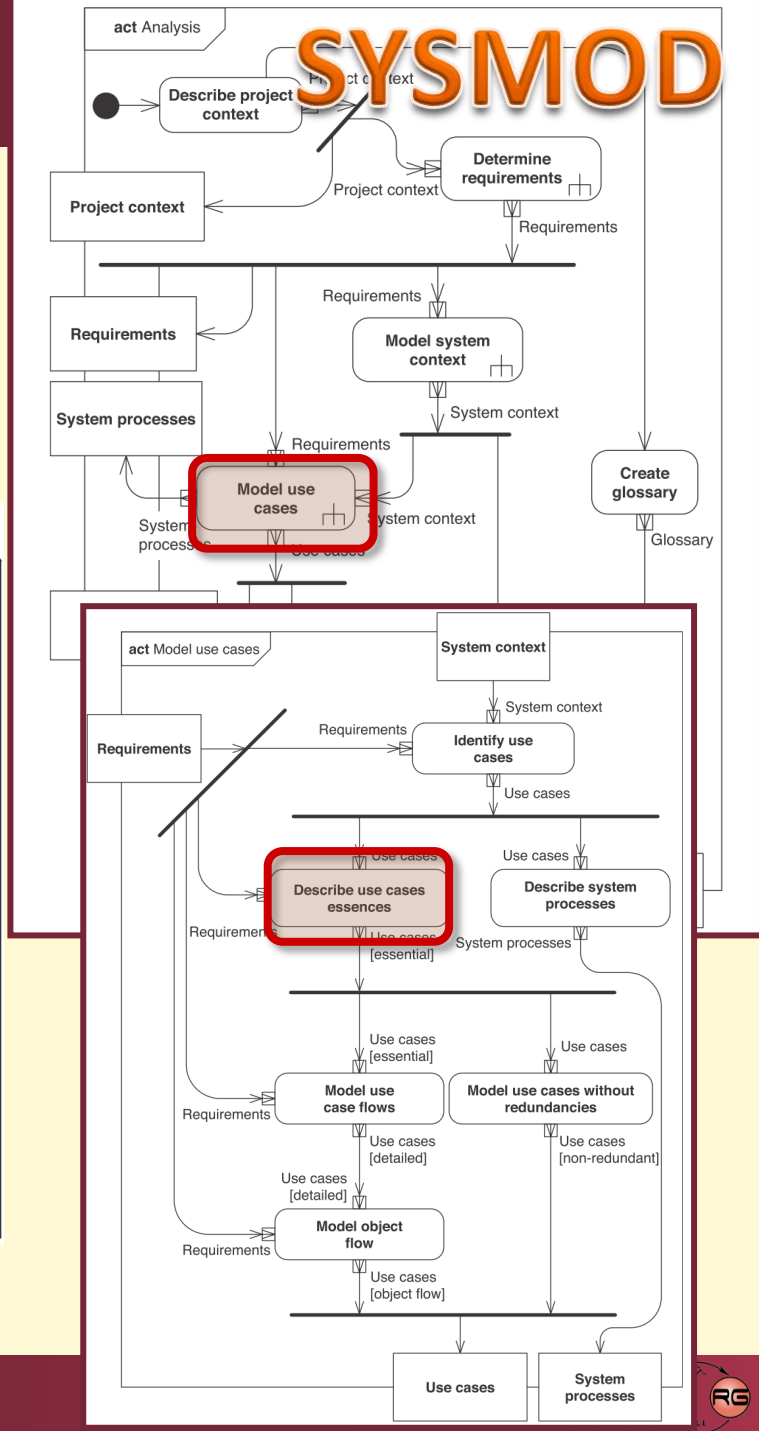
Apply customer card
Read card
Send SMS to reservation center
Receive SMS reply
Check usage right
Unlock car doors via central locking system
Poll PIN
Enter PIN
Verify PIN
Deactivate electronic drive-away protection
Remove key
Display customer welcome message





Describe Use Cases Essences

Pragmatic description	Essential description
Apply customer card Read card	Identify customer
Send SMS to reservation center Receive SMS reply Check usage right	Check usage right
Unlock car doors via central locking system	Unlock car
Poll PIN Enter PIN Verify PIN Deactivate electronic drive-away protection	Check disposal right
Remove key Display customer welcome message	Start car usage Confirm usage start



Describe Use Cases Essences

uc [package] Car usage [essence start car usage]

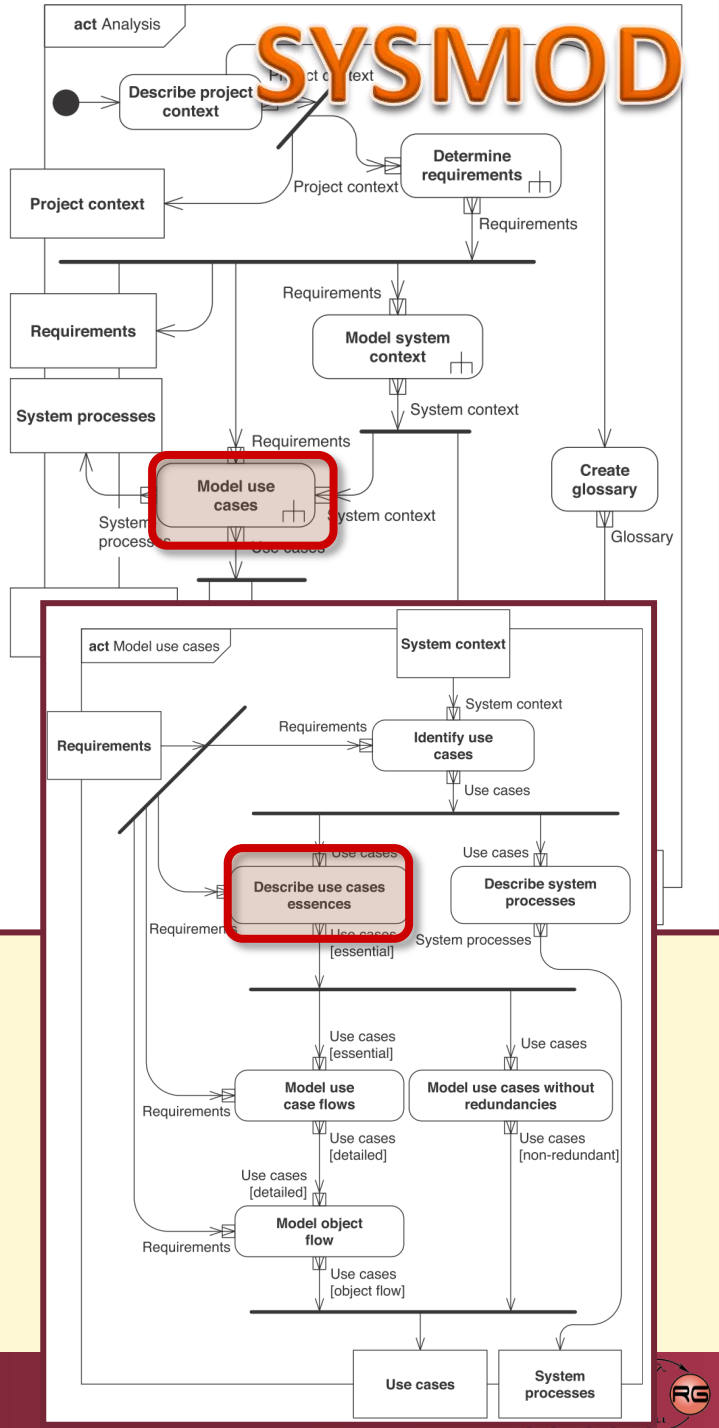
Start car usage

Essence

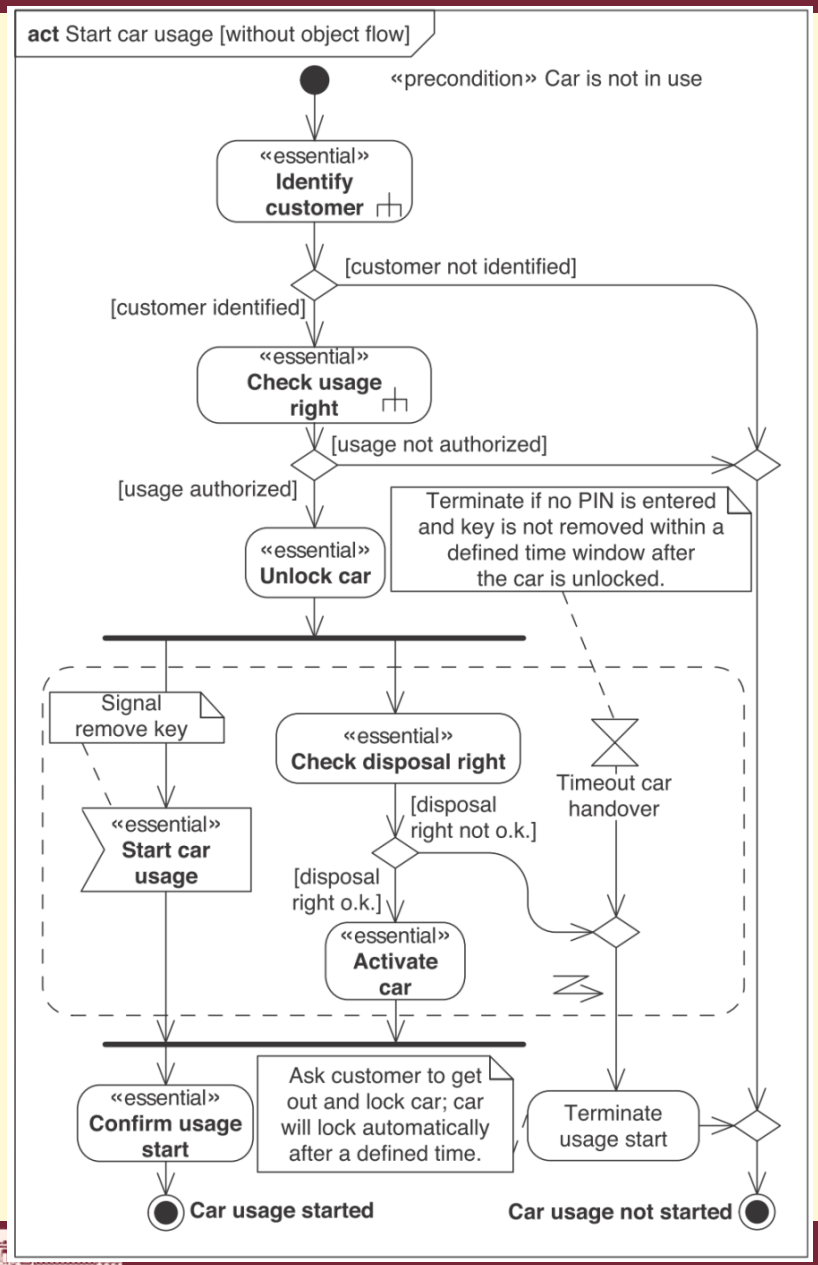
Identify customer
Check usage right
Unlock car
Check disposal right
Activate car
Start car usage
Confirm usage start
Open points

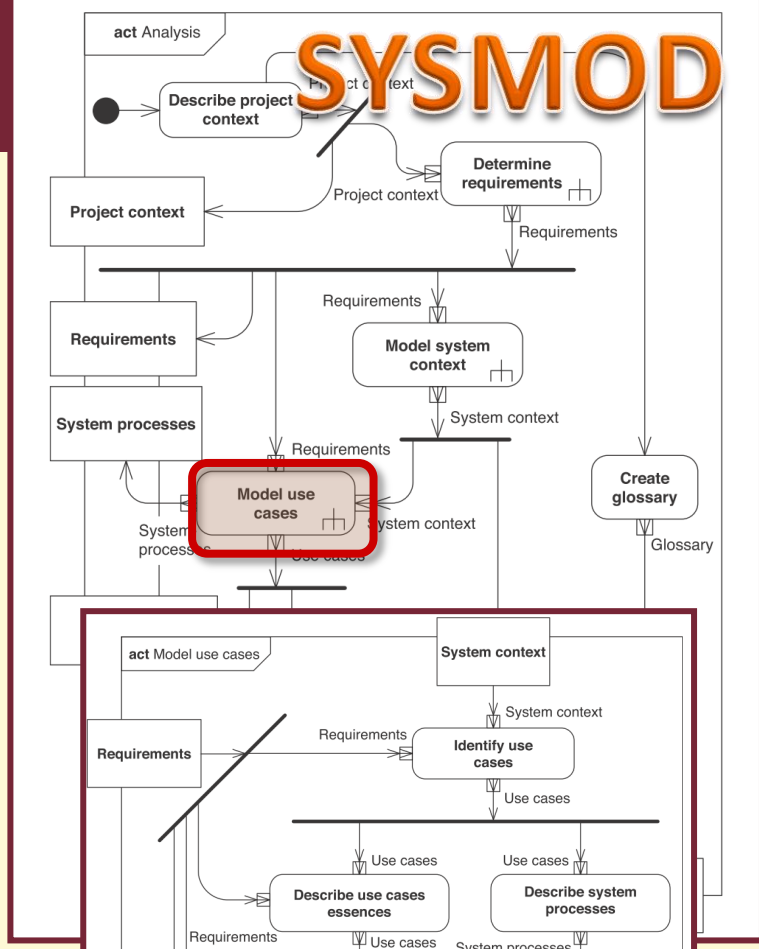
Open points

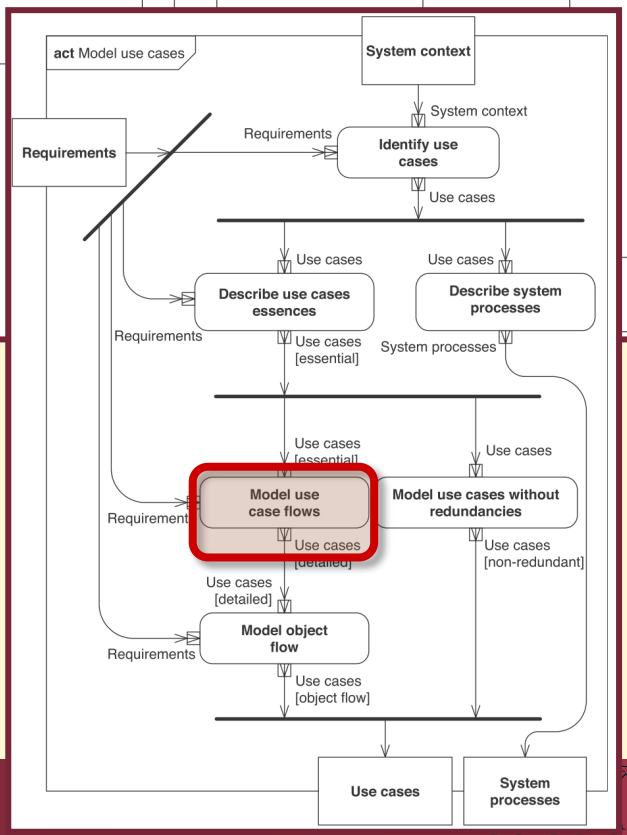
- Planned timeouts are still unclear.
- Does every car have a central locking system and drive-away protection?



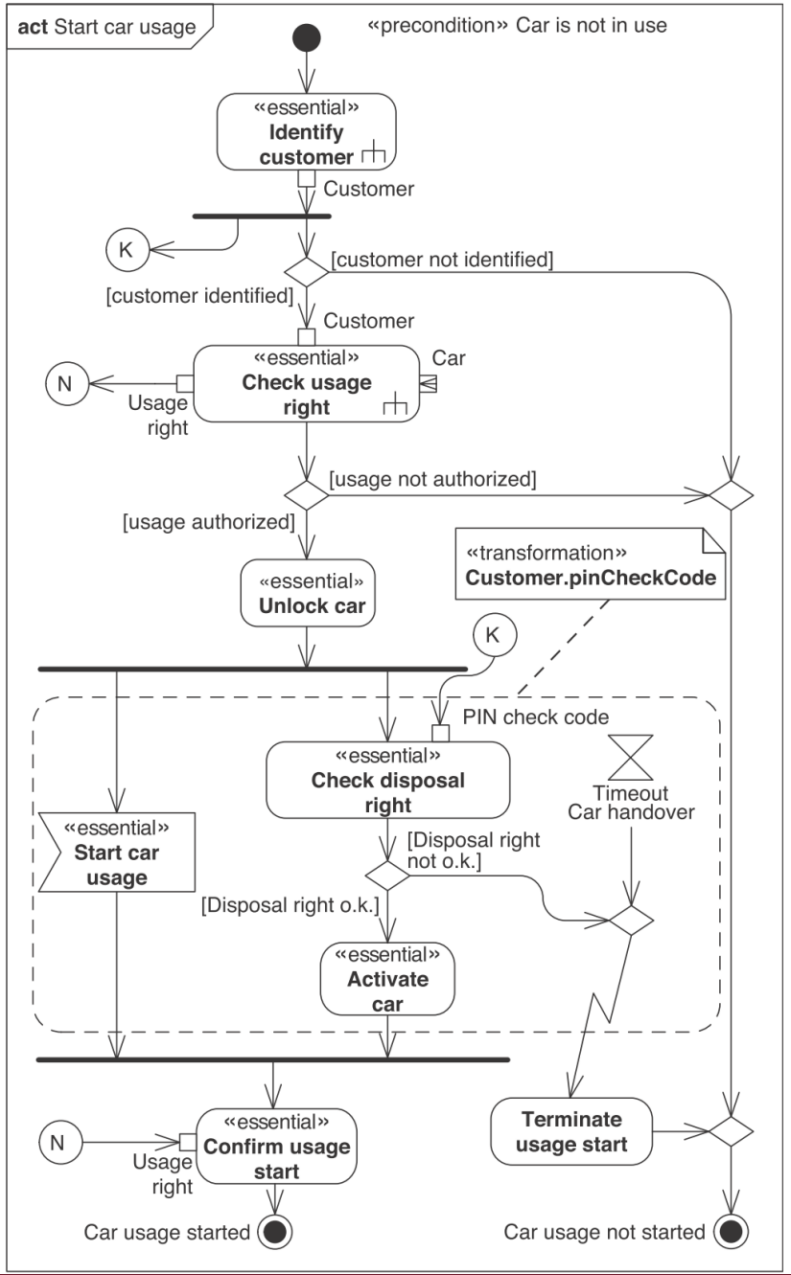
Model Use Case Flows

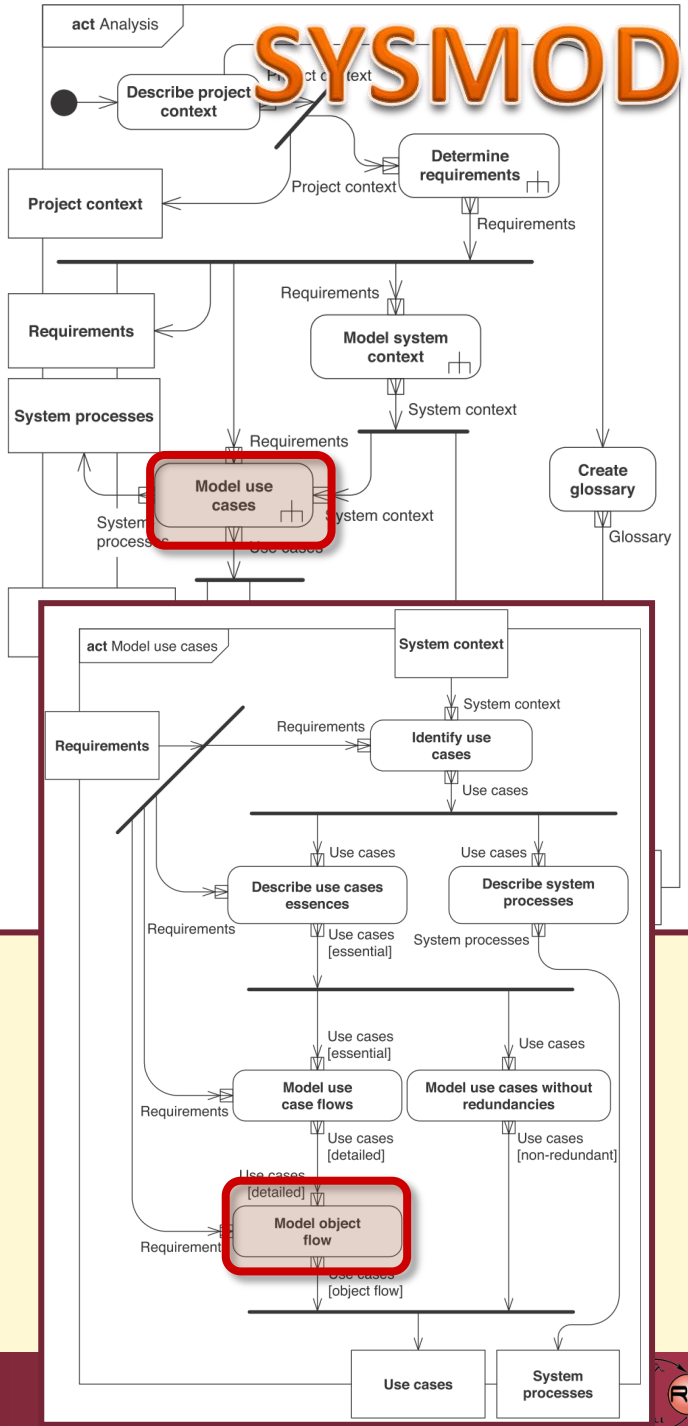




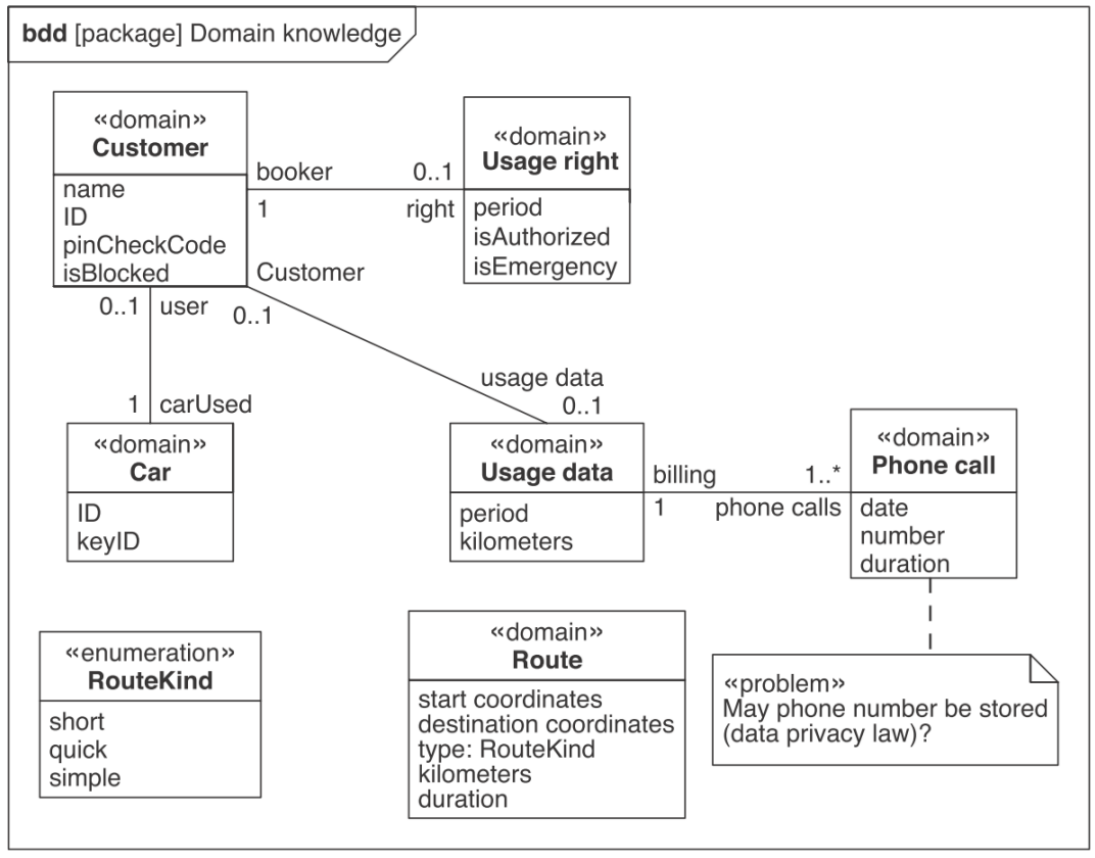


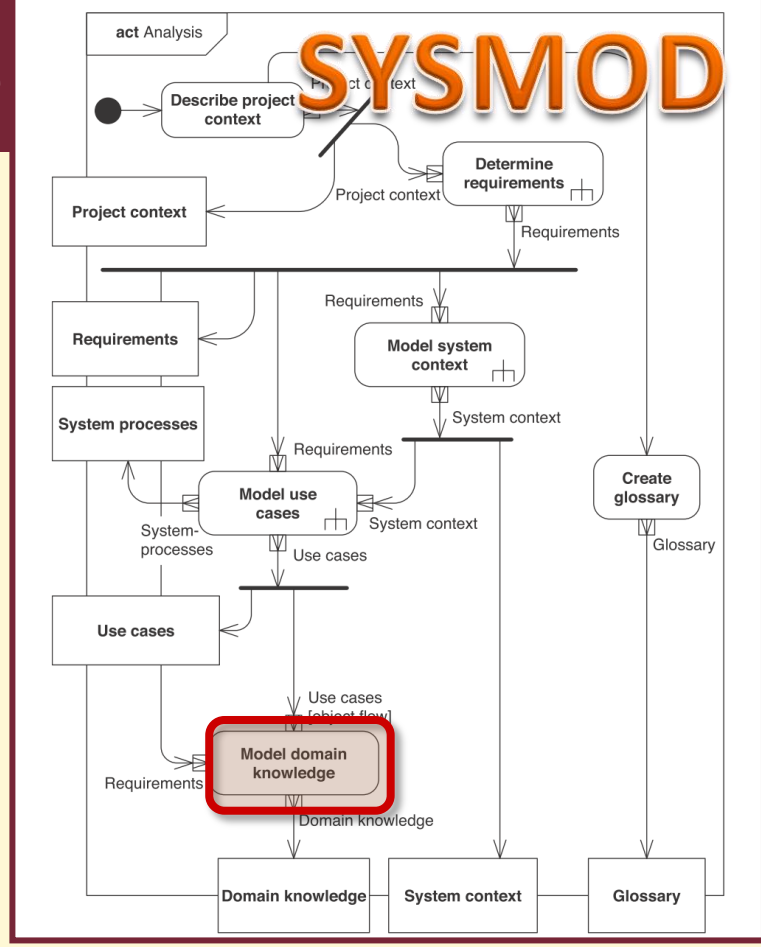
Model Object Flows





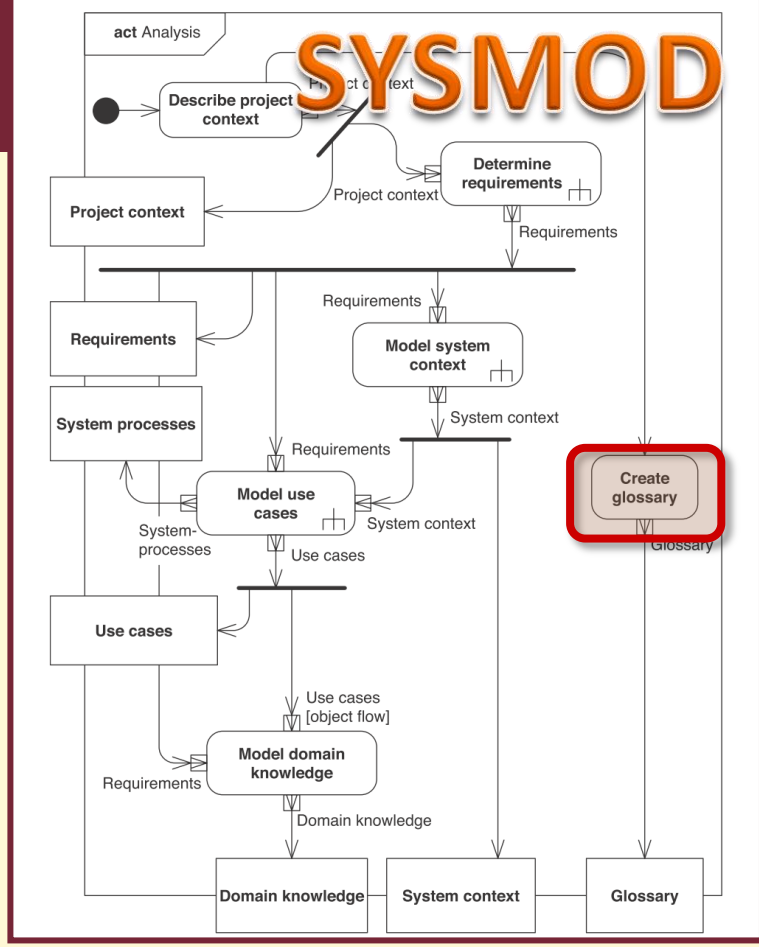
Model Domain Knowledge





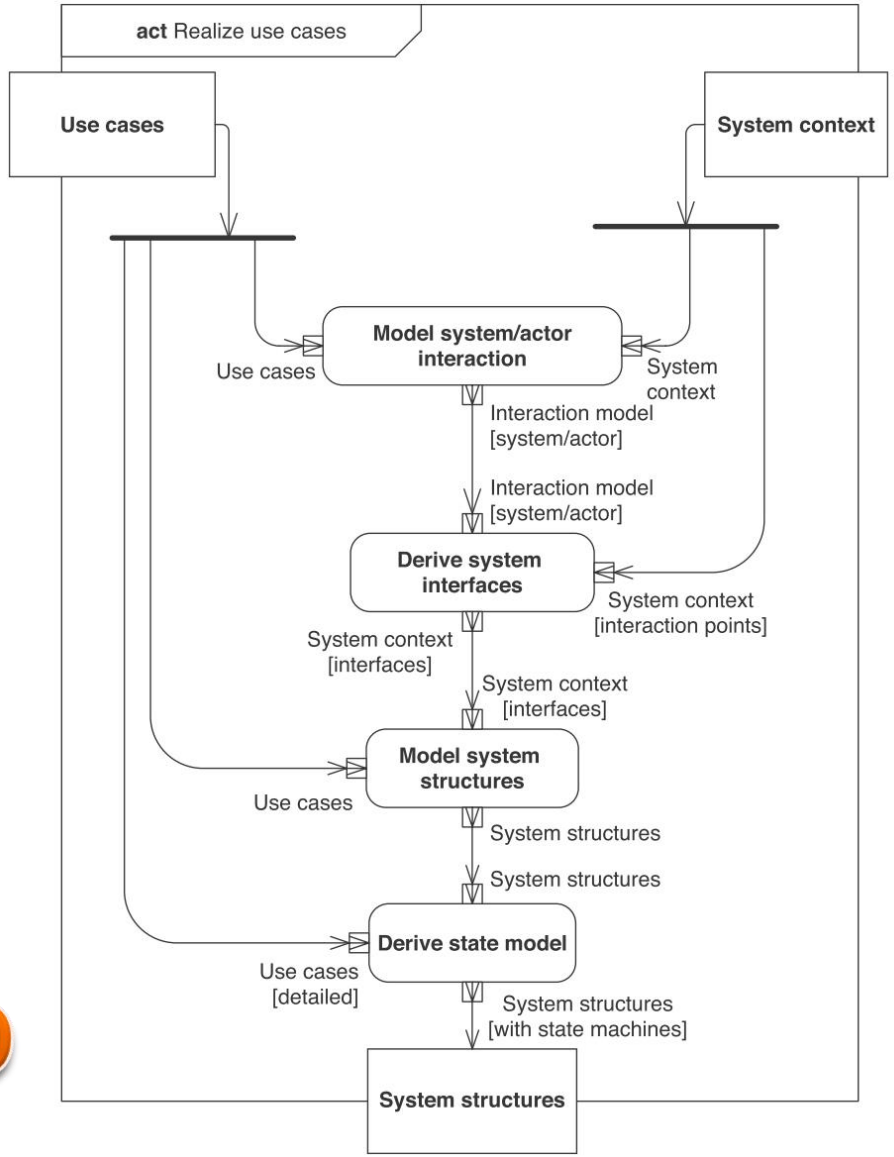
Create Glossary

Table 2.17 Glossary entries.	
Usage right	
Description:	A usage right describes whether or not a customer is entitled to use a car. It includes information about the customer, the booking period, and whether it is an emergency driving case.
Domain block:	Yes
Author, last change:	Tim Weilkiens, April 30, 2004
Disposal right	
Description:	The on-board computer grants a customer disposal right, if this customer has entered a correct customer PIN upon start of usage.
Domain block:	No
Author, last change:	Tim Weilkiens, April 30, 2004



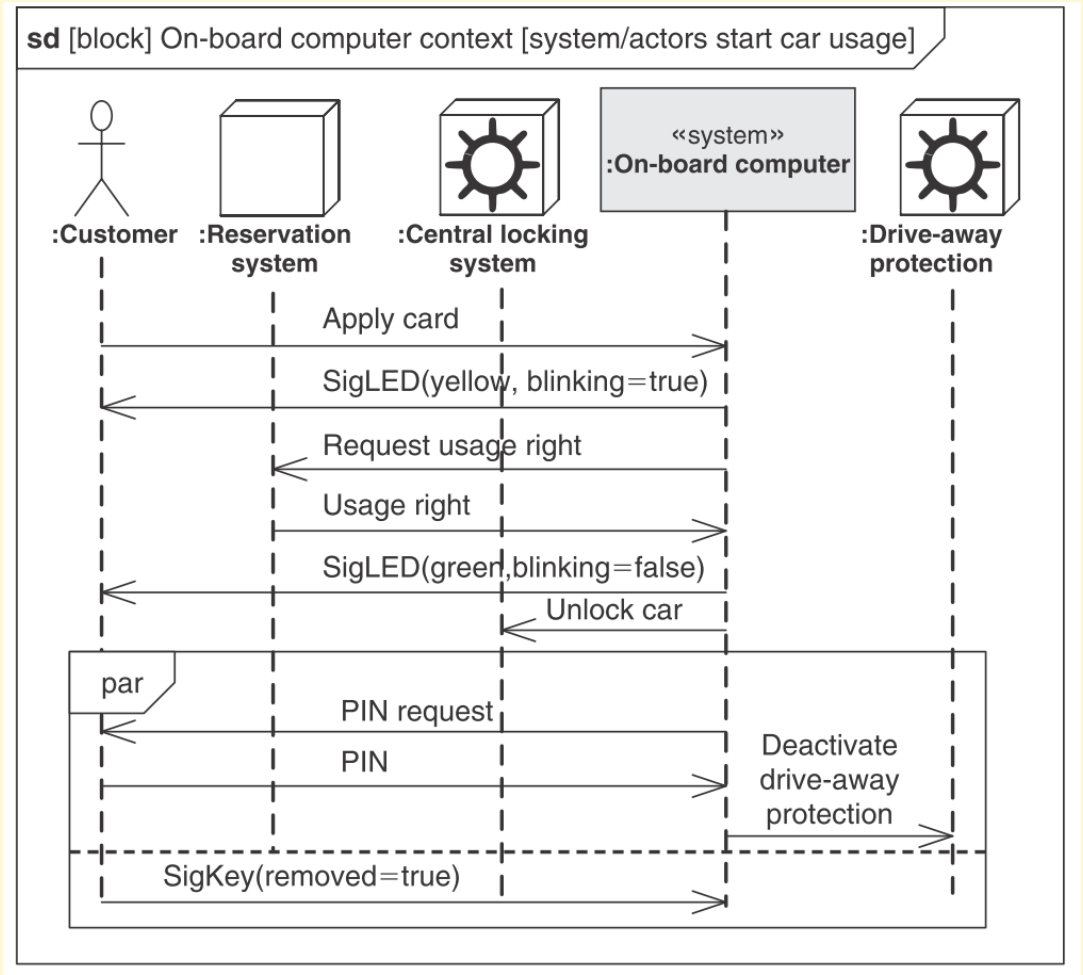
Supports better understanding

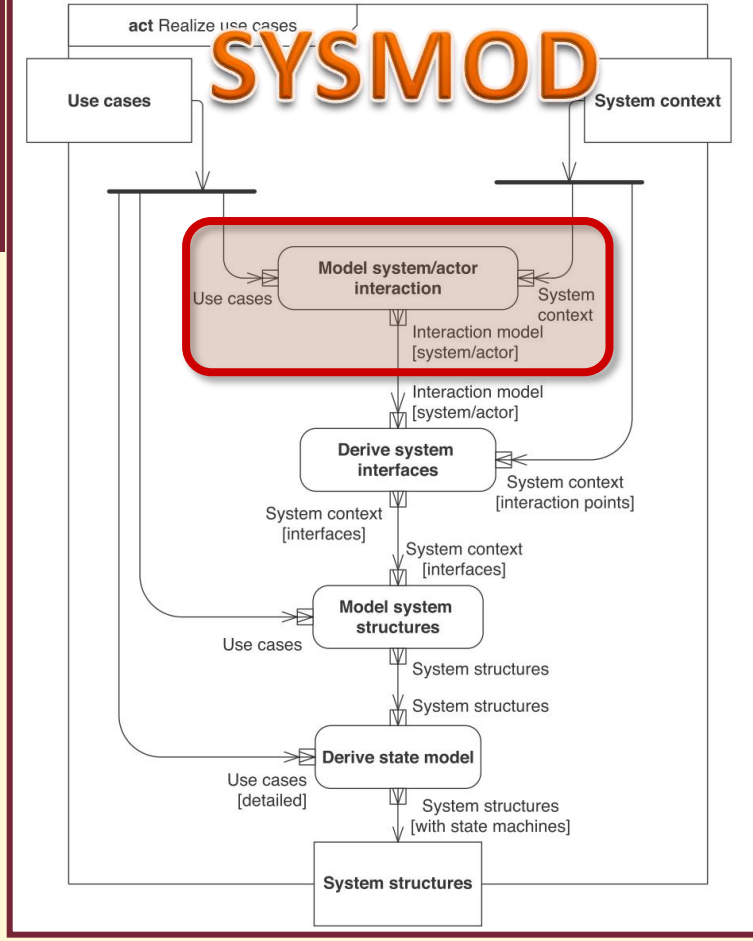
The SYSMOD approach for design



SYSMOD

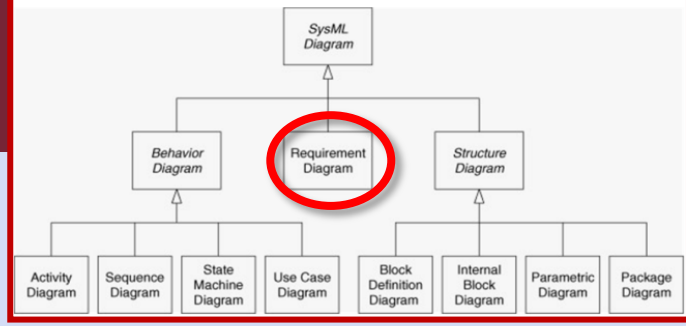
Model System/Actor Interaction

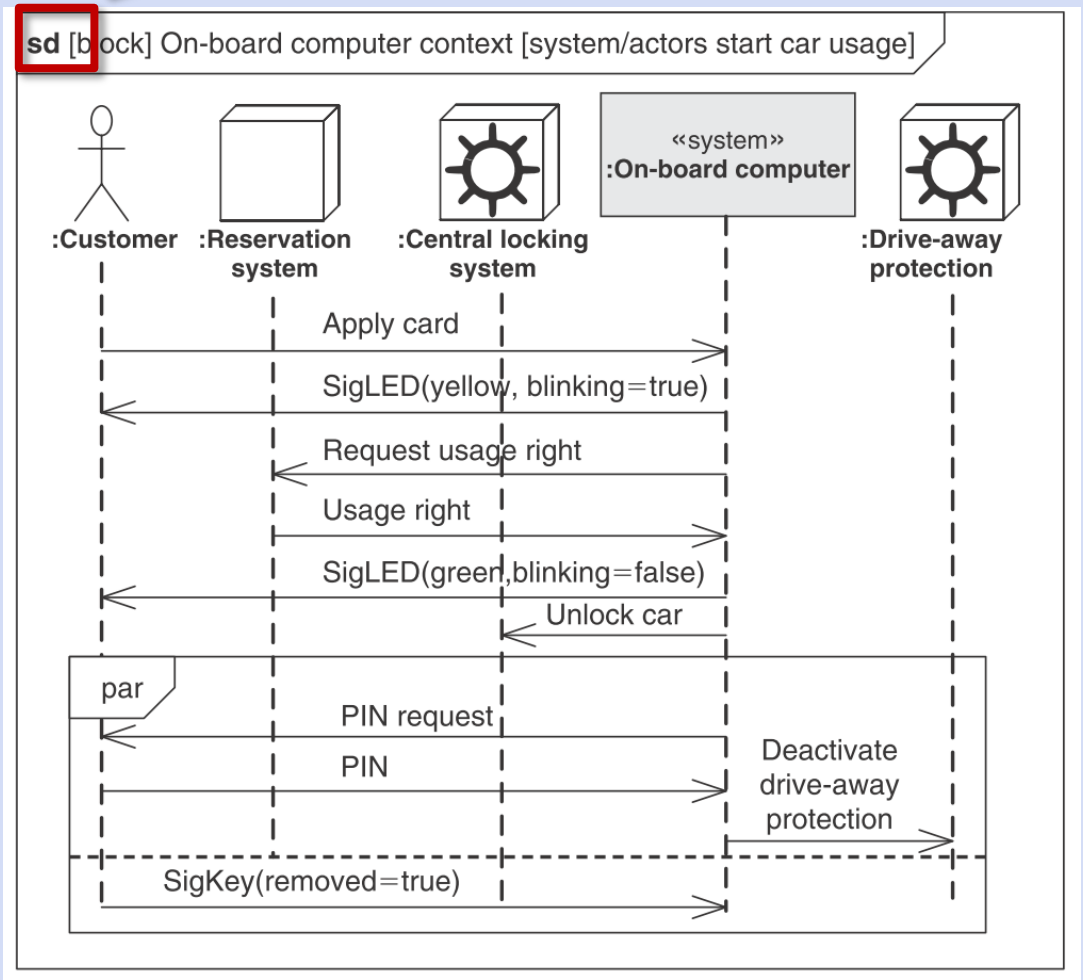




SysML

Sequence diagram

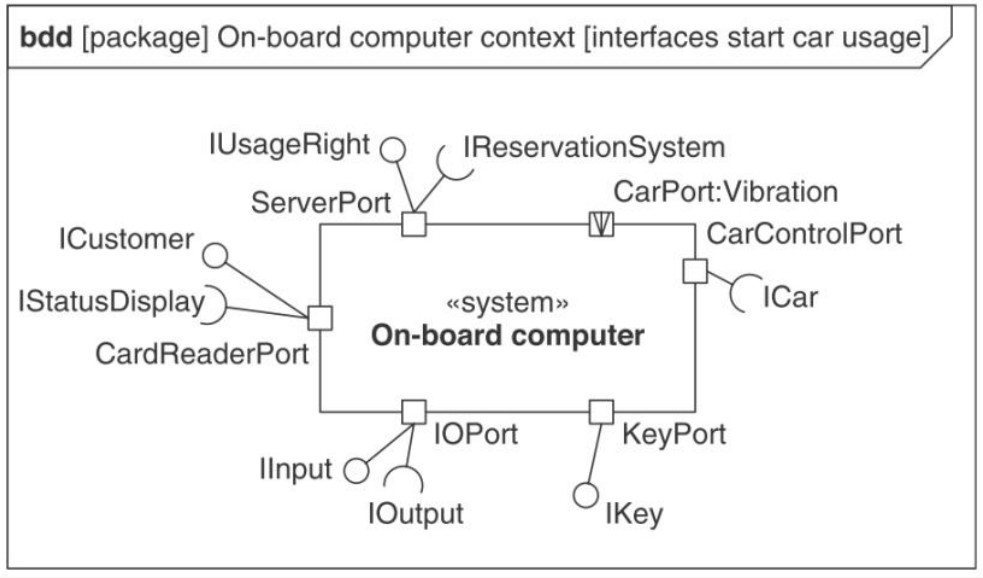


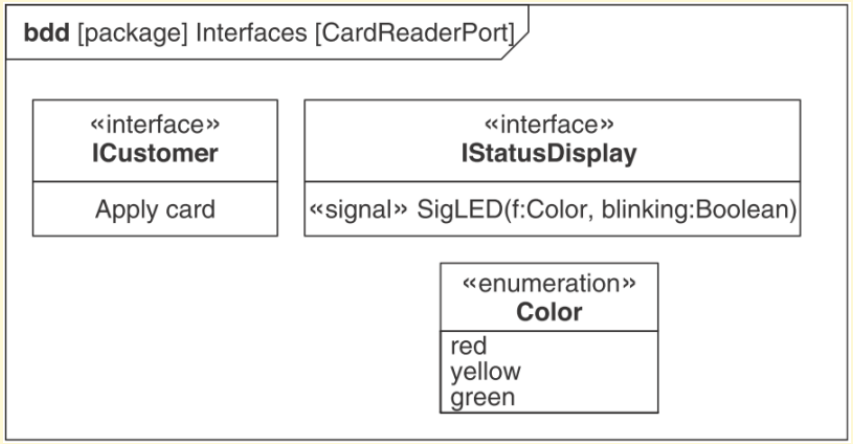


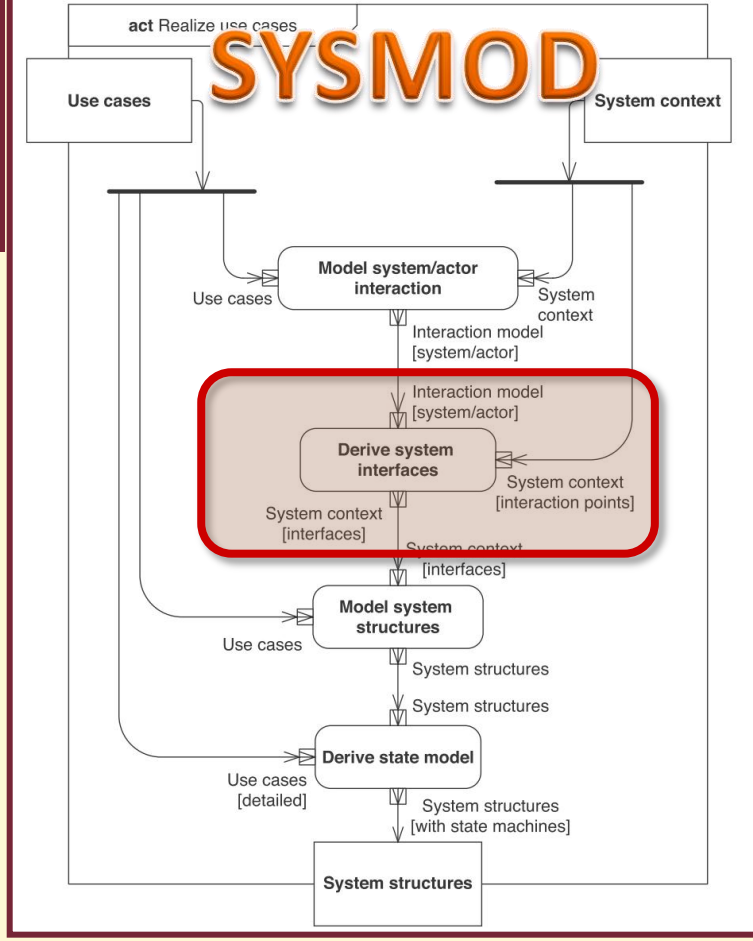
Sequence Diagram

Represents behavior in terms of a sequence of messages exchanged between parts

Derive System Interfaces

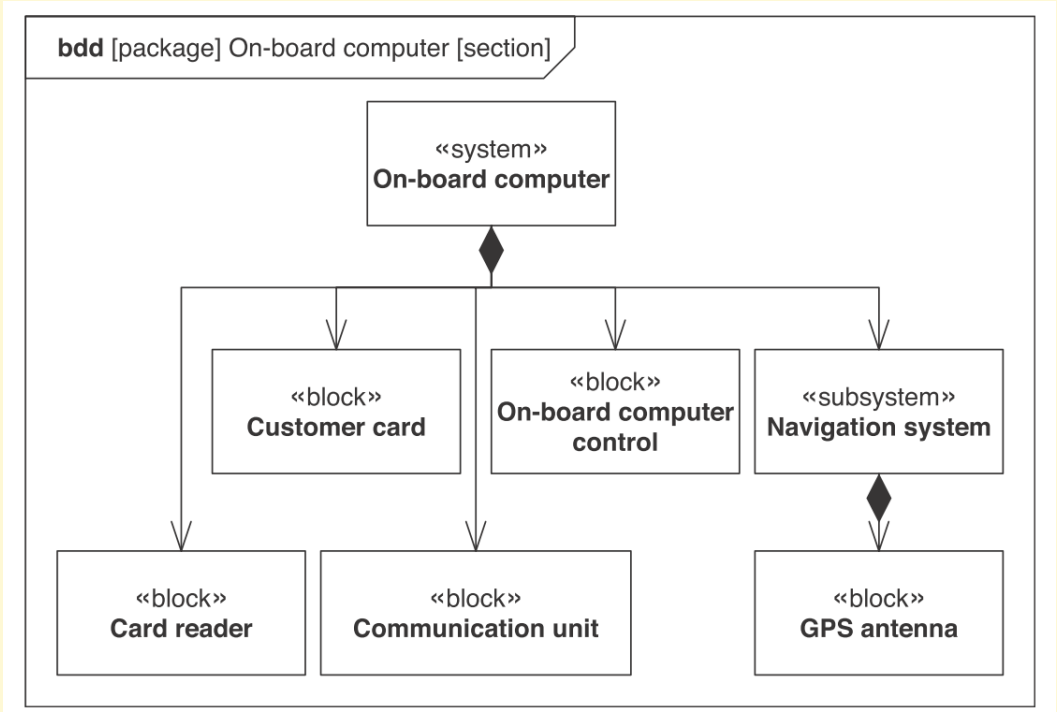


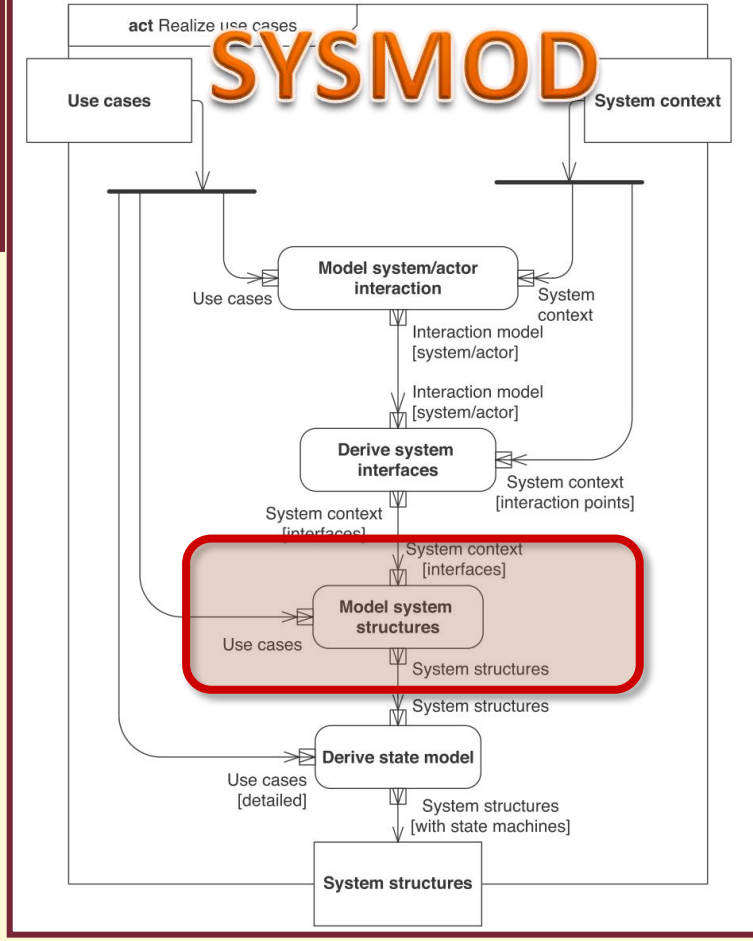




Model System Structures

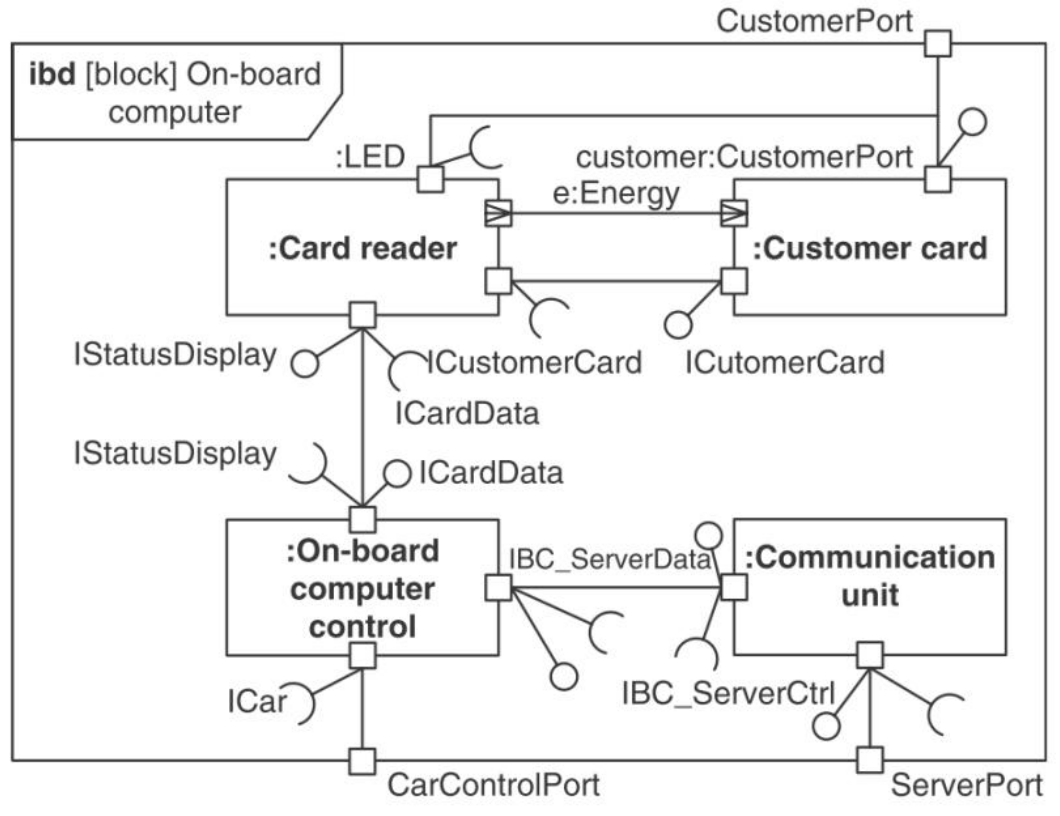
Identify system components

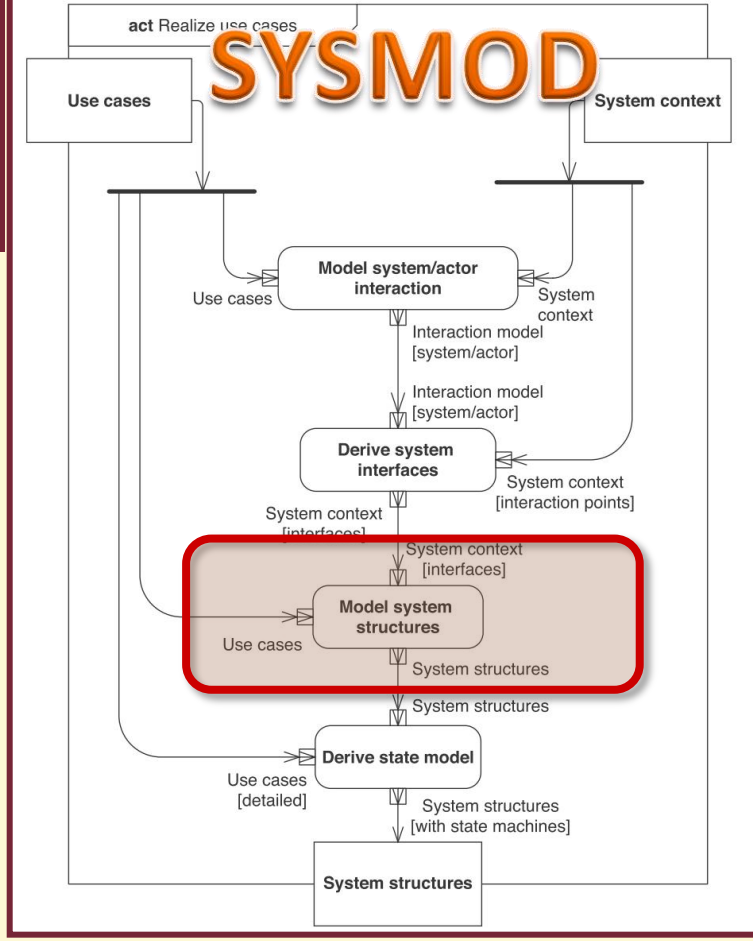




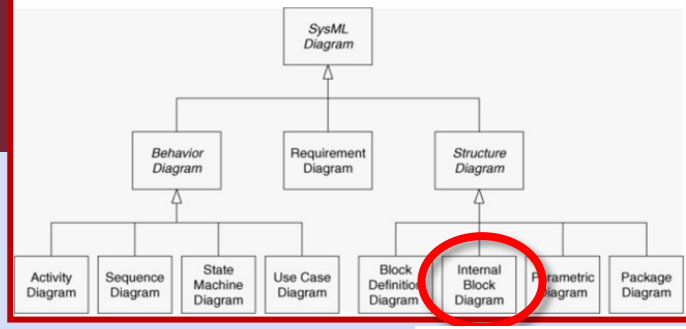
Model System Structures

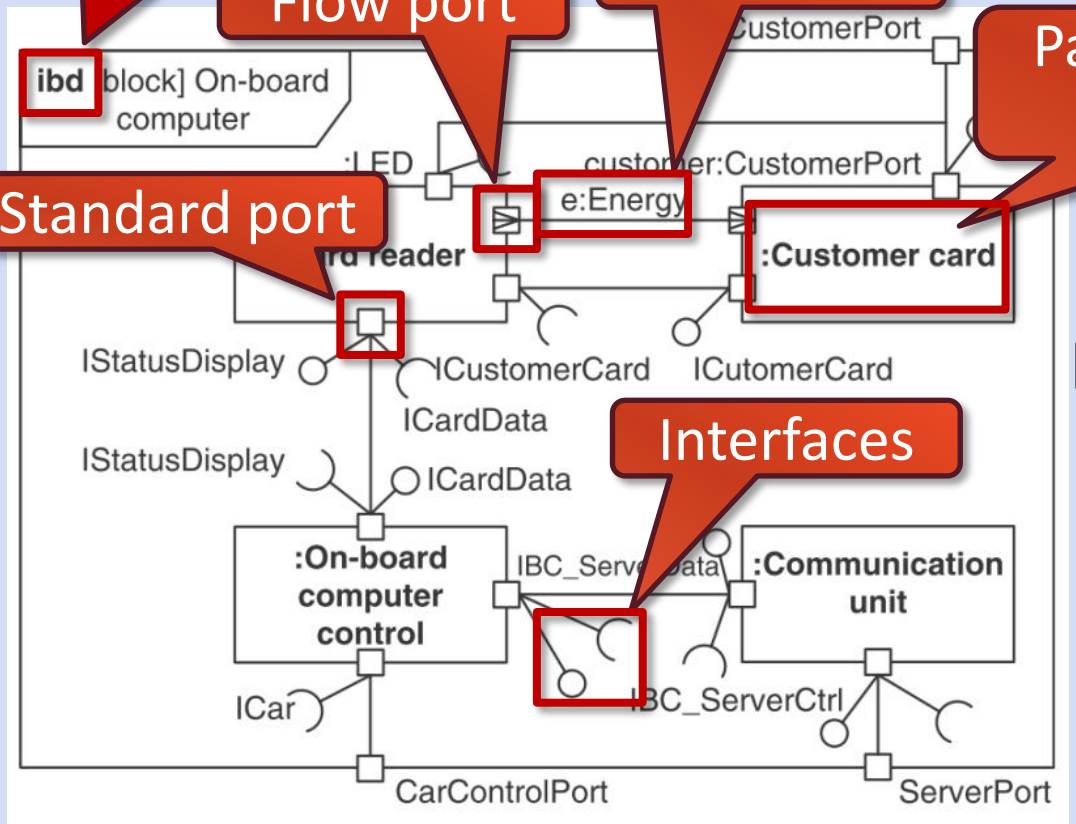
Identify relation of components





SysML



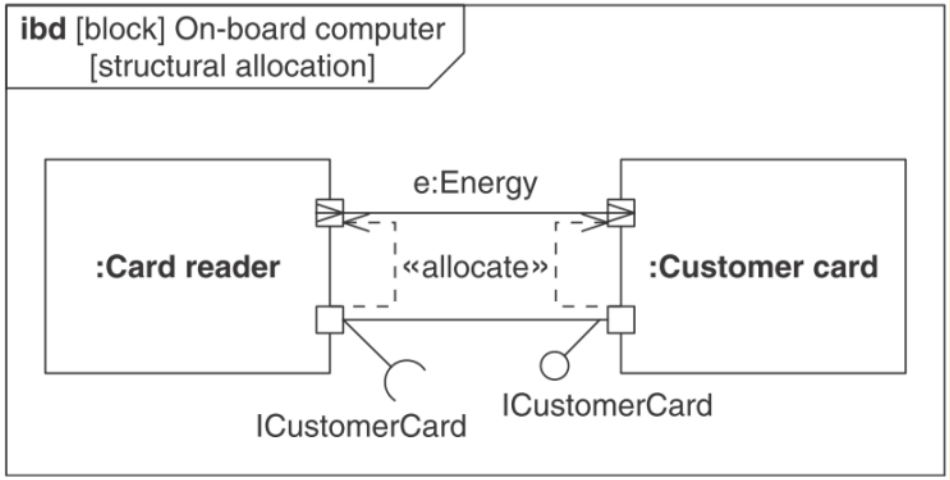


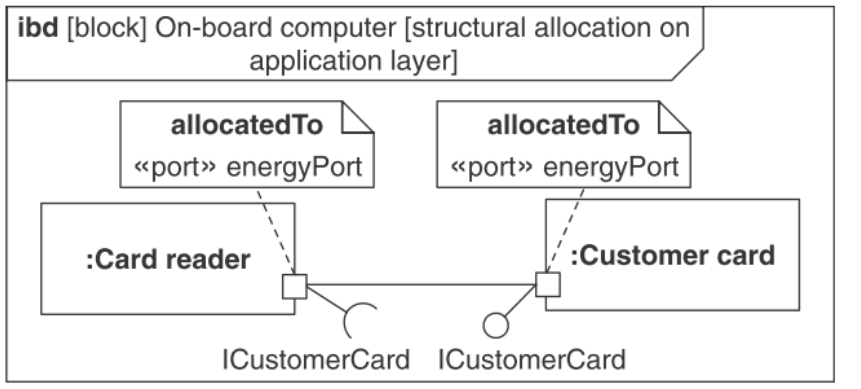
Internal Block Diagram

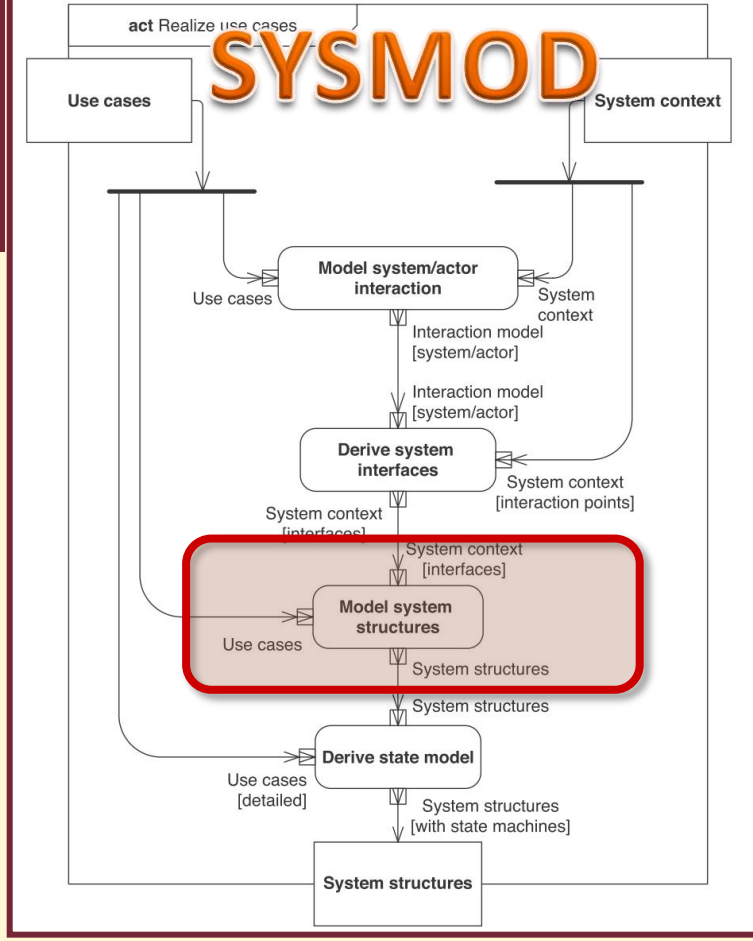
Represents interconnection and interfaces between the parts of a block

Model System Structures

Structural allocation



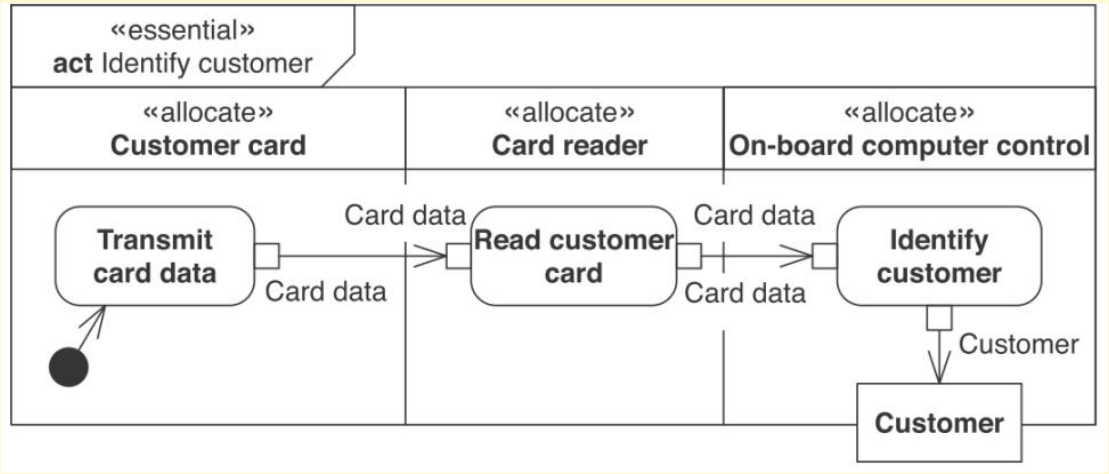


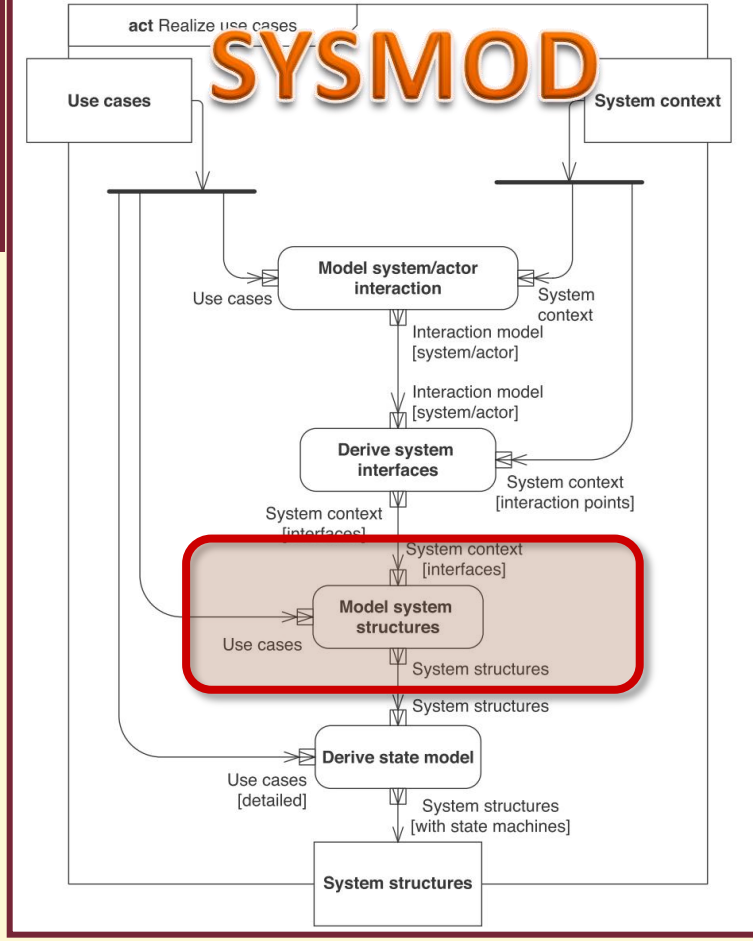


Model System Structures

Functional allocation:

Actions to System components

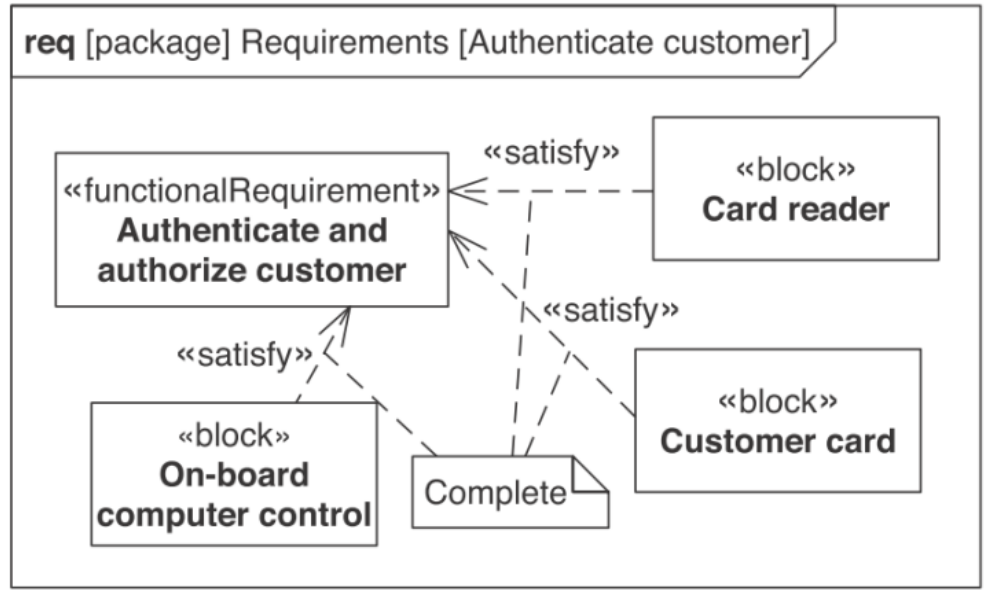


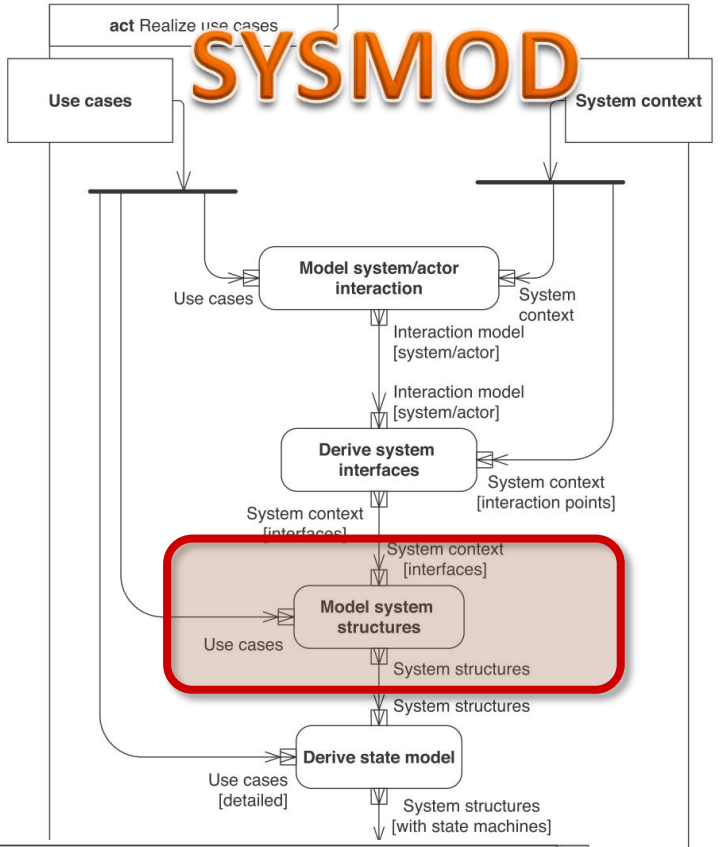


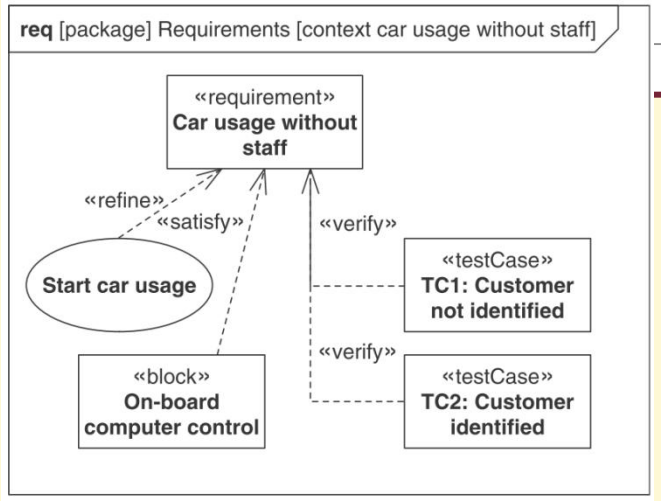
Model System Structures

Traceability to requirements:

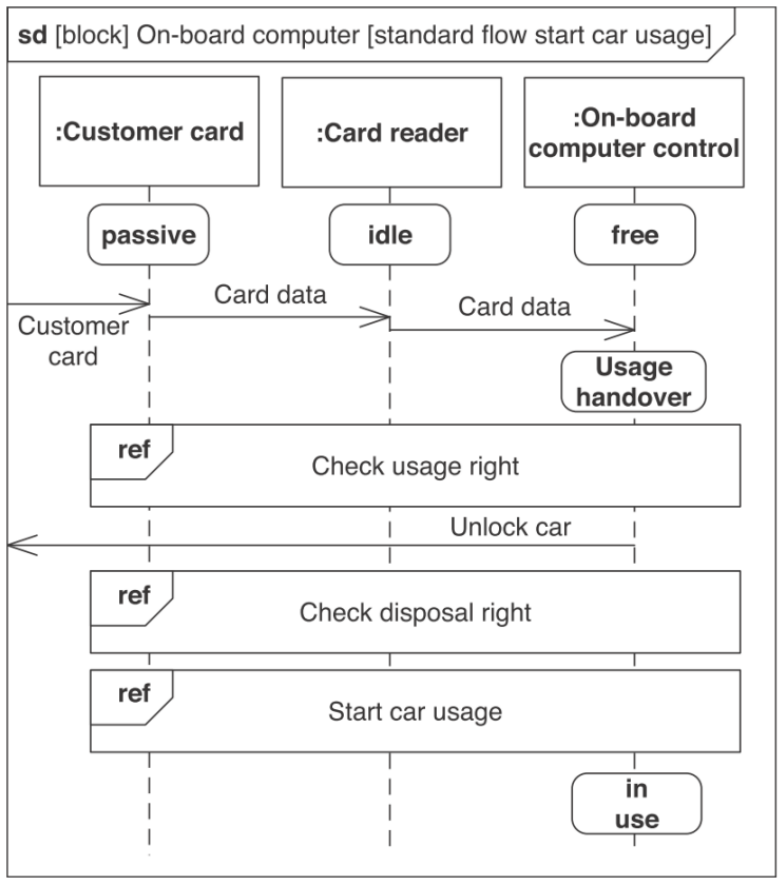
Blocks satisfying a req.

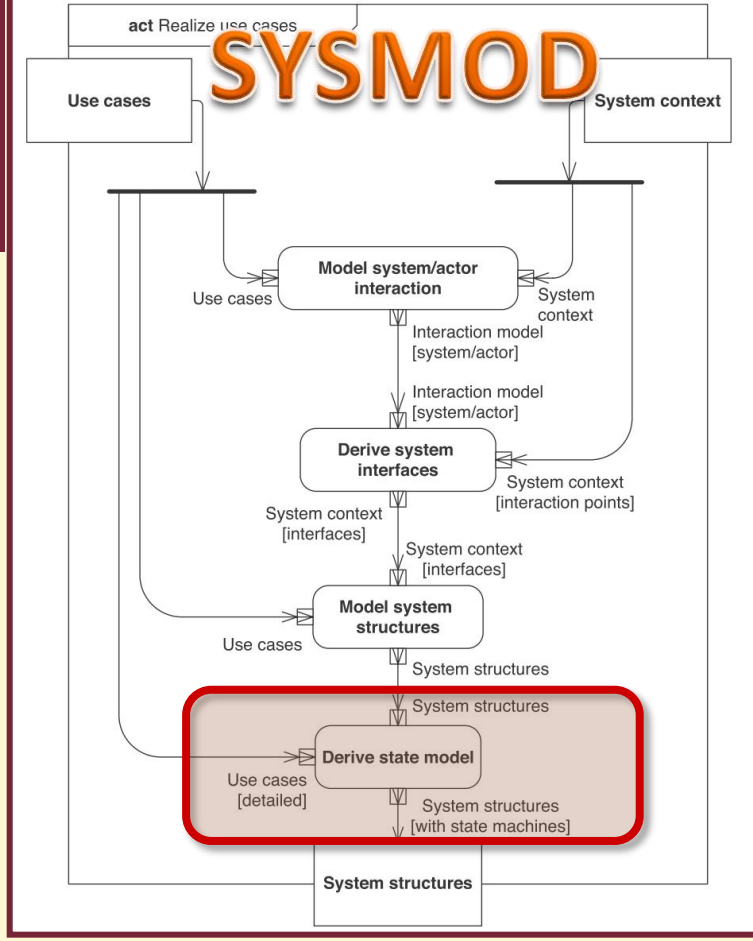




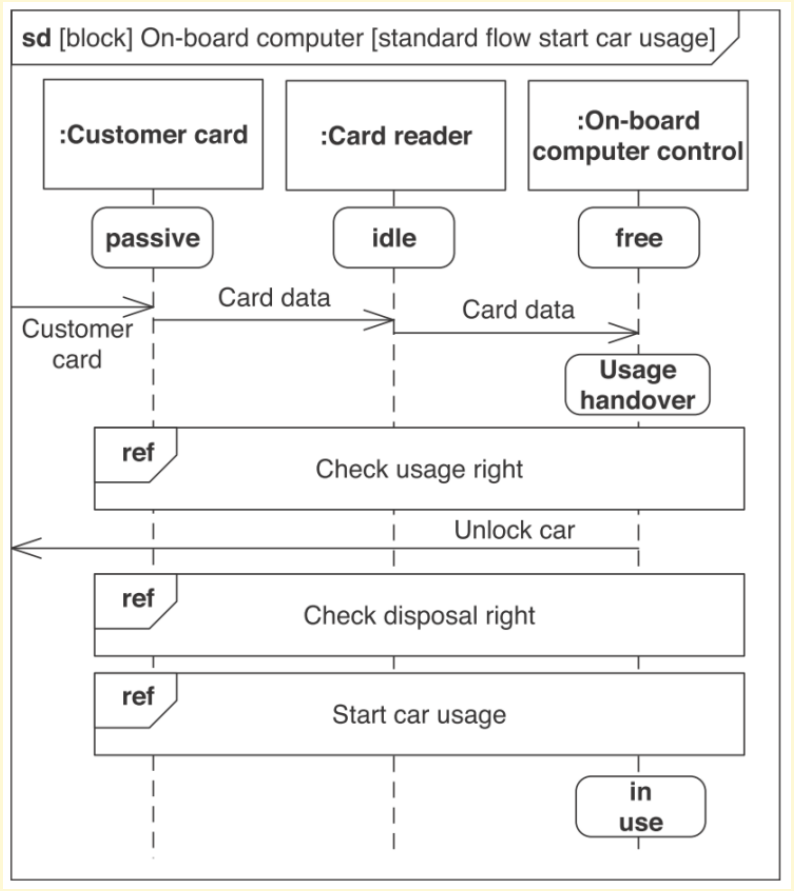


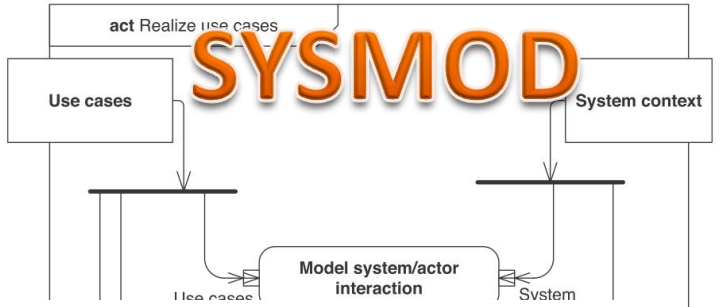
Derive State Model

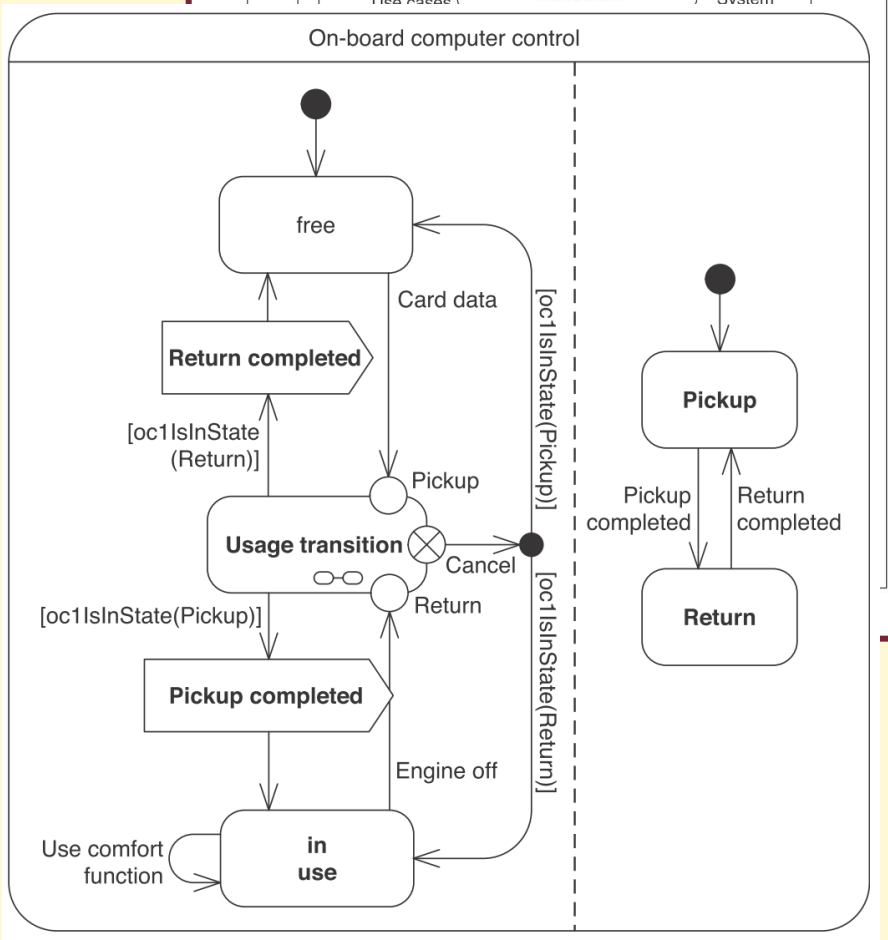




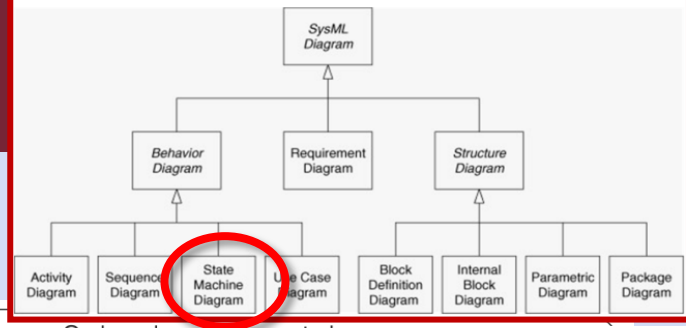
Derive State Model





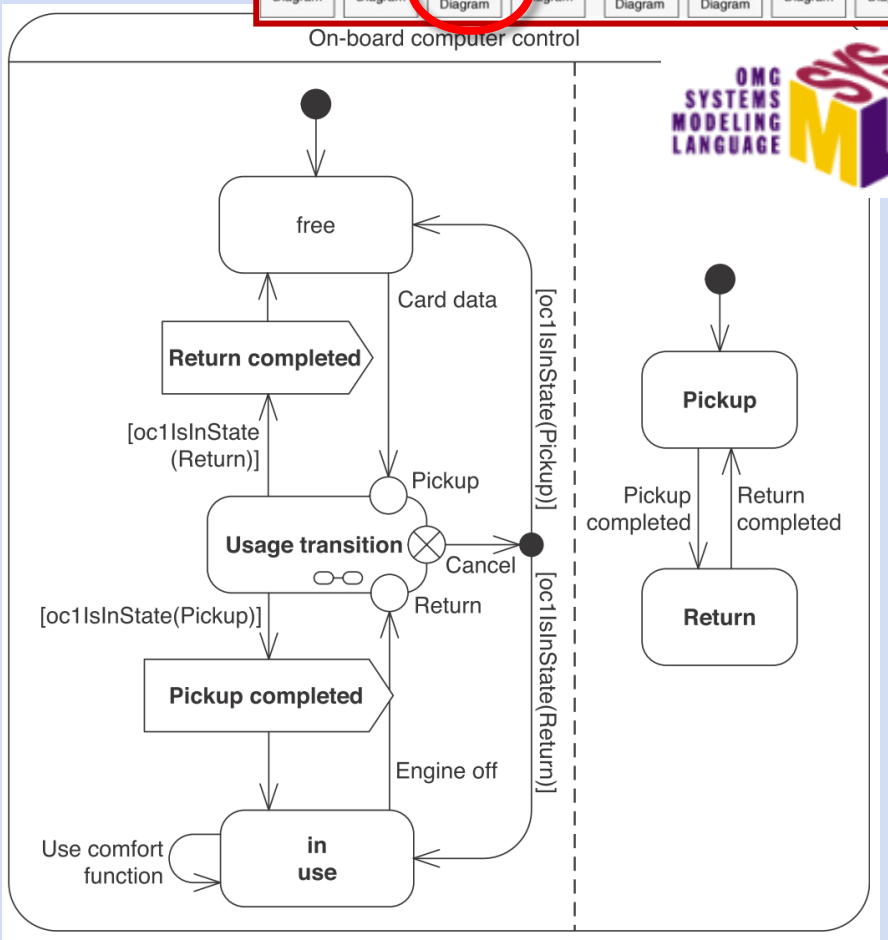


SysML



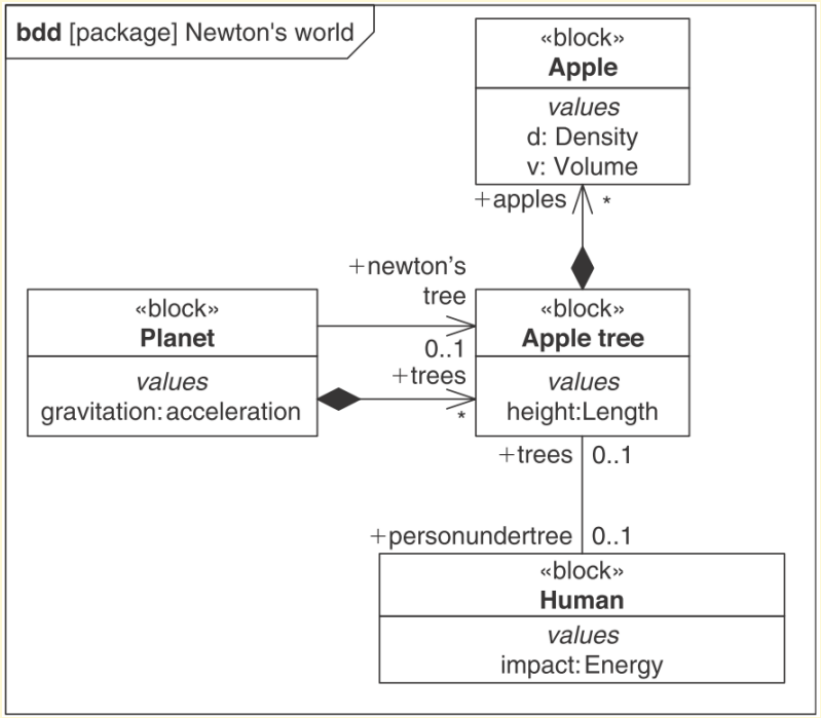
State Machine Diagram

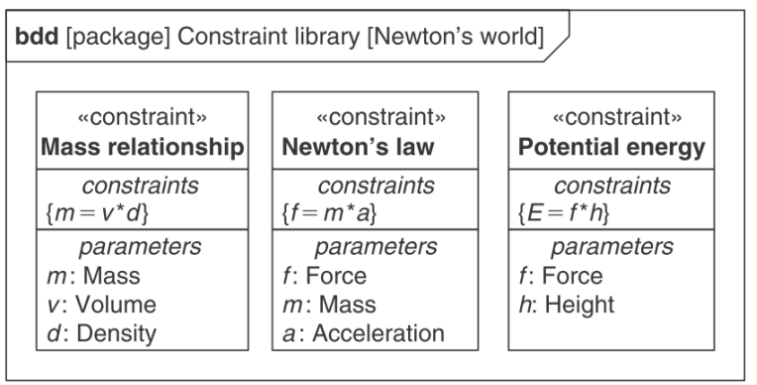
Represents behavior of an entity in terms of its transitions between states triggered by events

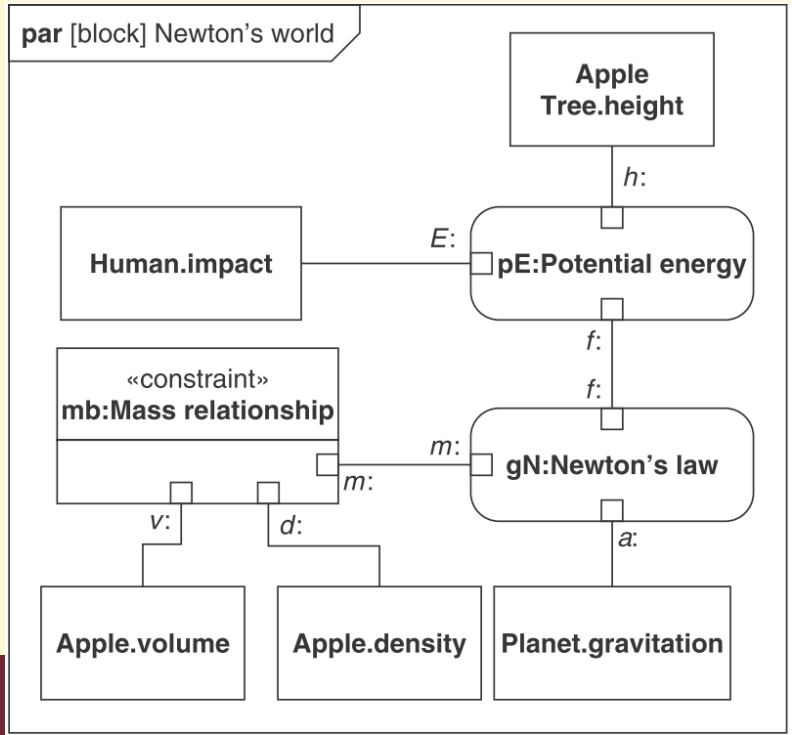


Parametrics

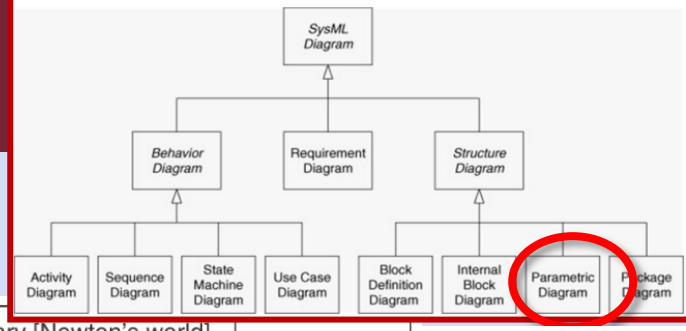
- Constraints on block properties







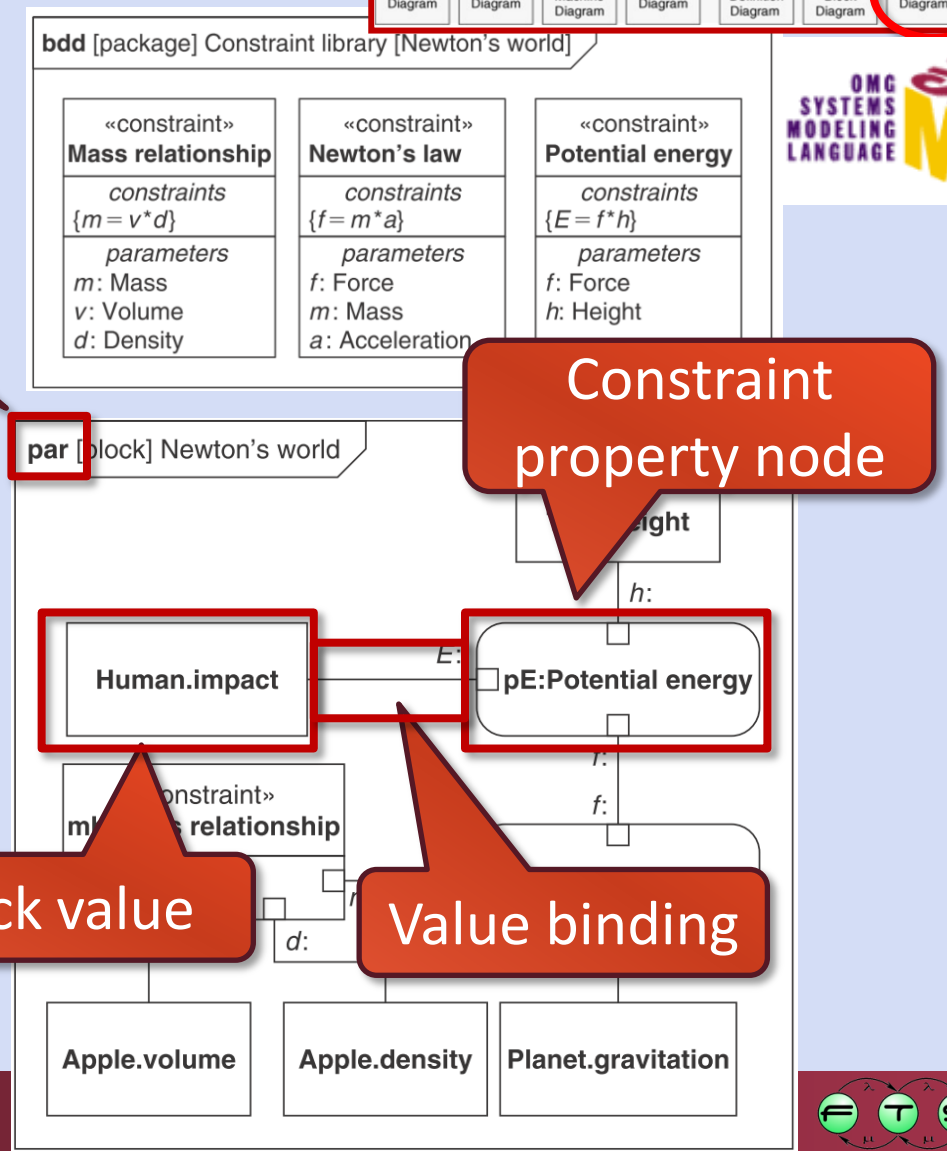
SysML



Parametric diagram

Parametric Diagram

Represents constraints on property values used to support engineering analysis



Constraint property node

Block value

Value binding

Summary

- SysML
  - OMG's most widely accepted standard
  - Heavily used by embedded system engineerings
  - Reuses the „better" part of UML
- SYSMOD
  - Provides a framework for MDE of embedded systems
  - (MDA) → more focused and fits to developer needs