An Engineering and Information Technologies Company

Theodore Kahn
Intrinsyx Technologies Corporation
NASA Ames Research Center
Moffett Field, CA
www.intrinsyx.com

theodore.e.kahn@nasa.gov
theokahn@gmail.com

NASA/Army Systems and Software Engineering Forum
May 12 & 13, 2009
Marshall space Flight Center

MODEL BASED PROJECT MANAGEMENT

Applying the Systems Modeling Language to Project Management

Problem statement

- **Project and system information is now document centric:**
  - Relies on textual, tabular, and graphic documents from a variety of sources which means…
  - interpretation of the information is difficult and consistency hard to maintain, especially under changing conditions.
- **Model centric approach relies far less on documents:**
  - Depends instead on a single repository for storing project structural and behavioral information…
  - Promotes a more common interpretation and consistency of project information, even under very dynamic conditions.

Presentation objectives

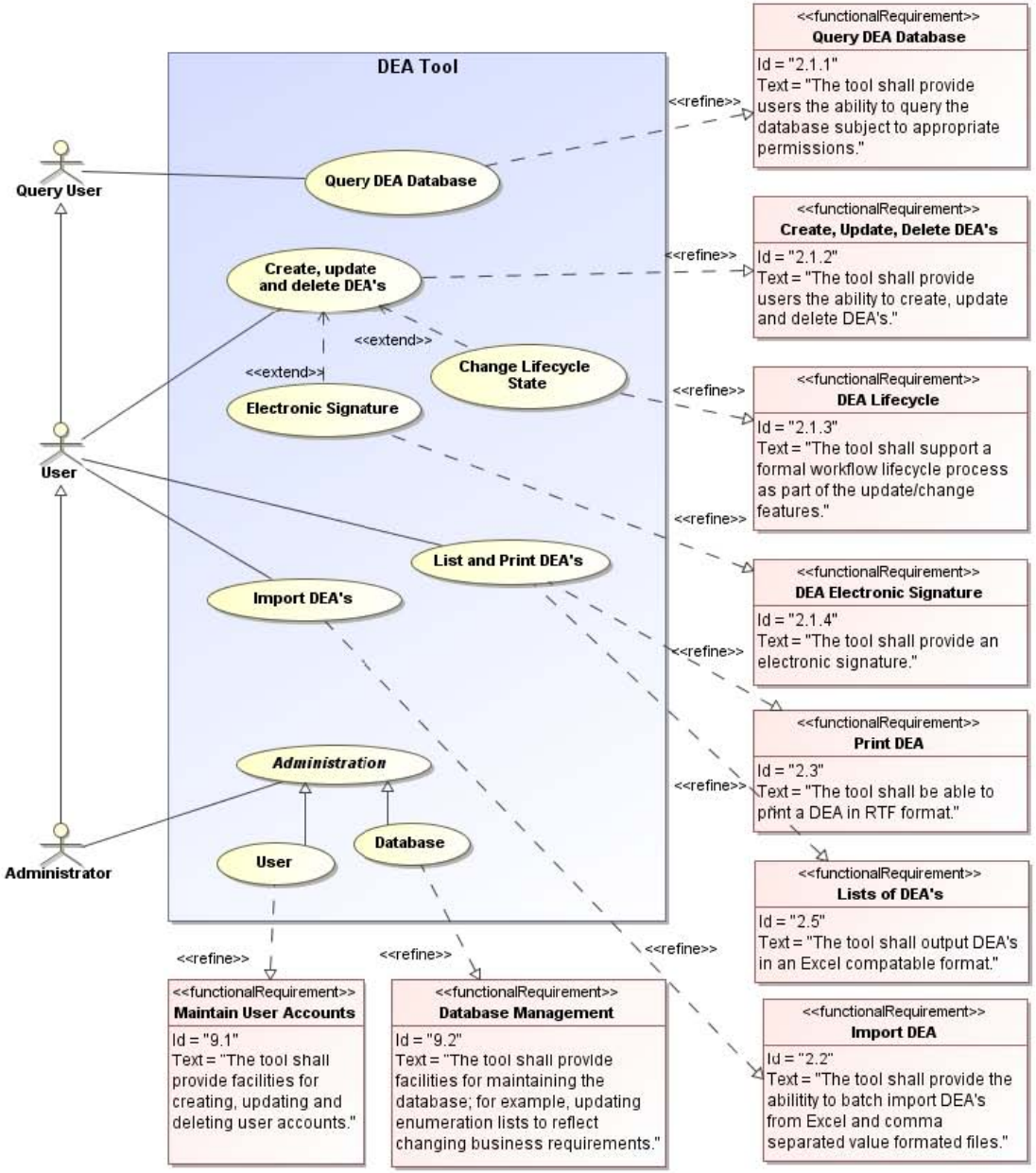
- Provide a conceptual understanding of modeling as it might be applied to project management and system engineering.
- Enable you to make informed decisions regarding its applicability to your situation.

The talk does not address implementation issues or methodology, though those would be interesting areas for the discussion at the end.

Presentation overview

- Some basic SysML semantics
- Theoretical aspects of modeling
- Demonstration of a SysML tool and models
- Q & A, and discussion

DEA Tool Functional Use Cases



Conclusion

- **Why you don't want to model**
  - Modeling is hard
  - Modeling tools are difficult
  - Modeling will likely require cultural changes
- **Why you do want to model**
  - It increases the rate of communications
  - It increases the precision of communications
  - It reduces tacit information
  - It promotes a common understanding of your project
- **Benefit:** Stakeholders having a common understanding of how the project is organized and its objectives will work together more effectively and make better decisions.

Theoretical aspects of modeling

- **SysML and project management**
- Modeling in the context of SysML
- SysML structure and semantics
- Conclusion

Relationship between project management and systems engineering

While project management and systems engineering share many of the same qualities, there are important differences.

Similarities between PM & SE

- Lots of behaviors having complex relationships
- Lots of entities having complex relationships
- Many relationships and dependences between behaviors and entities.

Systems engineering

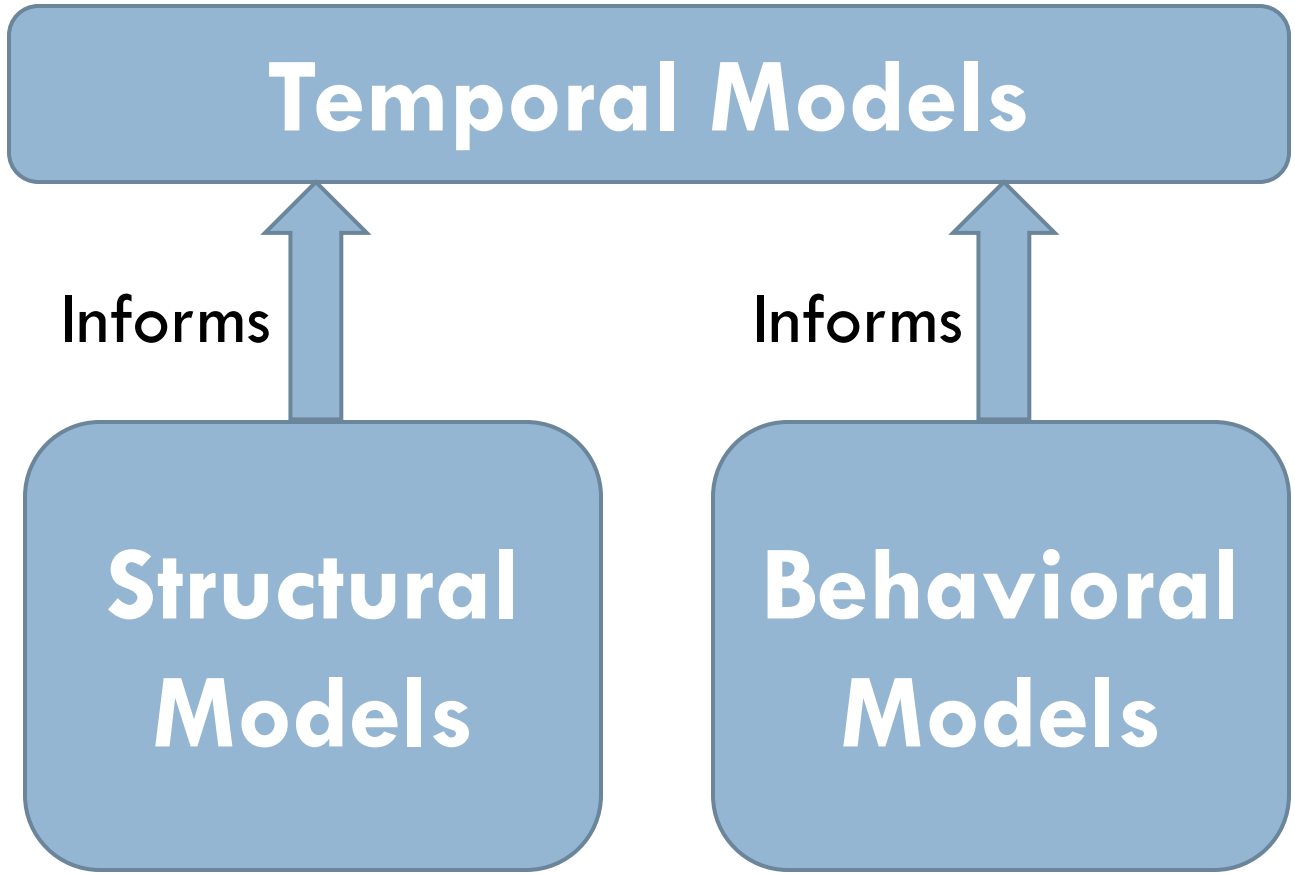
- Systems engineering is a multidisciplinary approach for developing balanced systems solutions in response to diverse stakeholder needs.
- It includes the application of both management and technical processes to achieve balance and mitigate project risk.
- The management process is applied to ensure that development cost, schedule and technical performance objectives are met.

Differences between PM & SE

Attribute	Project Management	Systems Engineering
Scheduling	Absolute times	Relative times
Economic cost	Mandatory/aggregation centric	Optional/unit centric
People	Individuals and roles	Role centric
Entities and behaviors	Notional	Precise

Project Management Models

Gantt, PERT Charts



SysML

Being a successful PM or SE means:

- Understanding the project/system information and how it fits together.
- Communicating and otherwise making this information available to stakeholders in a timely, consistent fashion in a form relevant to their backgrounds and needs.

Part II, SysML

- SysML and project management
- **Modeling in the context of SysML**
- SysML structure and semantics
- Summary

SysML modeling is about communications

- It's a language having syntax and structure…
- that uses a medium, primarily graphics…
- and has a methodology, which currently is largely undefined.

Three main attributes of languages

- Abstractions of the world around us
- Some form of persistence
- A shared experience: a producer and a consumer

Three main attributes of SysML

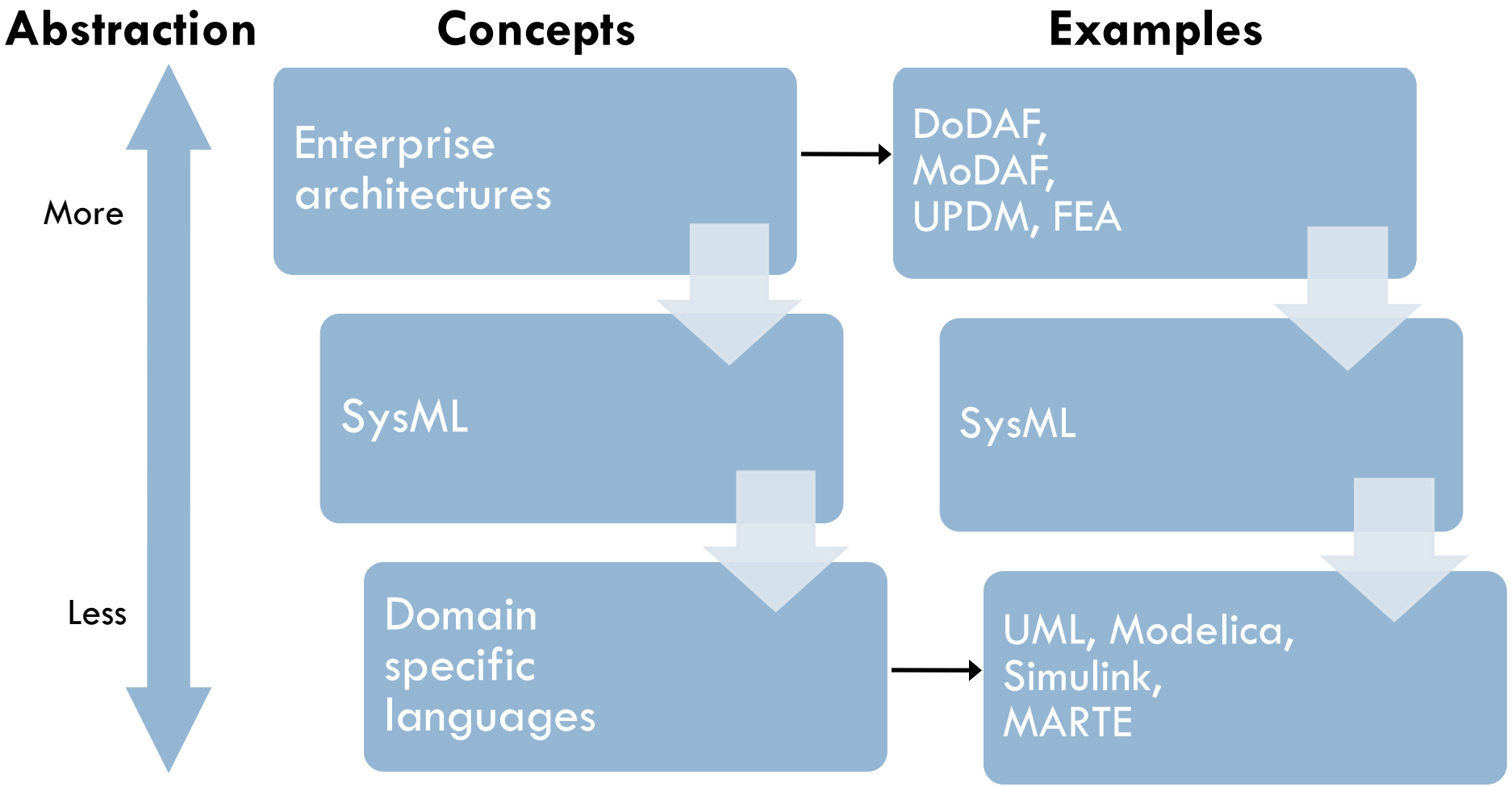
- Abstractions of systems
- Database persistence
- A shared experience

Model, model on the wall…

who's the truest of us all:

- An electrical schematic of a radio
- An economic model
- A model student
- A non working model airplane
- A novel about present day life the author believes to be possible
- A description of a pencil

SysML and other modeling constructs



Abstraction levels

La Joconde

Femme au Chapeau Orné

Two criticisms of SysML…

- **Its old ideas warmed over** – Abstractions and persistence have been around at least since humans drew pictures on cave walls. There is nothing new here.

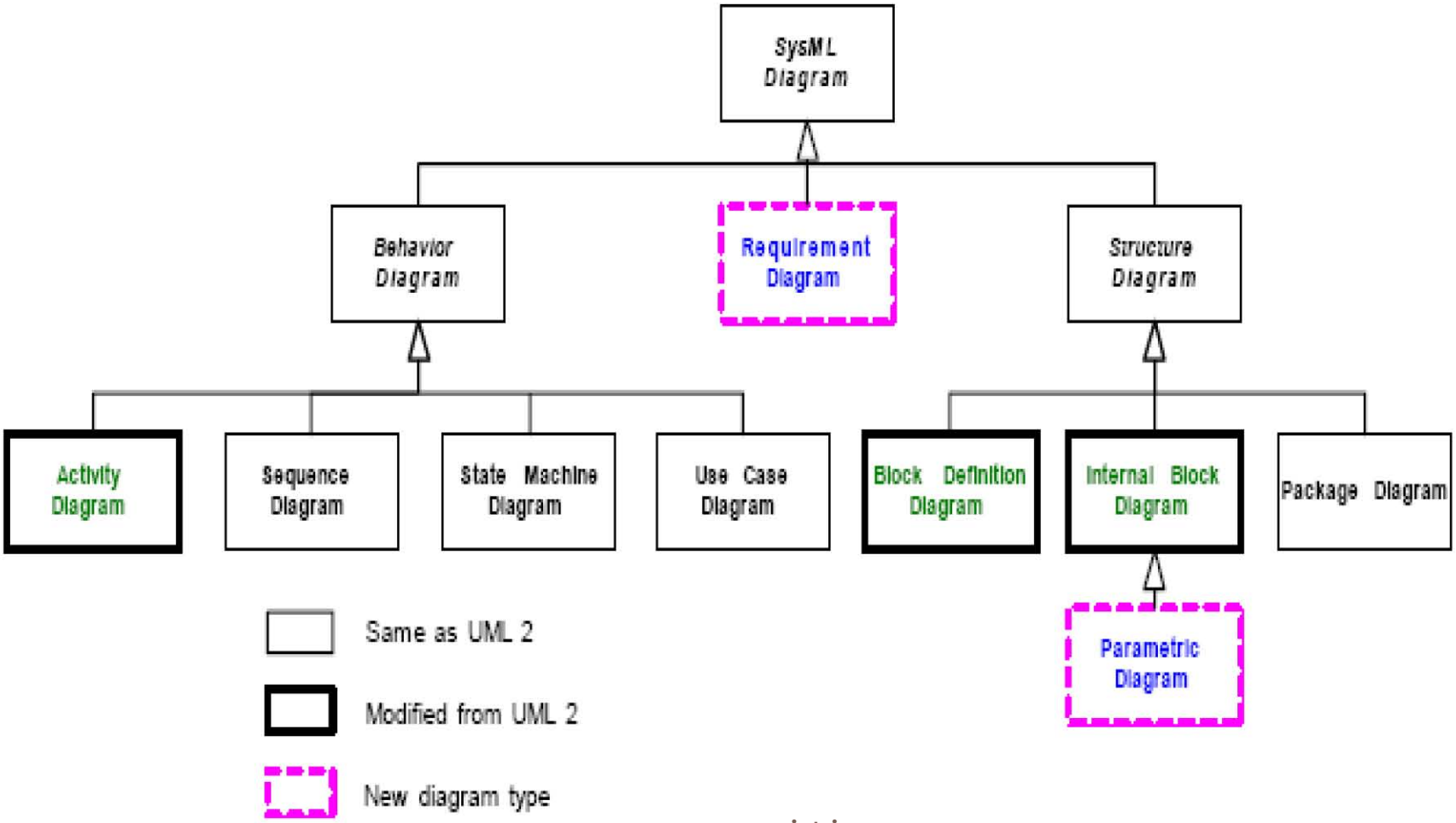
- **Its simply not practical** – Having my team think abstractly in the same way and put their information into a database in the same fashion is absurd. It is not workable.

A historical perspective

Part II, SysML

- SysML and project management
- Modeling in the context of SysML
- **SysML structure and semantics**
- Summary

SysML Diagram Taxonomy



Source: OMG Specification

Model elements

- Models consist of elements
- Elements must have unique names, within a namespace
- All model elements must reside in (be owned by) one package

Major types of model elements

- Structural
- Behavioral

Packages

- Models are composed of one or more packages
- Packages can contain other packages and/or any collection of model elements
- Package hierarchies define the name space

Packages shown as a containment tree

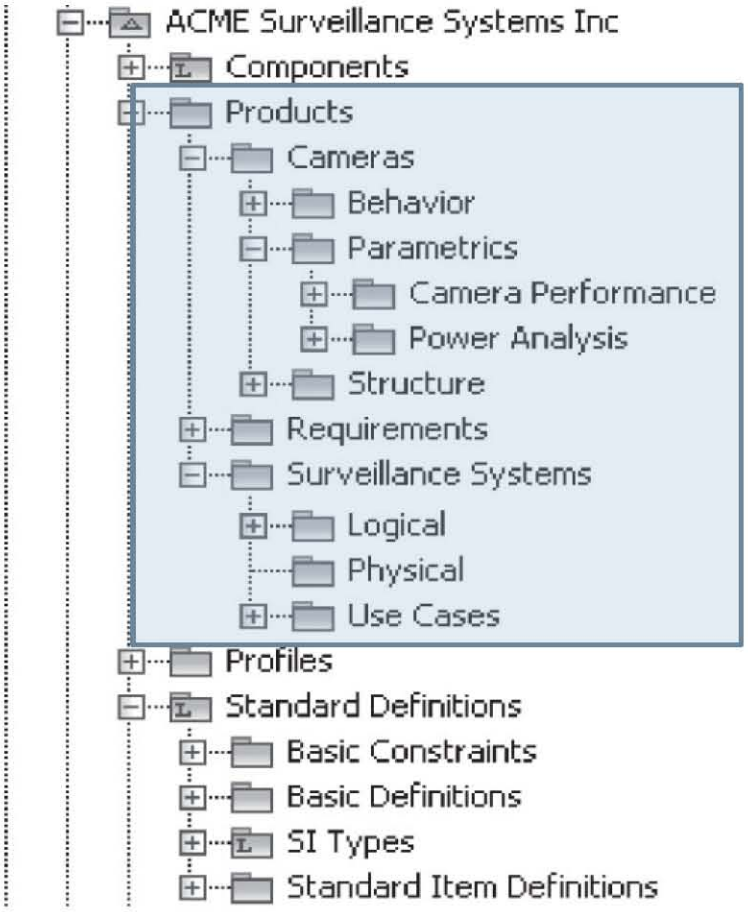


FIGURE 5.4

Browser view of the model's package hierarchy.

Packages shown graphically

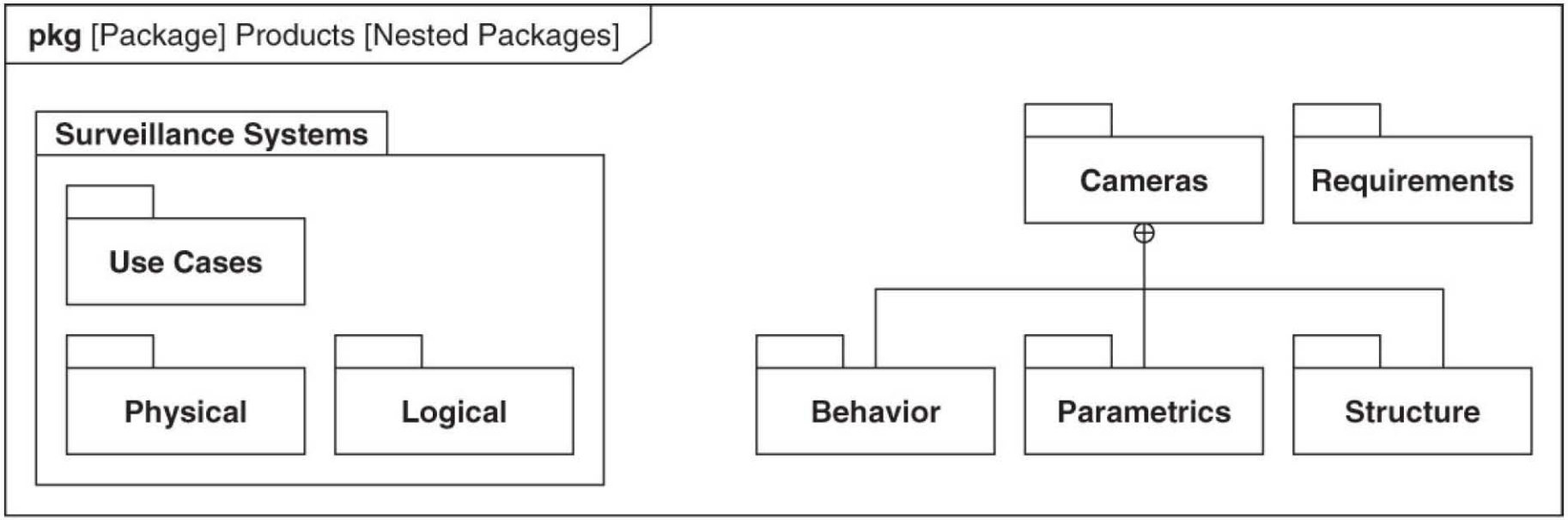


FIGURE 5.1

An example package diagram.

Dependences and stereotypes

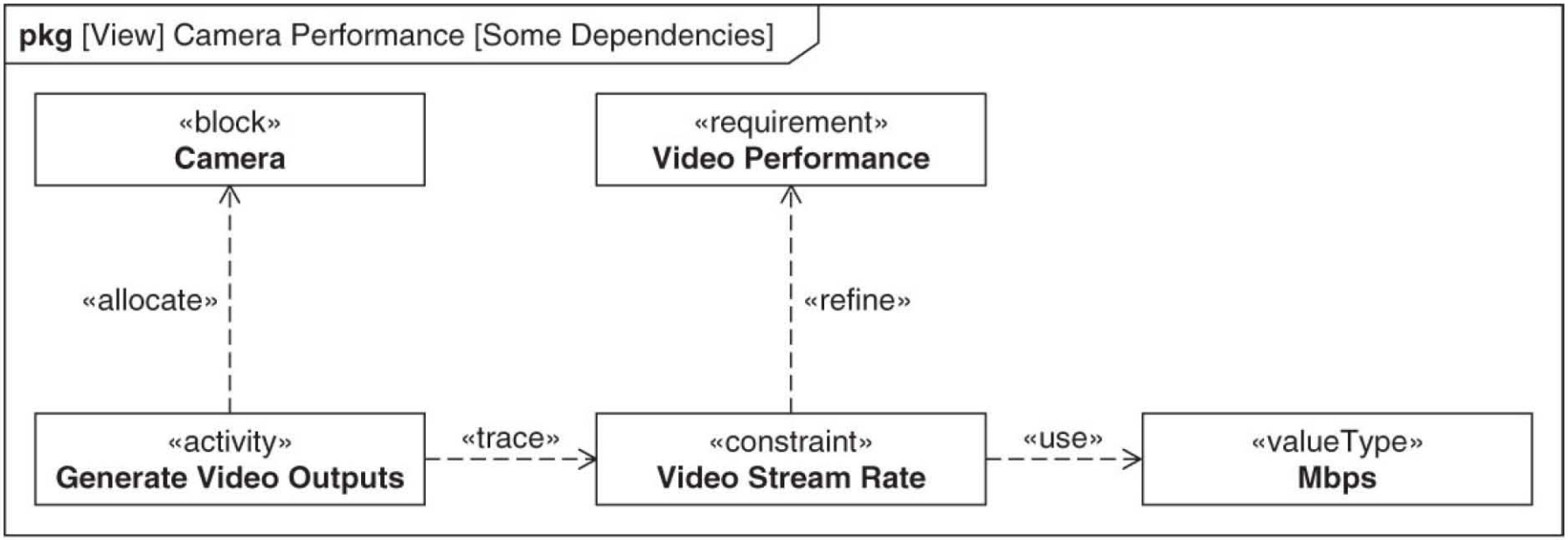


FIGURE 5.10

Example of dependencies in the camera performance view.

Blocks

- Blocks represent structural elements
  - Organizations
  - Data
  - People
  - Airplanes
- Blocks have two main attributes
  - Properties, other structural elements
  - Behavior, either as an intrinsic capability or through behavioral elements

Block property types

- Parts (in UML, composition)
- References (in UML, aggregation)
- Associations

Block definition diagram—Part

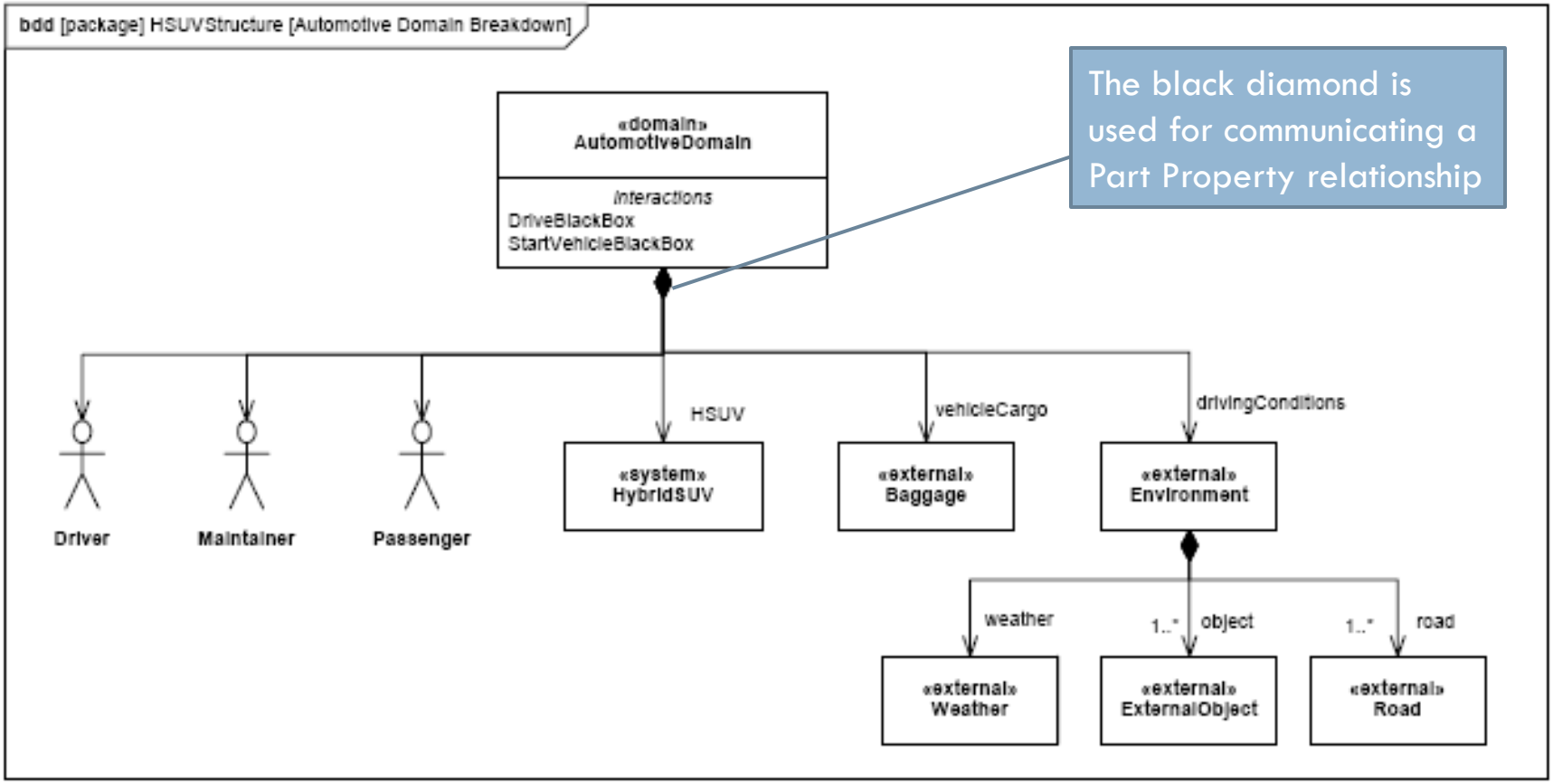


Figure B.15 - Defining the Automotive Domain (compare with Figure B.4) - (Block Definition Diagram)

Source: OMG Specification

Block definition diagram—Association

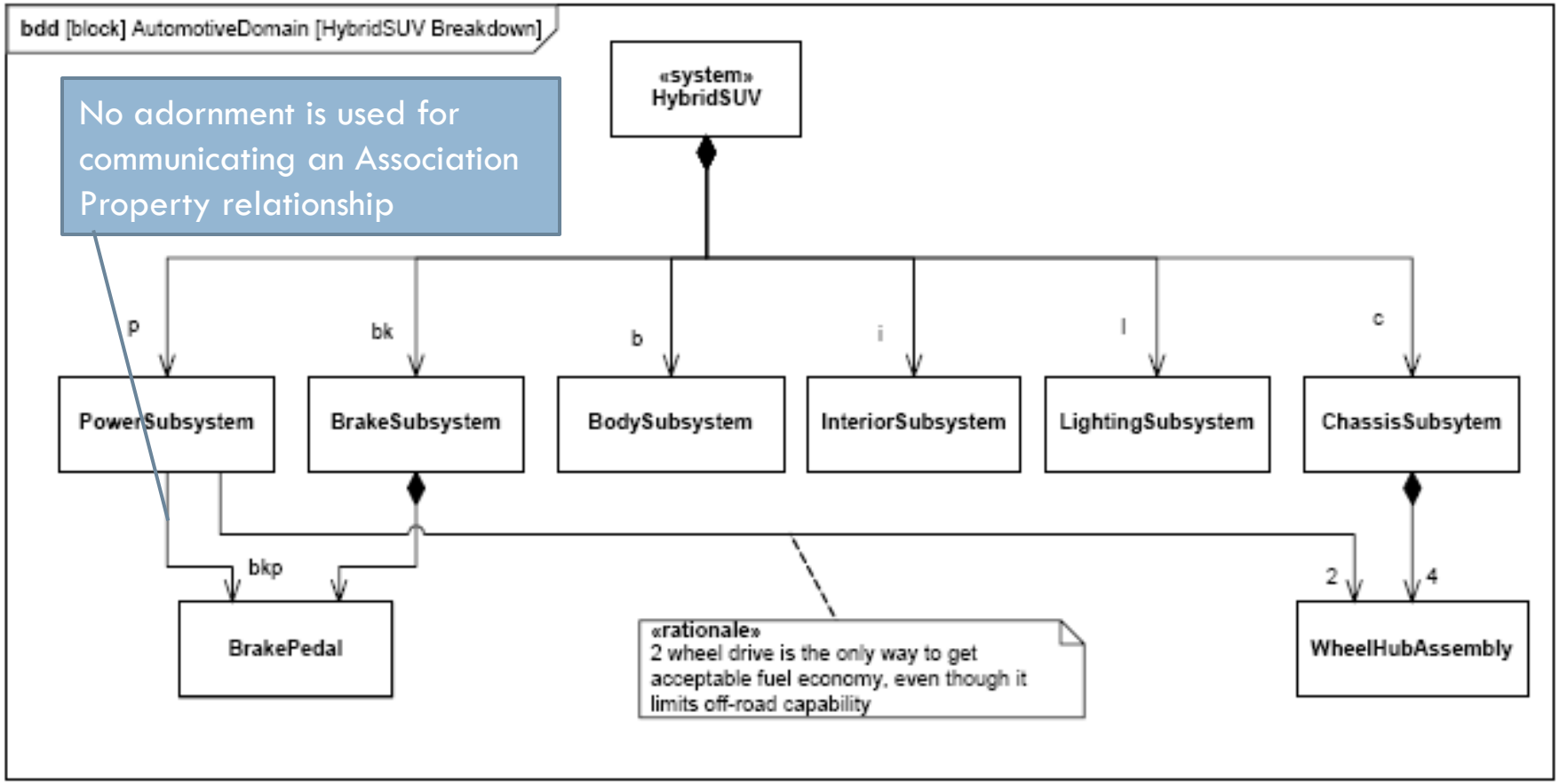


Figure B.16 - Defining Structure of the Hybrid SUV System (Block Definition Diagram)

Source: OMG Specification

Block definition diagram—Reference

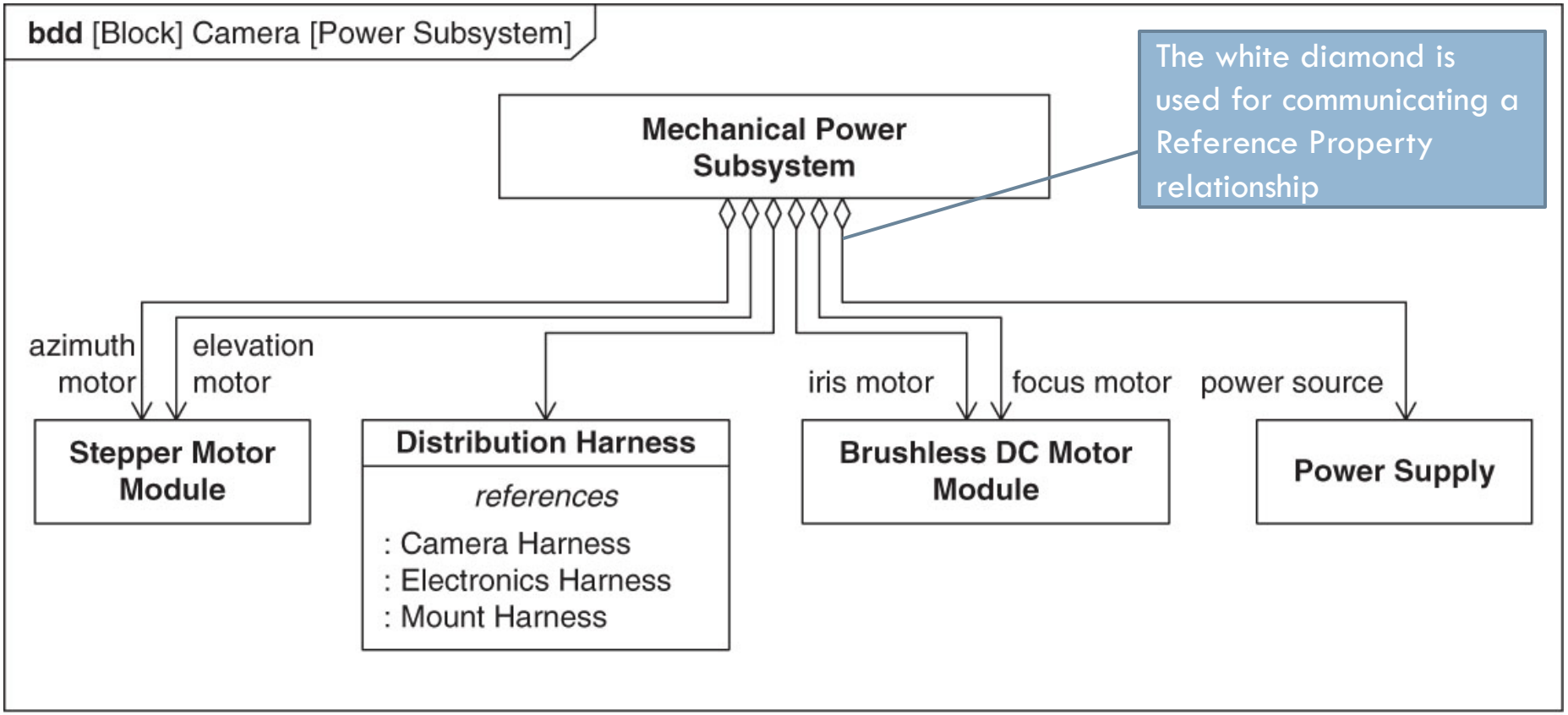


FIGURE 6.10

A reference association on a block definition diagram.

Block compartments

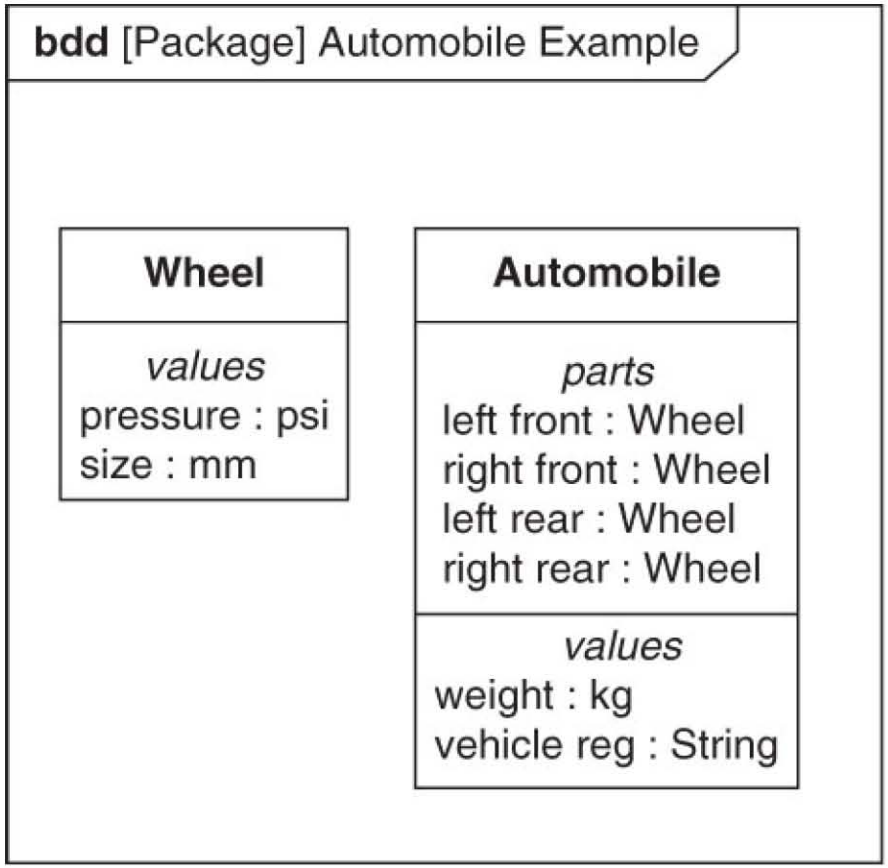


FIGURE 6.4

An automobile with four wheels described as separate parts.

Blocks and inheritance

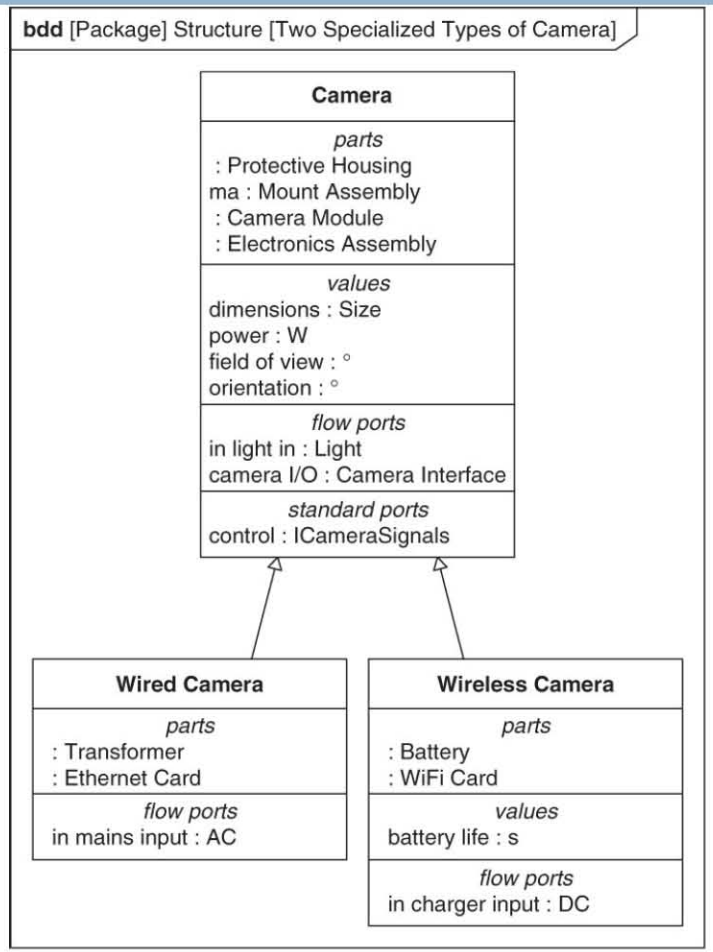
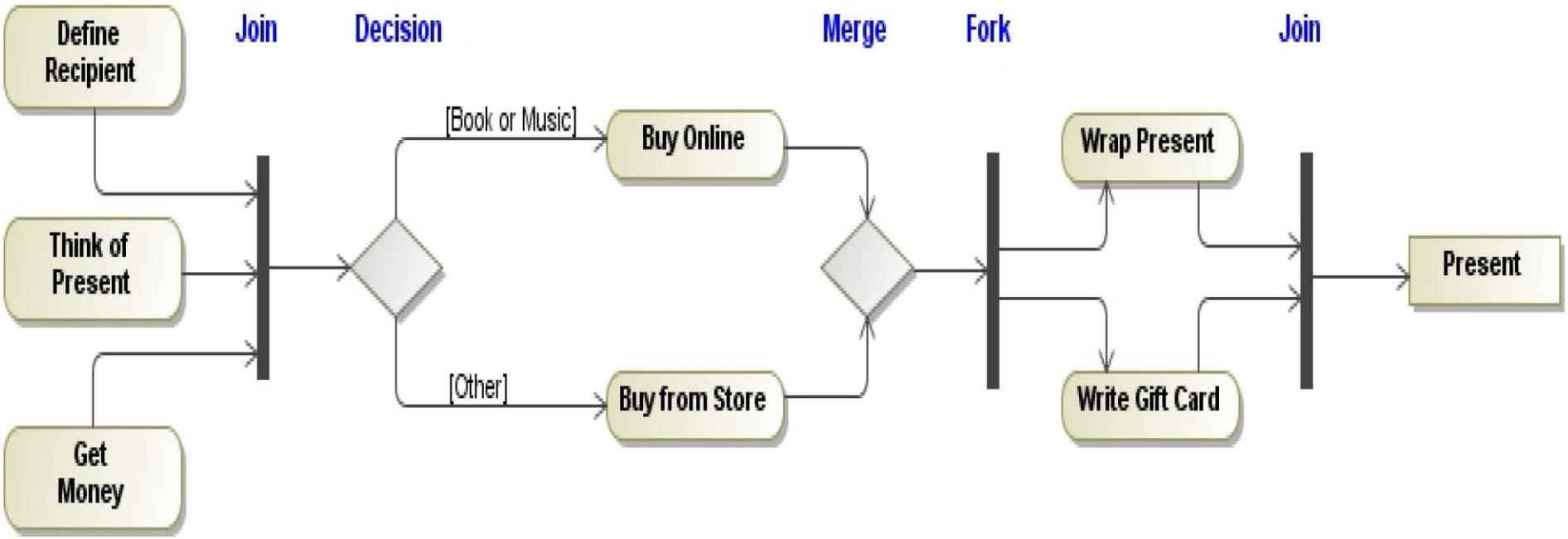


FIGURE 6.35

Example of block specialization.

Activity Diagrams



Sample activity diagram 2

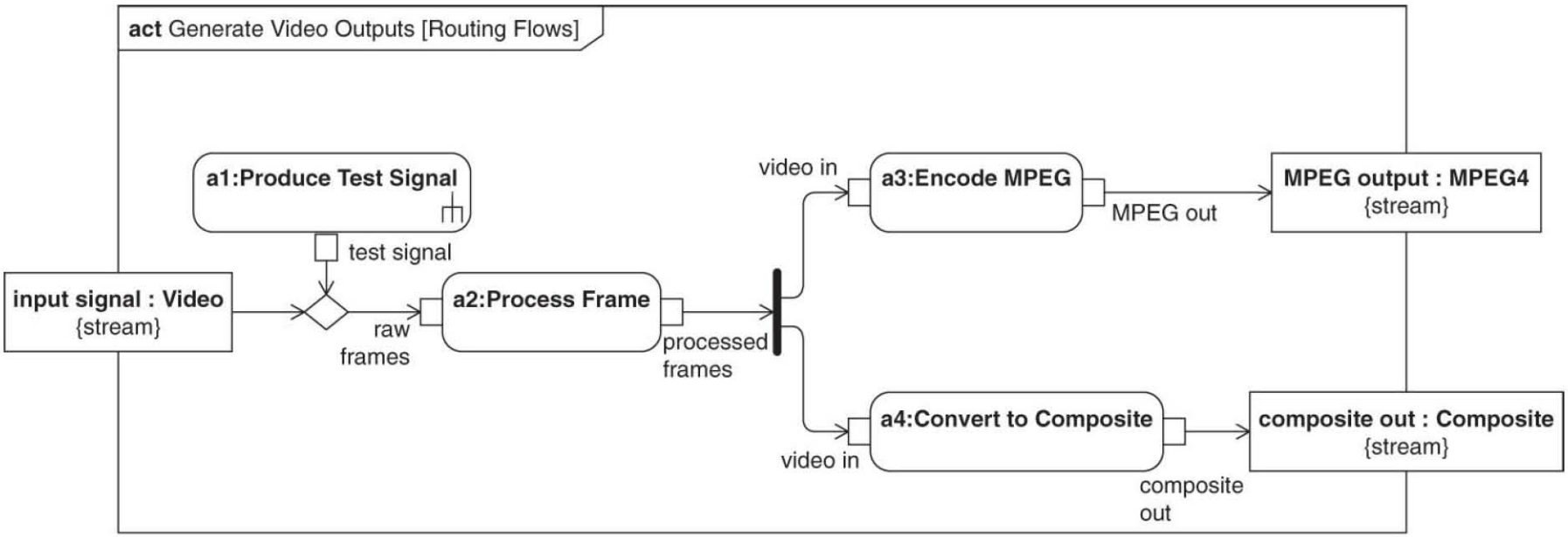
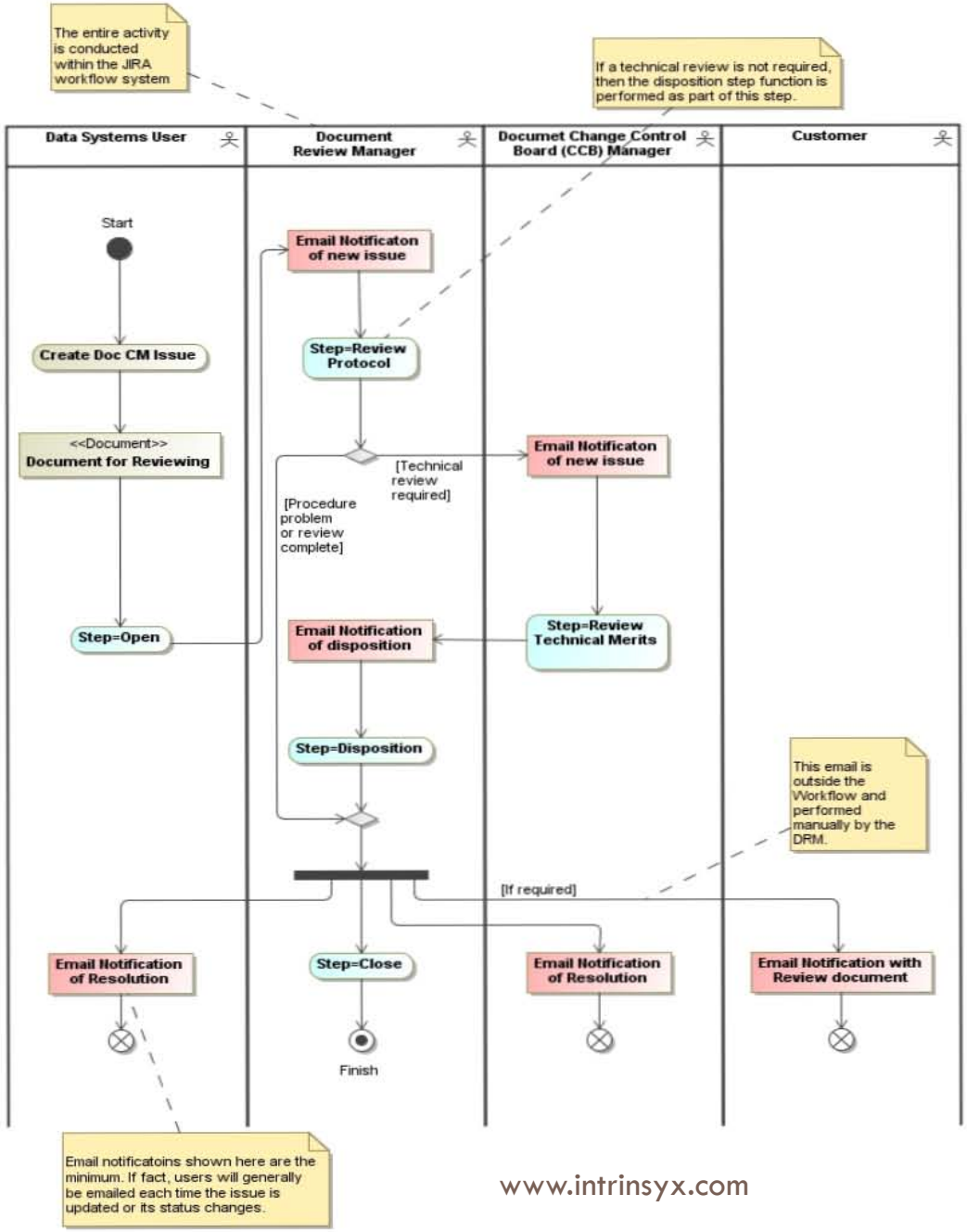


FIGURE 8.1

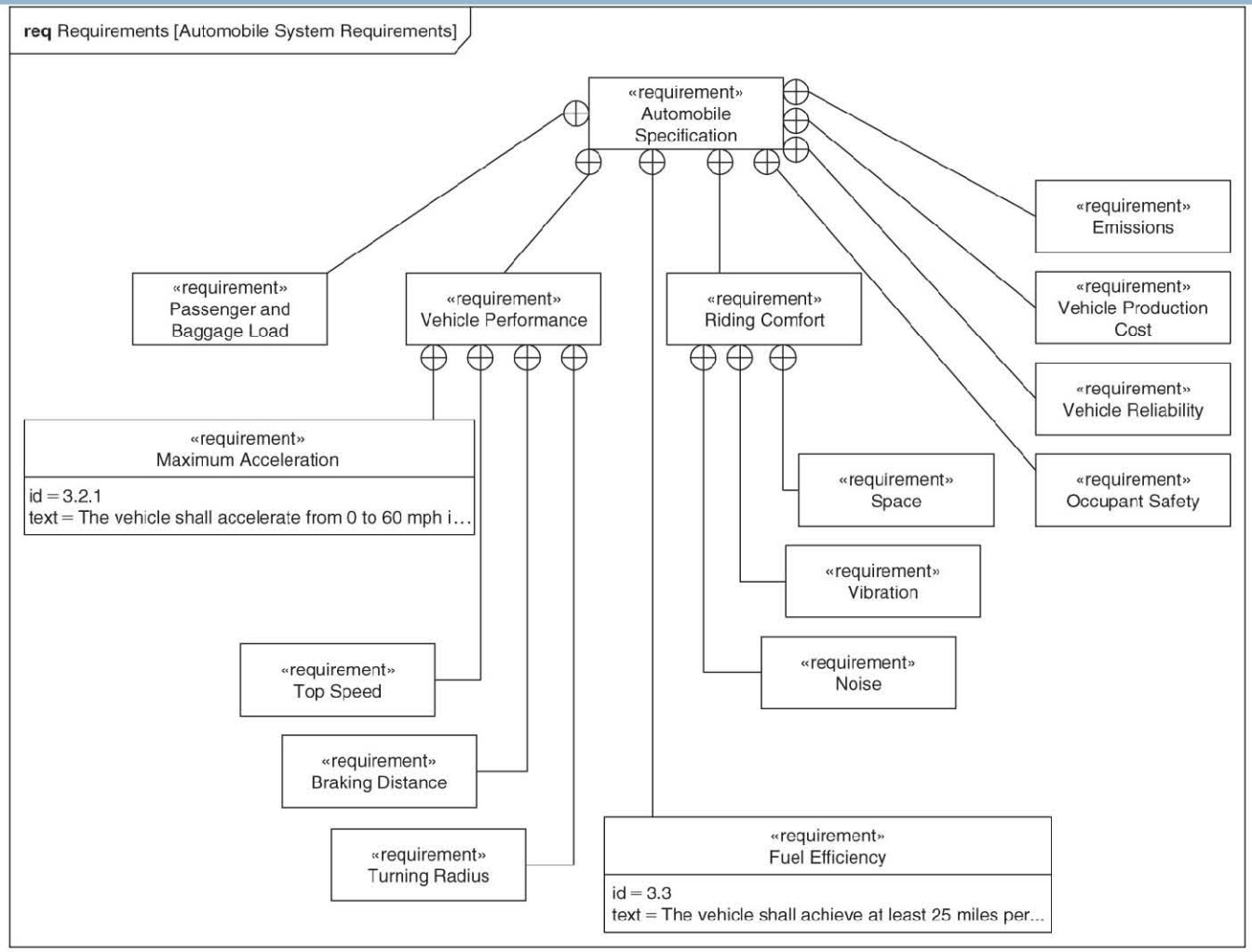
An example activity diagram.


The entire activity is conducted within the JIRA workflow system
If a technical review is not required, then the disposition step function is performed as part of this step.
Data Systems User
Document Review Manager
Documet Change Control Board (CCB) Manager
Customer
Start
Email Notificaton of new issue
Create Doc CM Issue
Step=Review Protocol
<<Document>>
Document for Reviewing
[Technical review required]
Email Notificaton of new issue
[Procedure problem or review complete]
Step=Open
Email Notification of disposition
Step=Review Technical Merits
Step=Disposition
This email is outside the Workflow and performed manually by the DRM.
[If required]
Email Notification of Resolution
Step=Close
Email Notification of Resolution
Email Notification with Review document
Finish
Email notificatoins shown here are the minimum. If fact, users will generally be emailed each time the issue is updated or its status changes.

Requirement diagram—Purpose

- Fully and unambiguously specify what the model is to do (functional requirements) and the context in which it is to operate (non-functional requirements).
- Provide concise and unambiguous information showing how requirements relate to each other.
- Provide concise and unambiguous information showing how requirements relate to other model elements—the project lifecycle.

Requirements diagram—Example



Requirements dependences

- «refine»
- «satisfy»
- «deriveReqt»
- «copy»
- «verify»
- «trace»

Refine, satisfy & dervieReqt

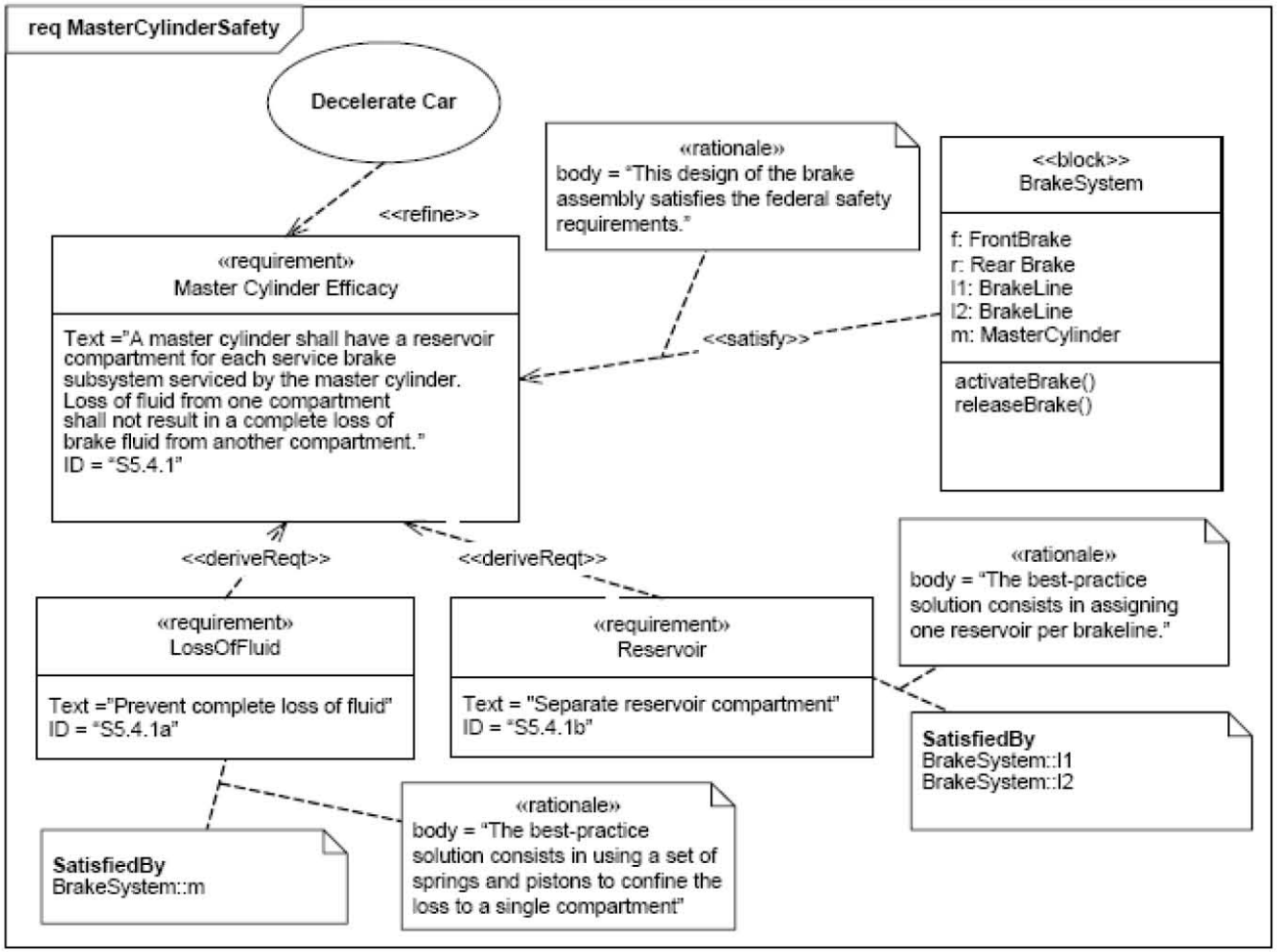


Figure 16.4 - Links between requirements and design

Source: OMG Specification

Copy

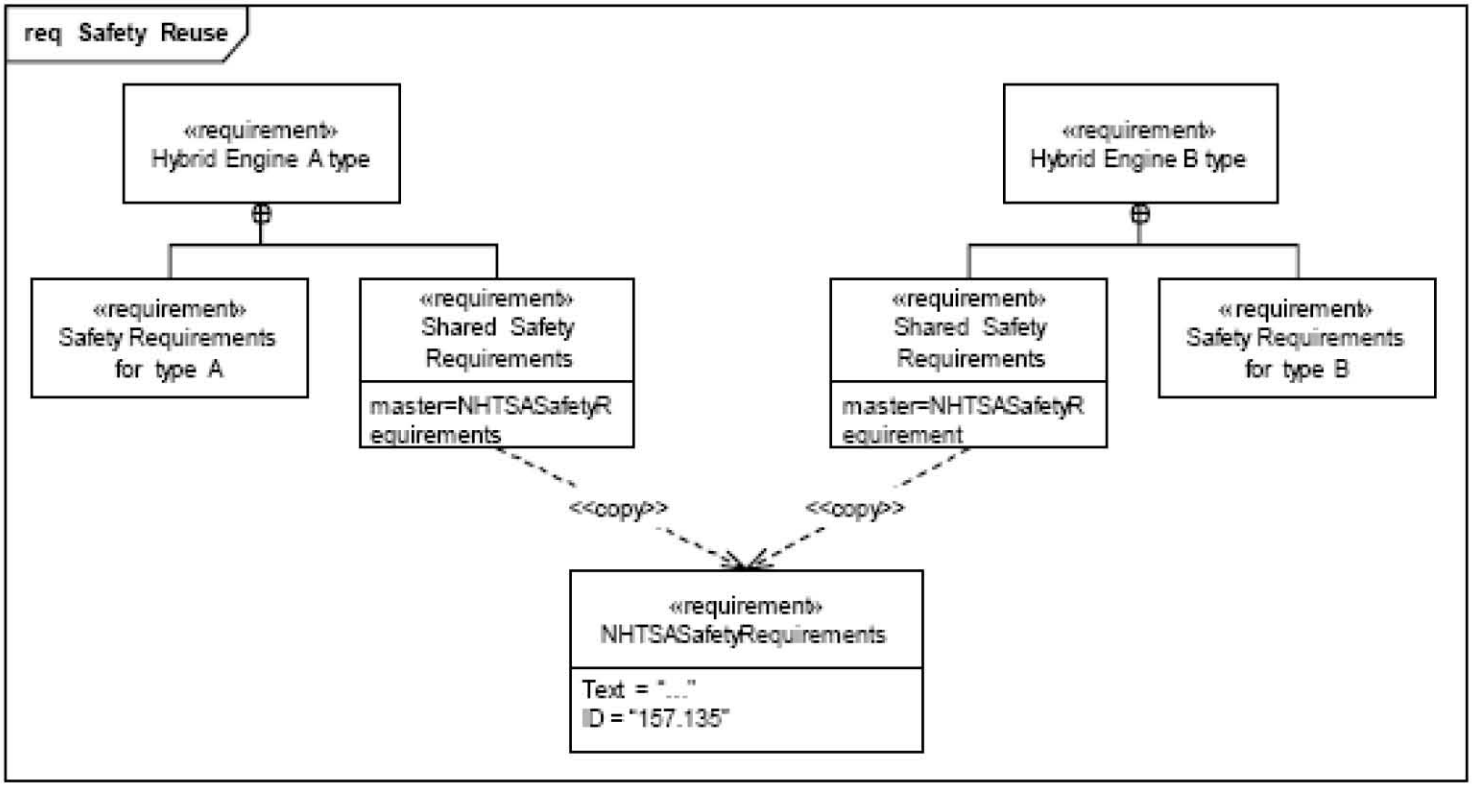
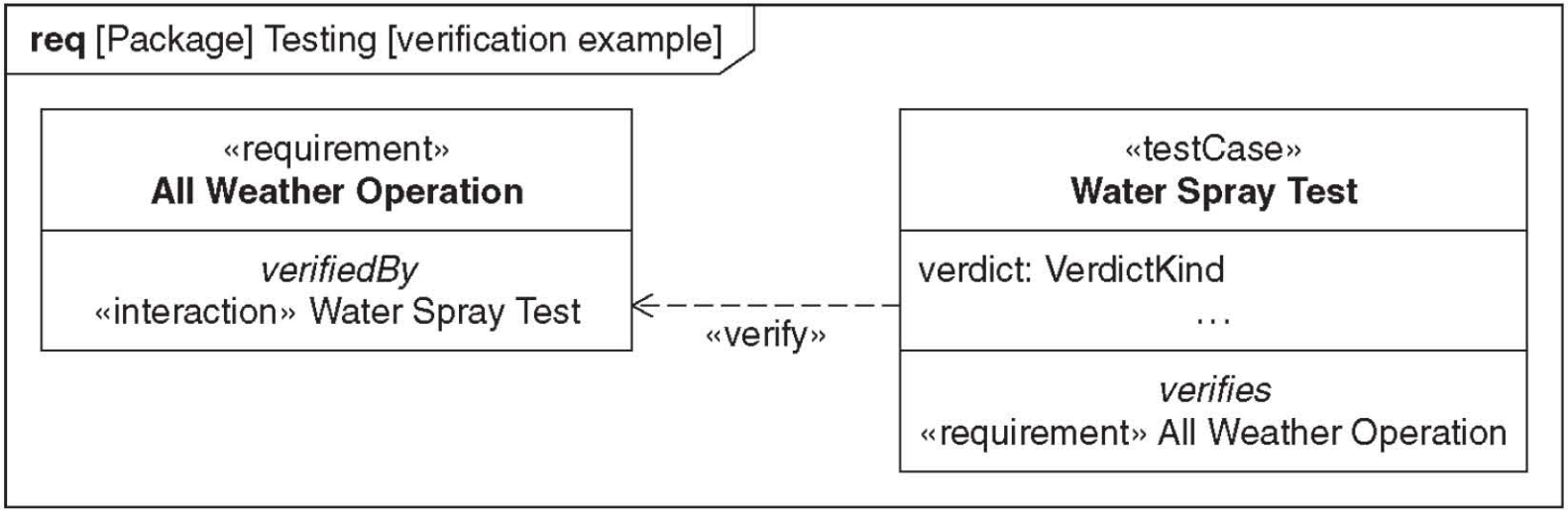


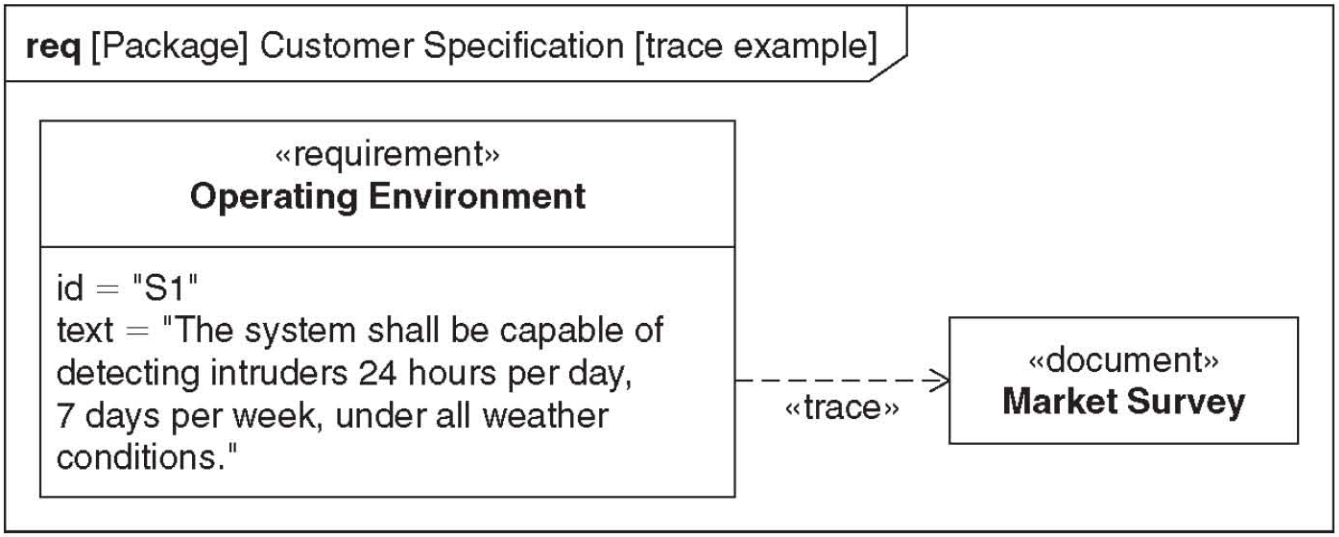
Figure 16.6 - Use of the copy dependency to facilitate reuse

Source: OMG Specification

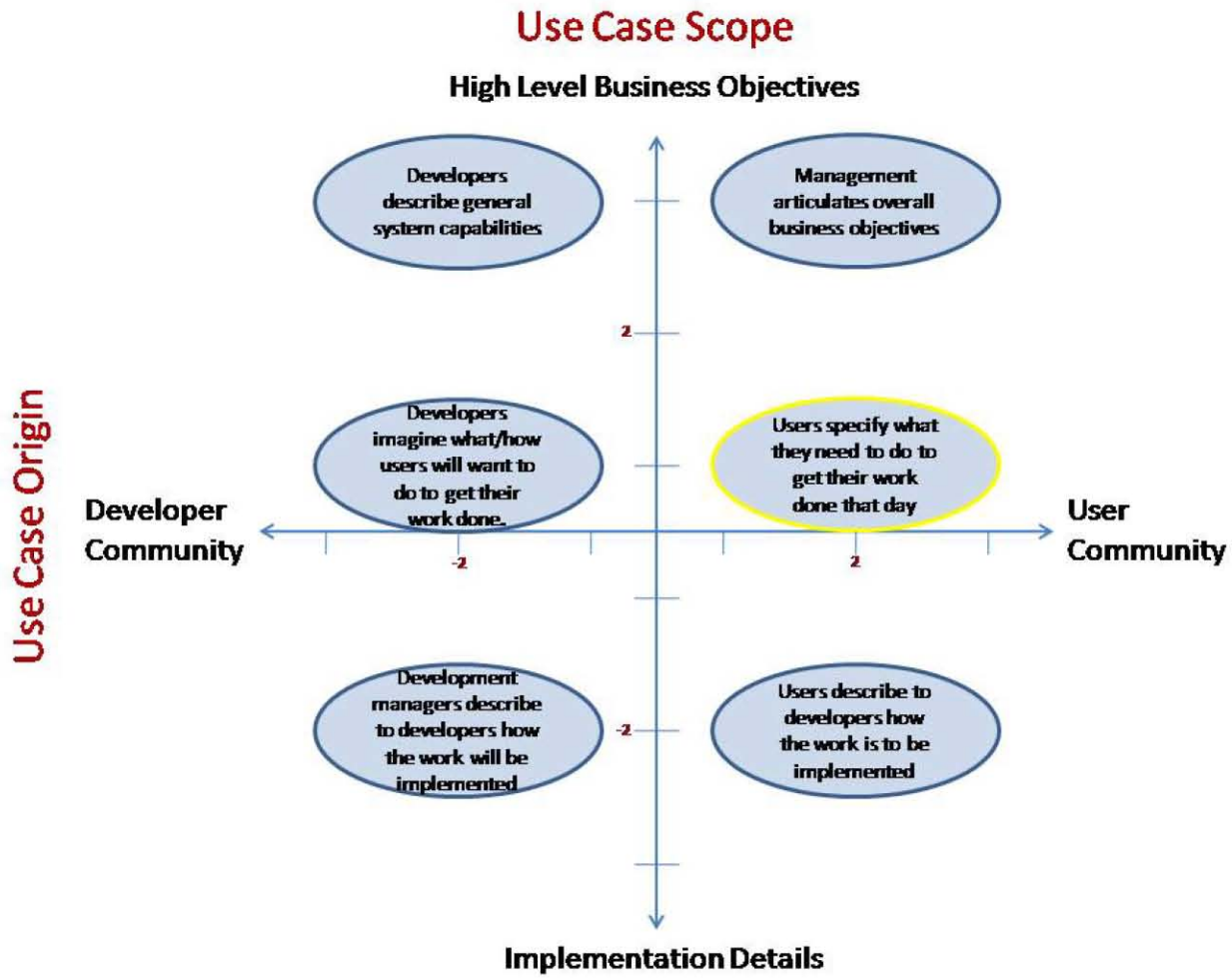
Verify



Trace



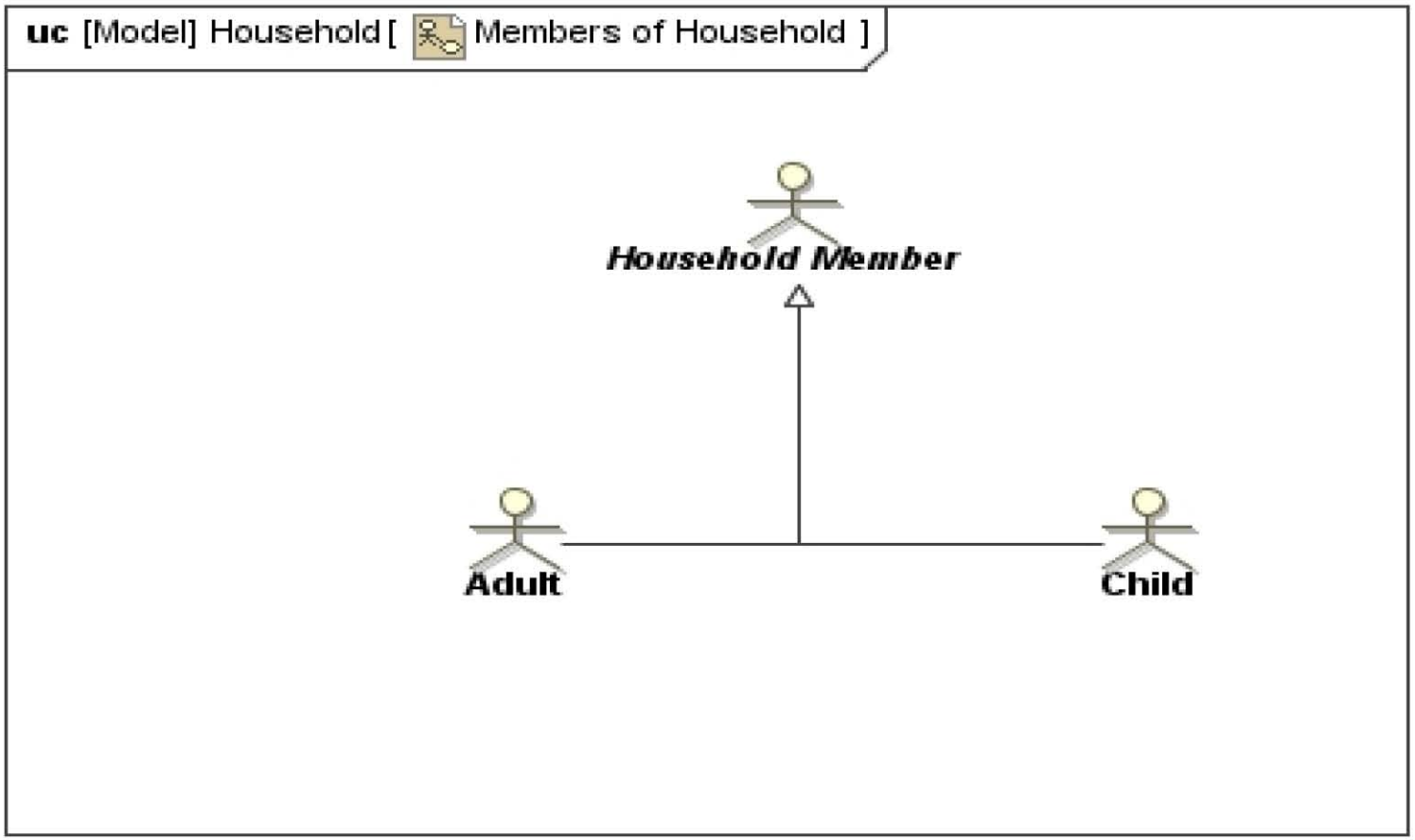
Use case scope



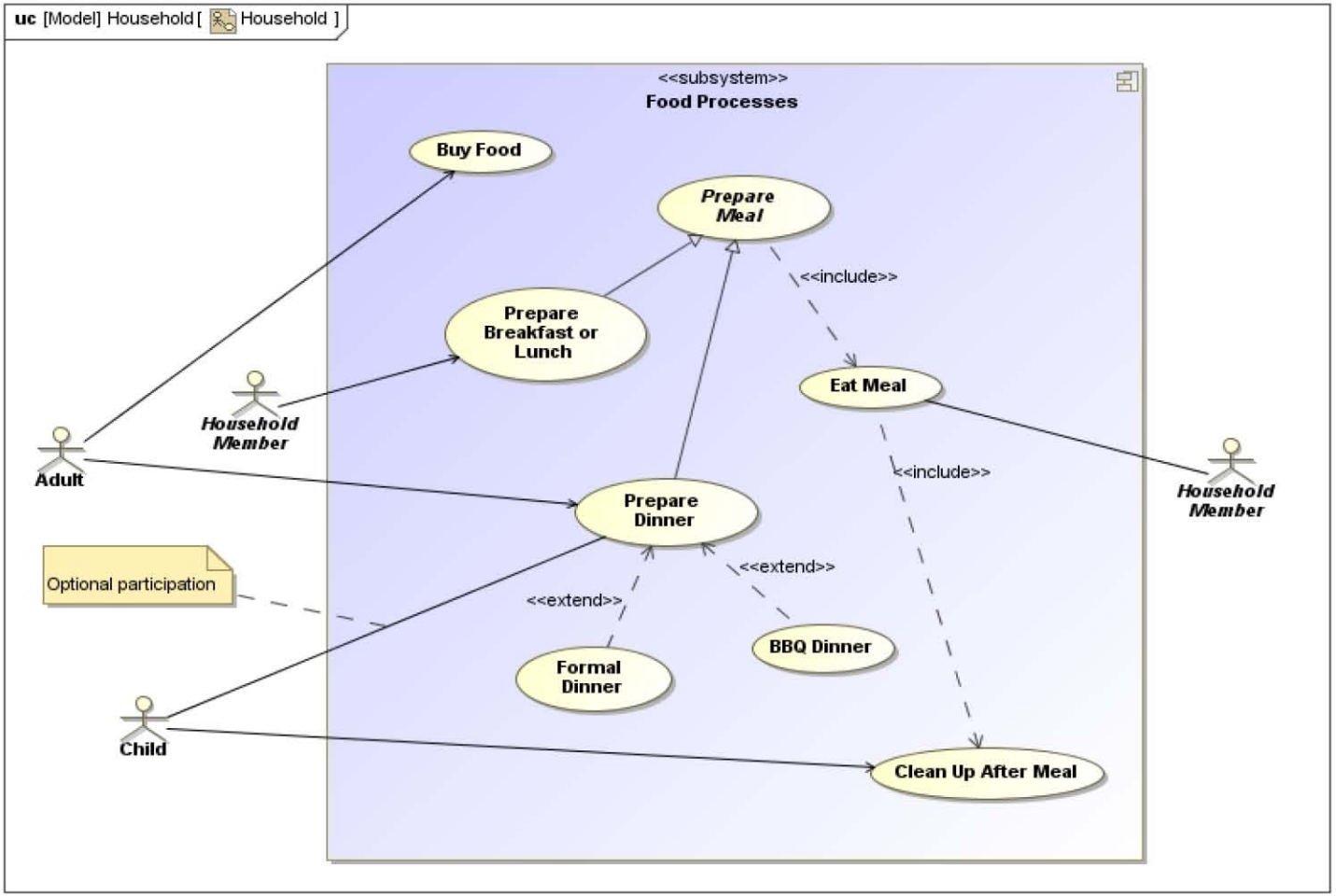
Use cases and actors

- Actors are external entities that interact with your system via use cases
- Actors can be people, computer systems or really any device that interacts with your system.
- It is high recommended that Actors be modeled before use cases.

Actors for a household model



Use cases for household food processes

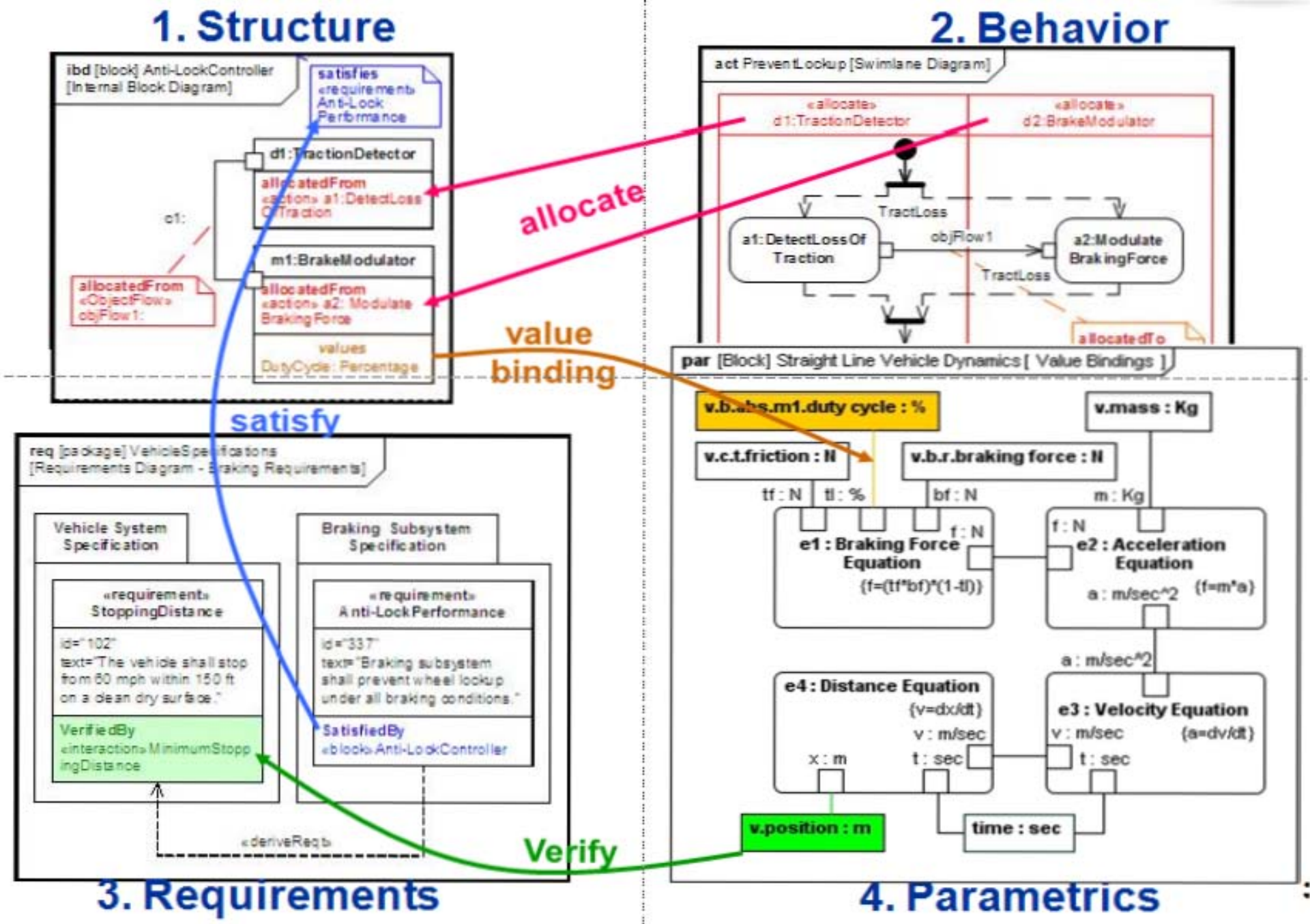


Part II, SysML

- SysML and project management
- Modeling in the context of SysML
- SysML structure and semantics
- **Summary**

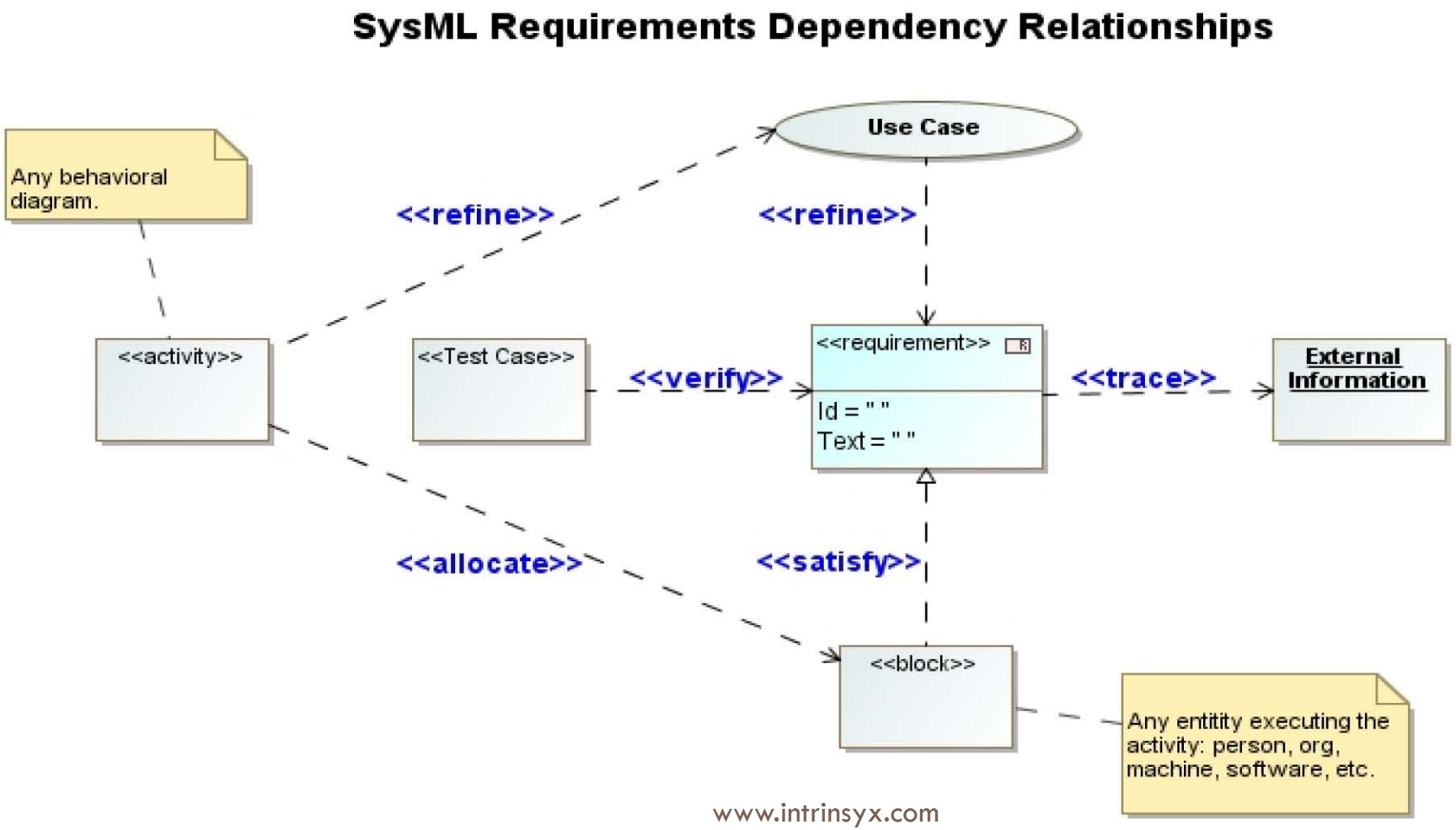
Key SysML features & capabilities

- Open standard
- Works equally well with structural and behavioral artifacts
- Treats requirements as first-class modeling elements
- Future plans include simulations
- Formal mechanism for extending SysML semantics
- Formal set of semantically consistent graphical elements.



Source: OMG Specification

SysML requirements relationships



Conclusion

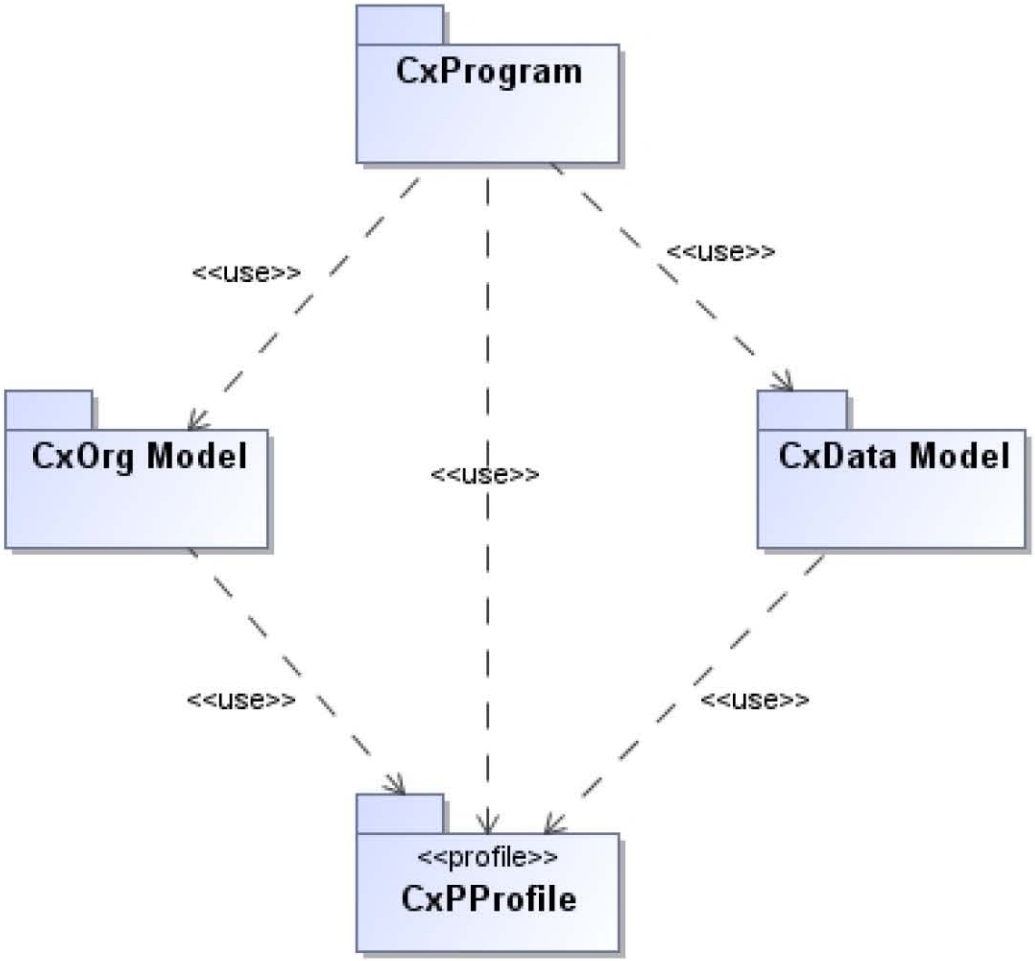
- **Why you don't want to model**
  - Modeling is hard
  - Modeling tools are difficult
  - Modeling will likely require cultural changes
- **Why you do want to model**
  - It increases the rate of communications
  - It increases the precision of communications
  - It reduces tacit information
  - It promotes a common understanding of your project
- **Benefit:** Stakeholders having a common understanding of how the project is organized and its objectives will work together more effectively and make better decisions.

Demonstration

An ad hoc initial attempt at modeling the NASA Constellation program, manned moon/Mars mission.

The notions and ideas expressed in these slides represent my own interpretations and have not been vetted or sanctioned by NASA in any way. They are presented solely for educational purposes regarding how large engineering projects might be rendered in a SysML model.

Model Organization



Constellation Program Model

Table of Contents

Structural Artifacts

Organization Information

- NASA Organization
- Constellation Organization
- SE&I
- PP &C

Data Taxonomy

- High Level Structure
- Operational Data

Information Systems

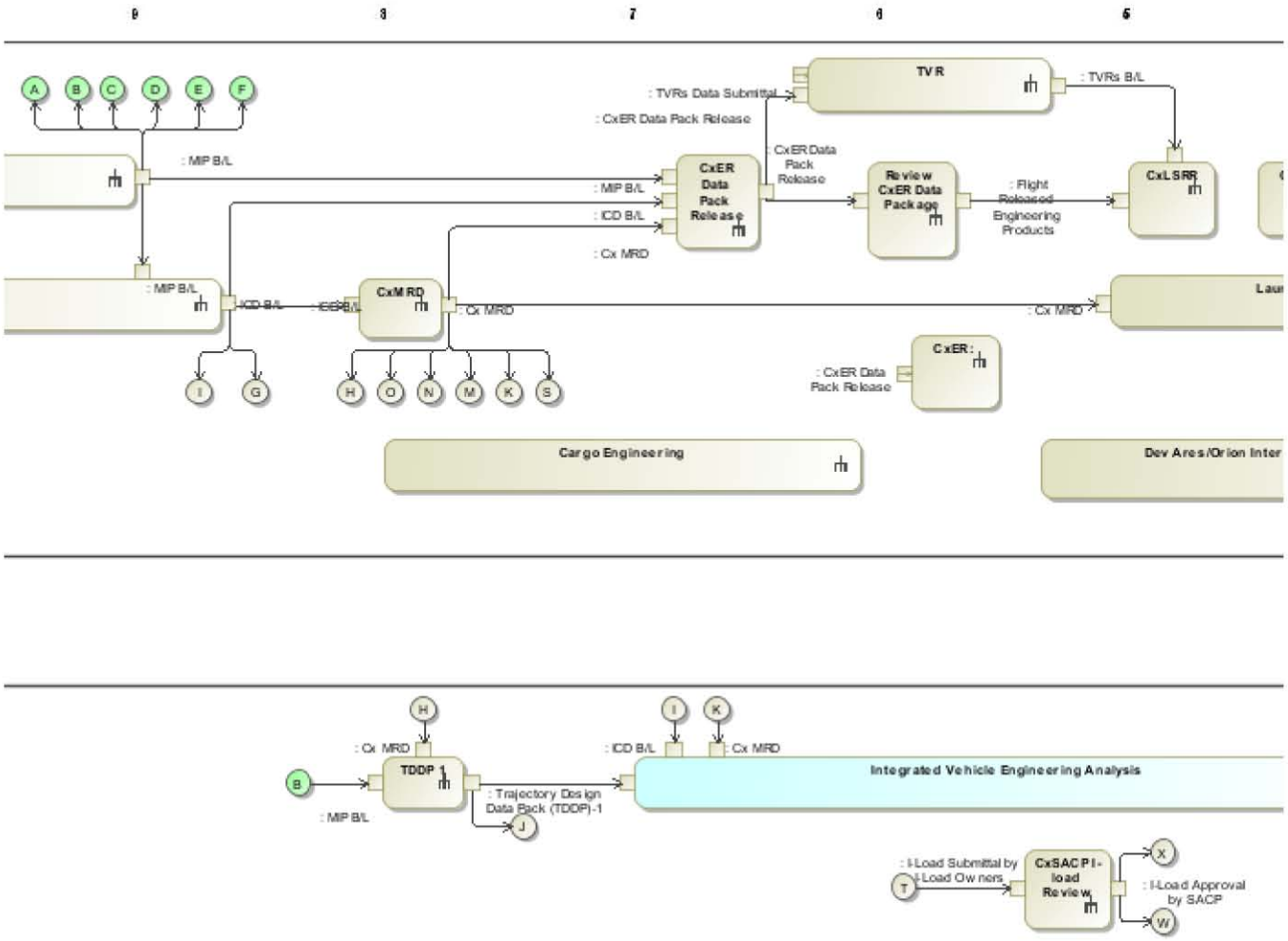
- Tools
- Tool Org and Functions

Behaviorial Artifacts

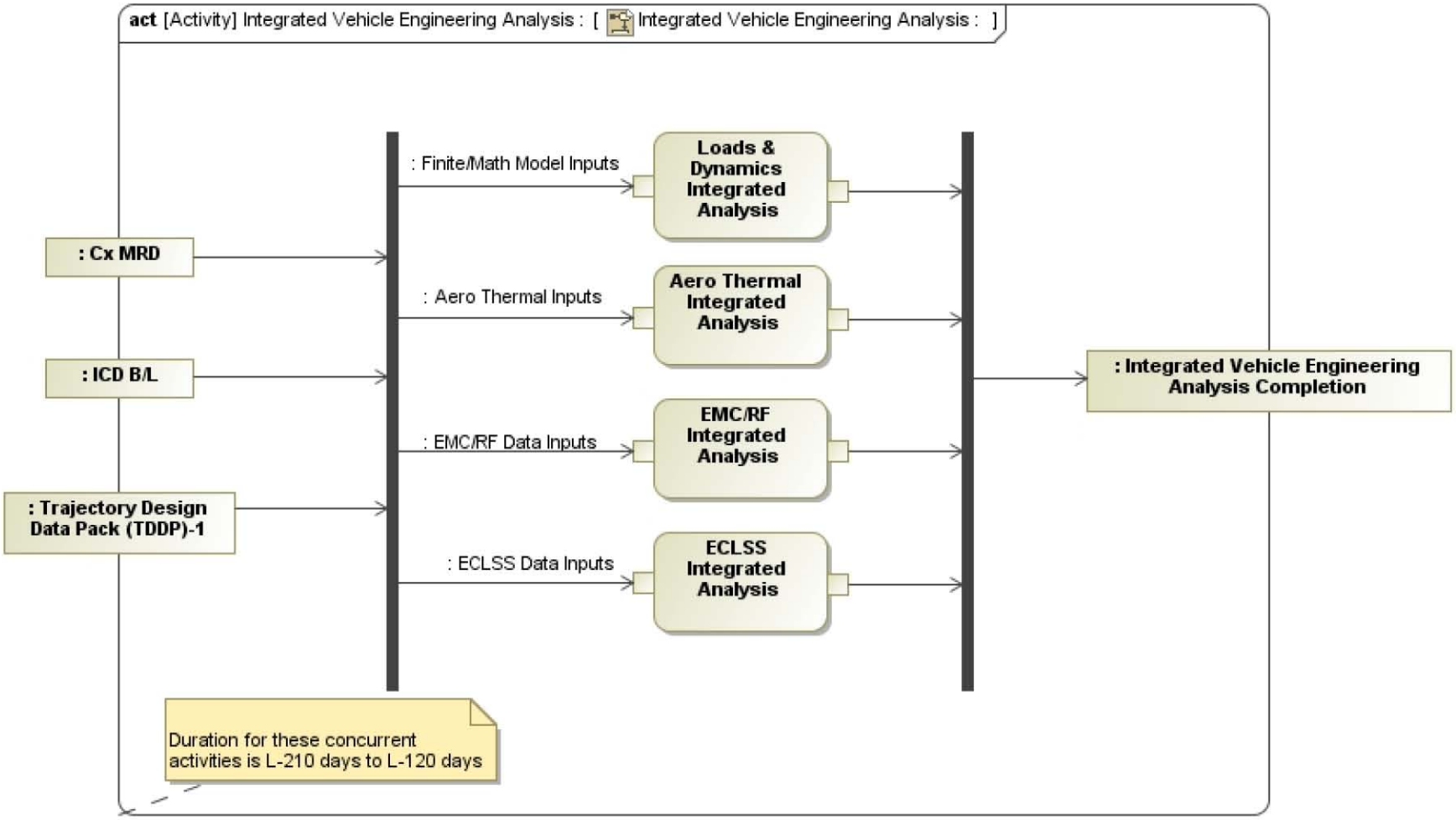
Virtual Missions and Operations

- GMIP Category Temporal Decomposition
- GMIP Category Information Decomposition
  - Design Category Decomposition
- GMIP
  - EVA H/W Processing & Crew Training
  - Integrated Vehicle Engineering Analysis :
  - MPPF
  - Offline Processing
    - CM/SM Stack Integration
  - VAB Integrated Operations

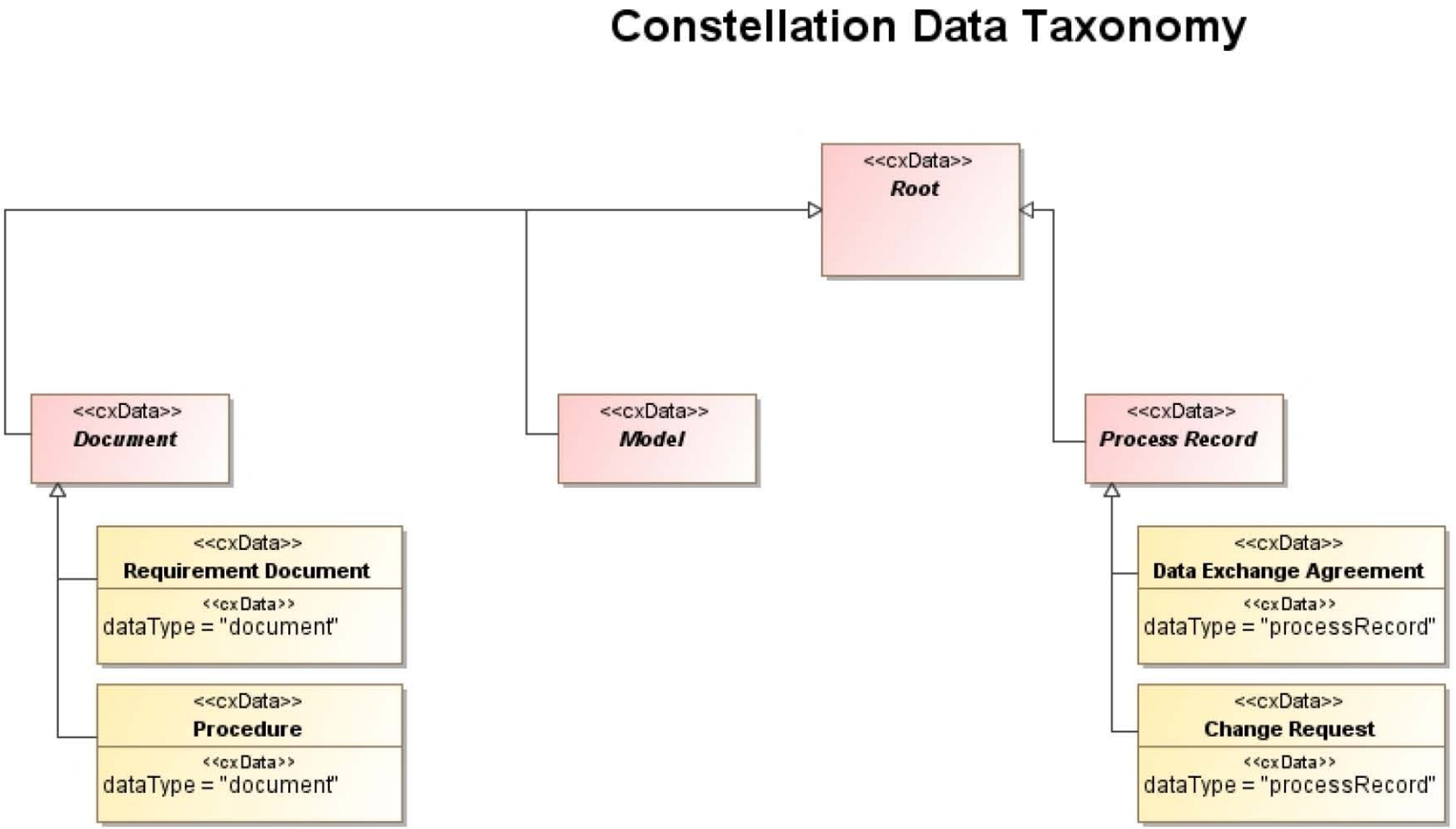
Generic Mission Template



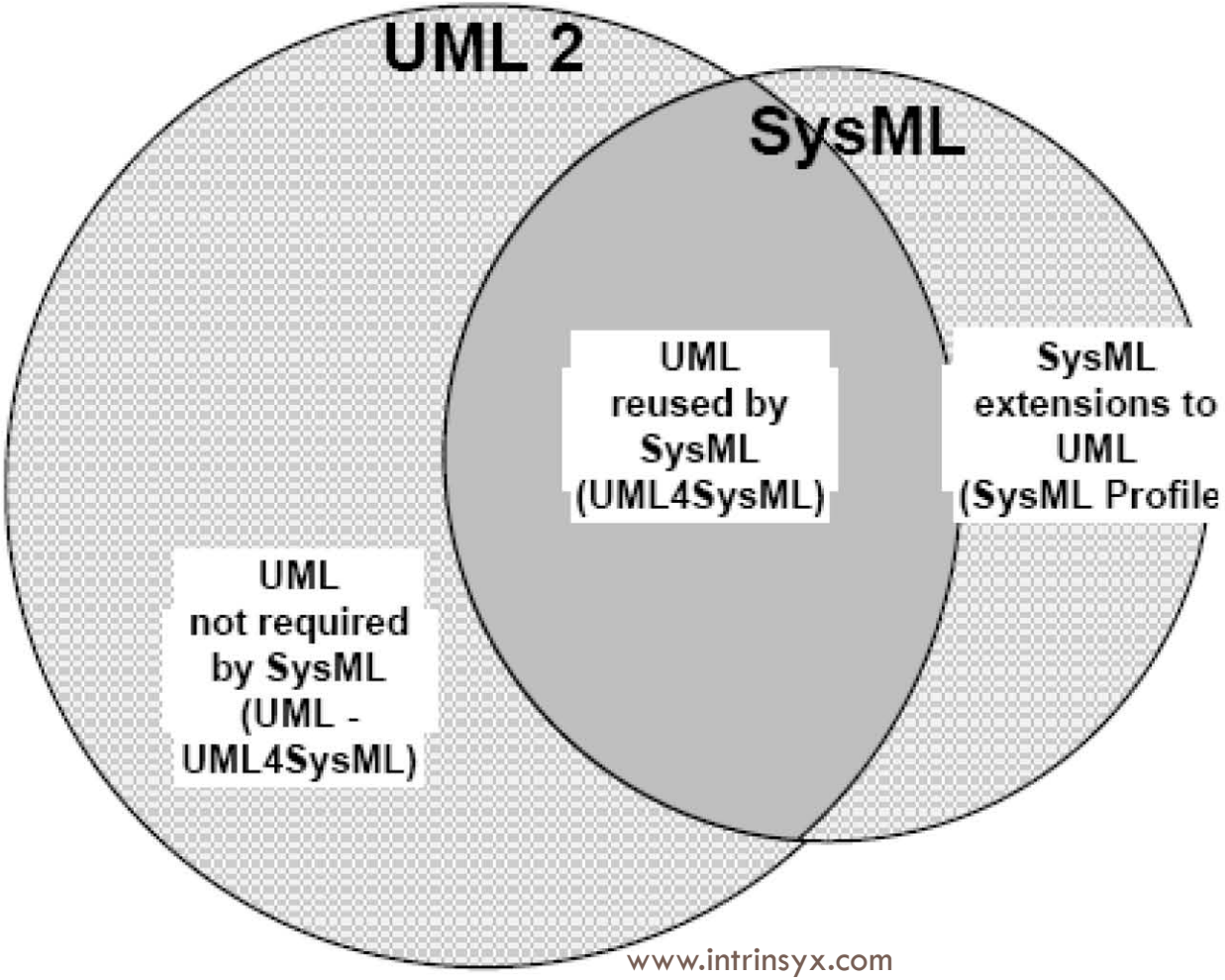
Integrated Vehicle Engineering Analysis



Constellation Data Taxonomy



Relationship between UML and SysML



Source: OMG Specification

UML/SysML metamodel

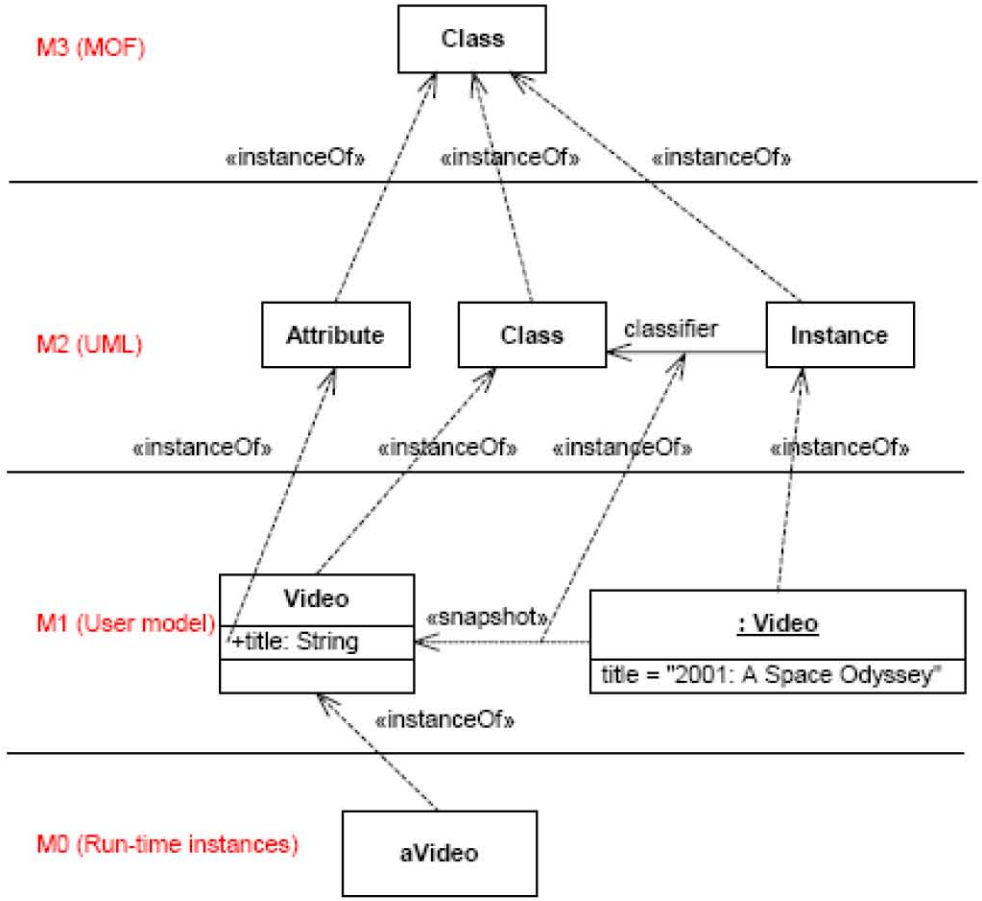


Figure 7.8 - An example of the four-layer metamodel hierarchy

Source: OMG Specification

Copyright Information