NASA/Army Systems and Software
Engineering Forum
May 11 & 12, 2010
University of Alabama, Huntsville

Theodore Kahn
Ian Sturken
NASA Ames Research Center
Moffett Field, CA

theodore.e.kahn@nasa.gov
ian.b.sturken@nasa.gov

Applying Enterprise Architecture to Model Based Systems Engineering

A pragmatic approach to modeling large systems

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 information and enforces **formal semantics** of its expression…
  - Promotes a more **common interpretation and consistency** of project information, even under very dynamic conditions.

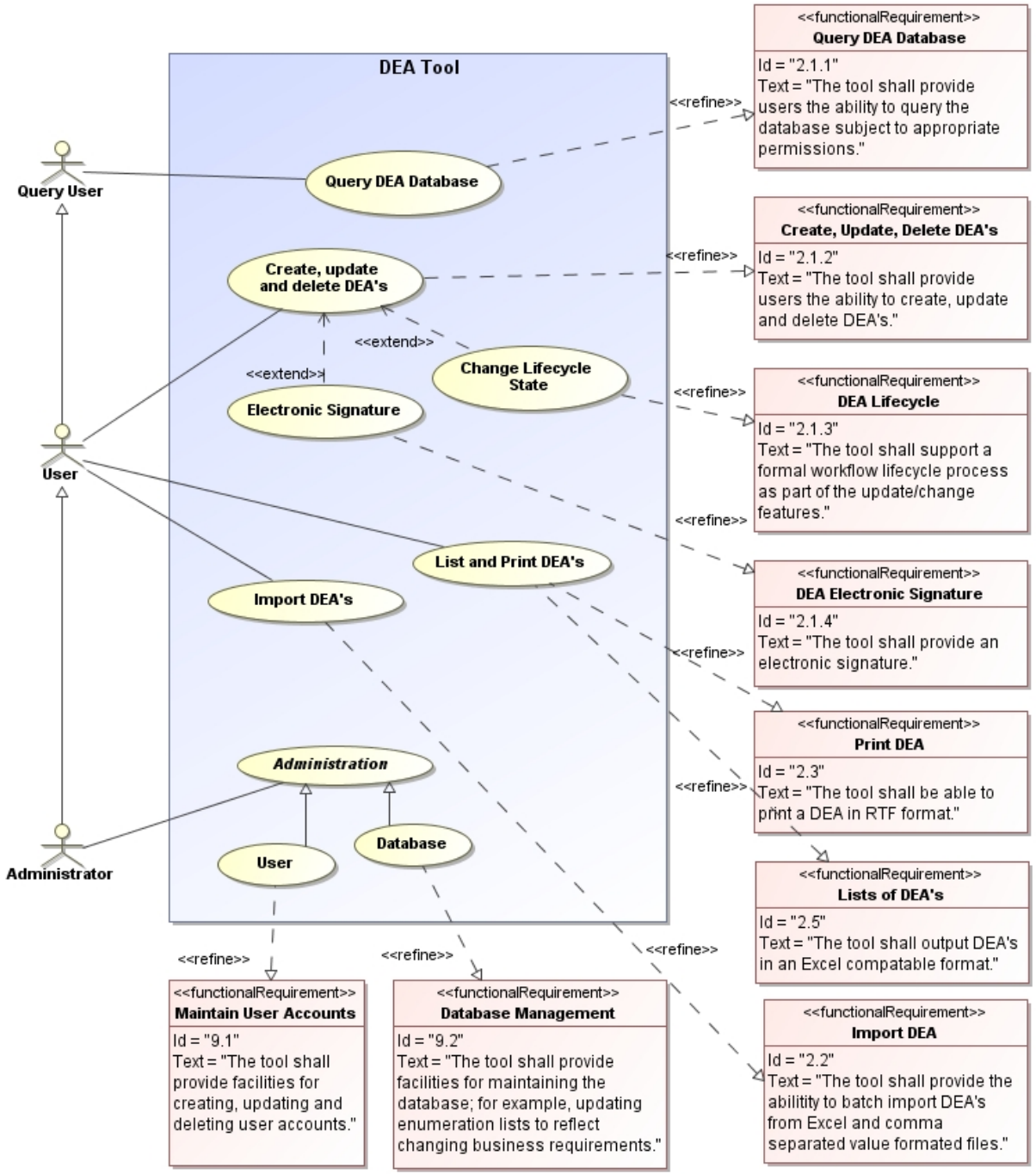
Presentation objectives

- Provide conceptual understandings of **modeling** and **enterprise architecture:** as applied to **systems engineering** and **project management** of large efforts.
- Share our experiences in applying UPDM and SysML to components of NASA's Constellation Program.
- Enable you to make informed decisions regarding the applicability of these approaches to your situations.

Presentation overview

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

DEA Tool Functional Use Cases



Conclusion

- **Why you <u>don't</u> want to model**
  - Modeling is hard
  - Modeling tools are difficult
  - Modeling will likely require cultural changes
- **Why you <u>do</u> want to model**
  - It increases the <u>rate</u> of communications
  - It increases the <u>precision</u> of communications
  - It reduces <u>tacit</u> information
  - It promotes a <u>common understanding</u> 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 and Enterprise Architecture

- Relationship between systems engineering and project management
- EA and Modeling: definitions in the context of UPDM/SysML
- The relationship between EA, UPDM and SysML

Relationship Between SE and PM

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

Similarities between PM & SE

- Lots of <u>behaviors</u> having complex relationships
- Lots of <u>entities</u> 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.

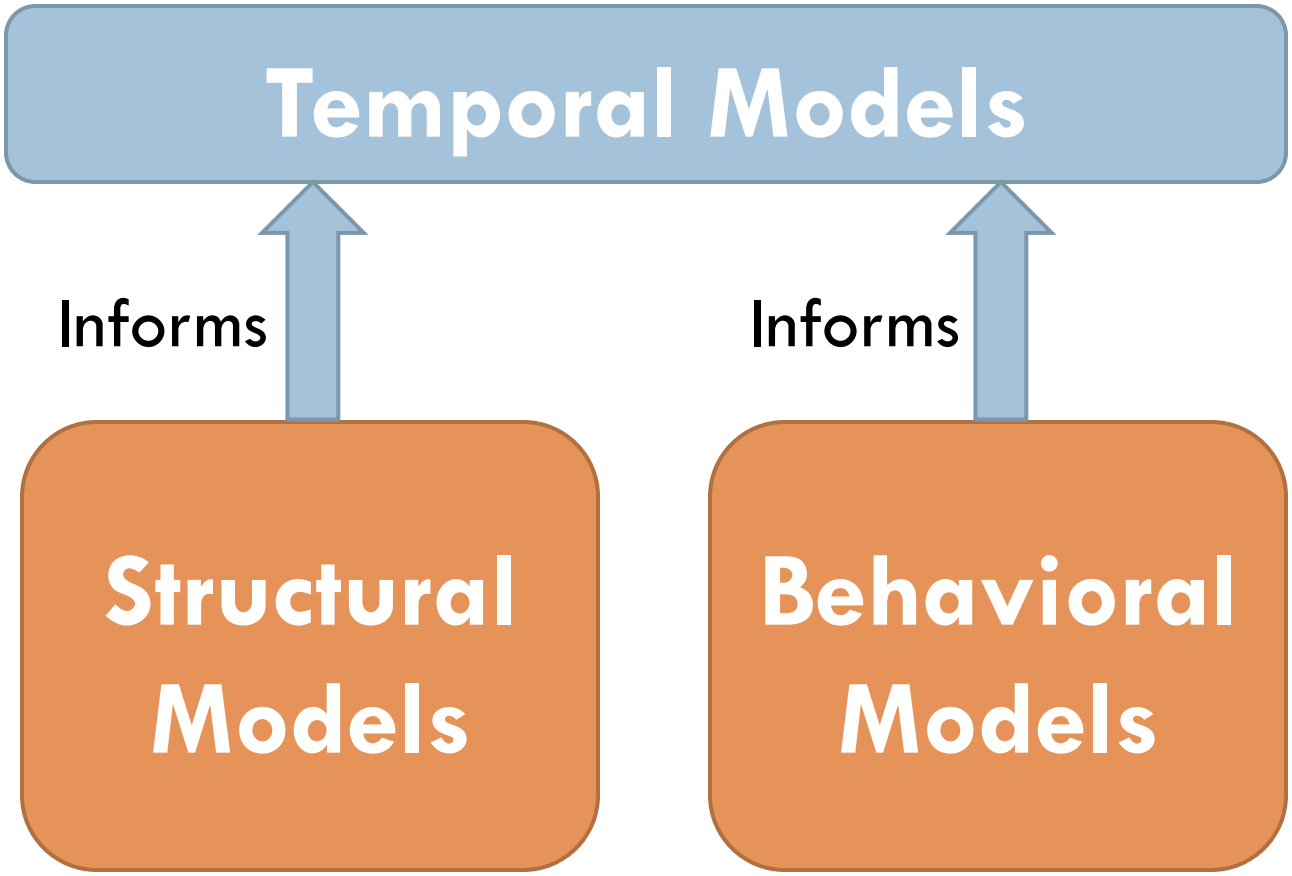
Source: A Practical Guide to SysML., copyright © 2008 by Elsevier, Inc. All rights reserved.

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.

EA and Modeling: concepts in the context of UPDM/SysML

- Looking at modeling from SysML'sperspective
- Ideas behind enterprise architectures

SysML modeling is about communications

- It's a <u>language</u> having syntax and structure…
- that uses a <u>medium</u>, primarily graphics…
- and has a <u>methodology</u>, 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

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. <u>There is nothing new here.</u>

- **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. <u>It is not workable.</u>

Ideas Behind Enterprise Architectures

To Be Done By Ian?

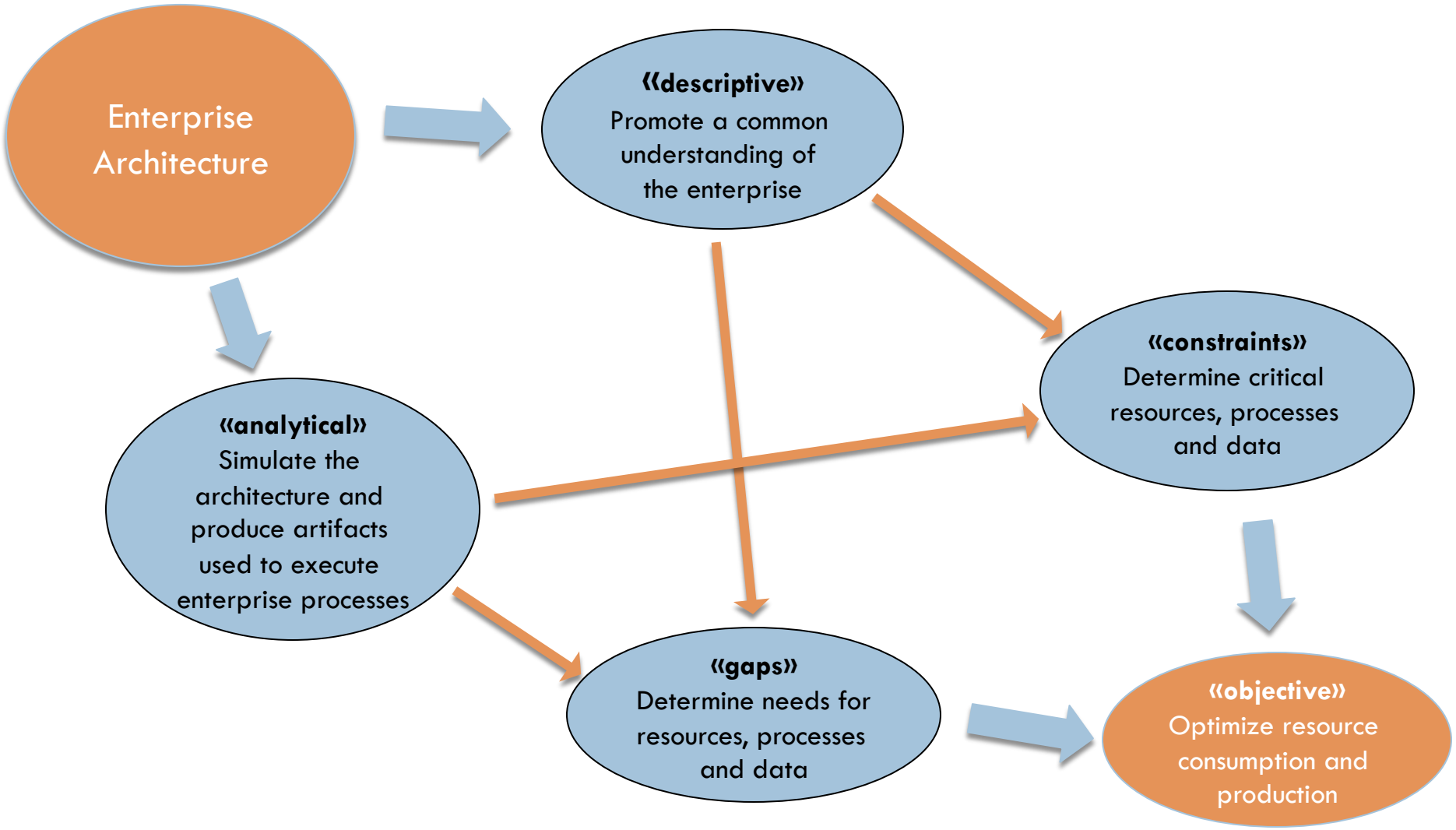
Enterprise Architecture

- An *enterprise architecture* describes the *organizations (and their people), processes, inputs, products* and *systems* that comprise an enterprise and the *relations* between each of these entities.
- It can be used to understand the "as-is" state of the enterprise and plan the "to-be" vision of where the enterprise should be.

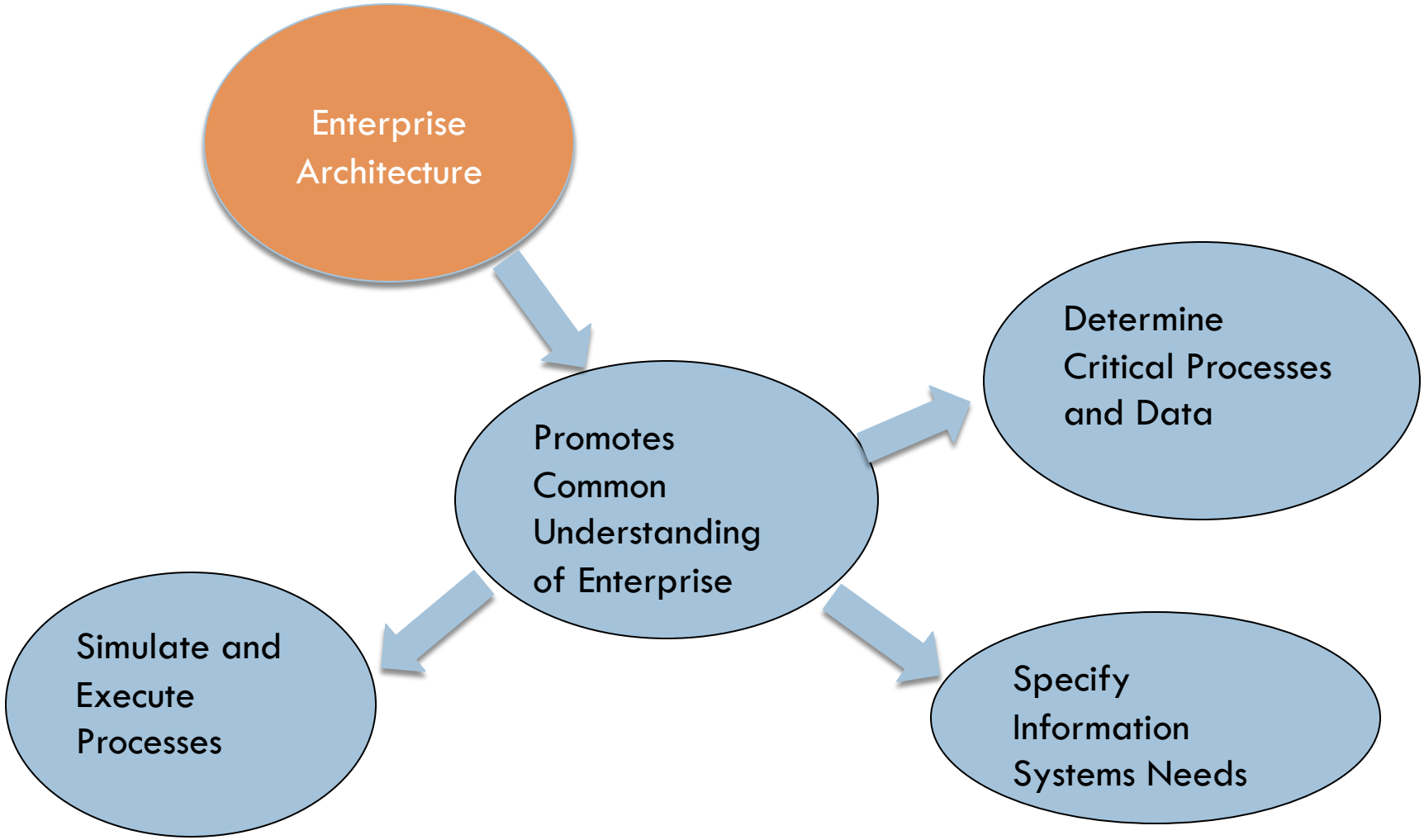
Enterprise Architecture Evolution

- Zachman (1987)
- Enterprise Architecture Planning (Spewak 1992)
- TOGAF (1995)
- FEA (1999)
- DoDAF 1.0(2003)
- DoDAF 1.5(2007)
- UPDM 1.0 (2008)
- DoDAF 2.0(2009?)

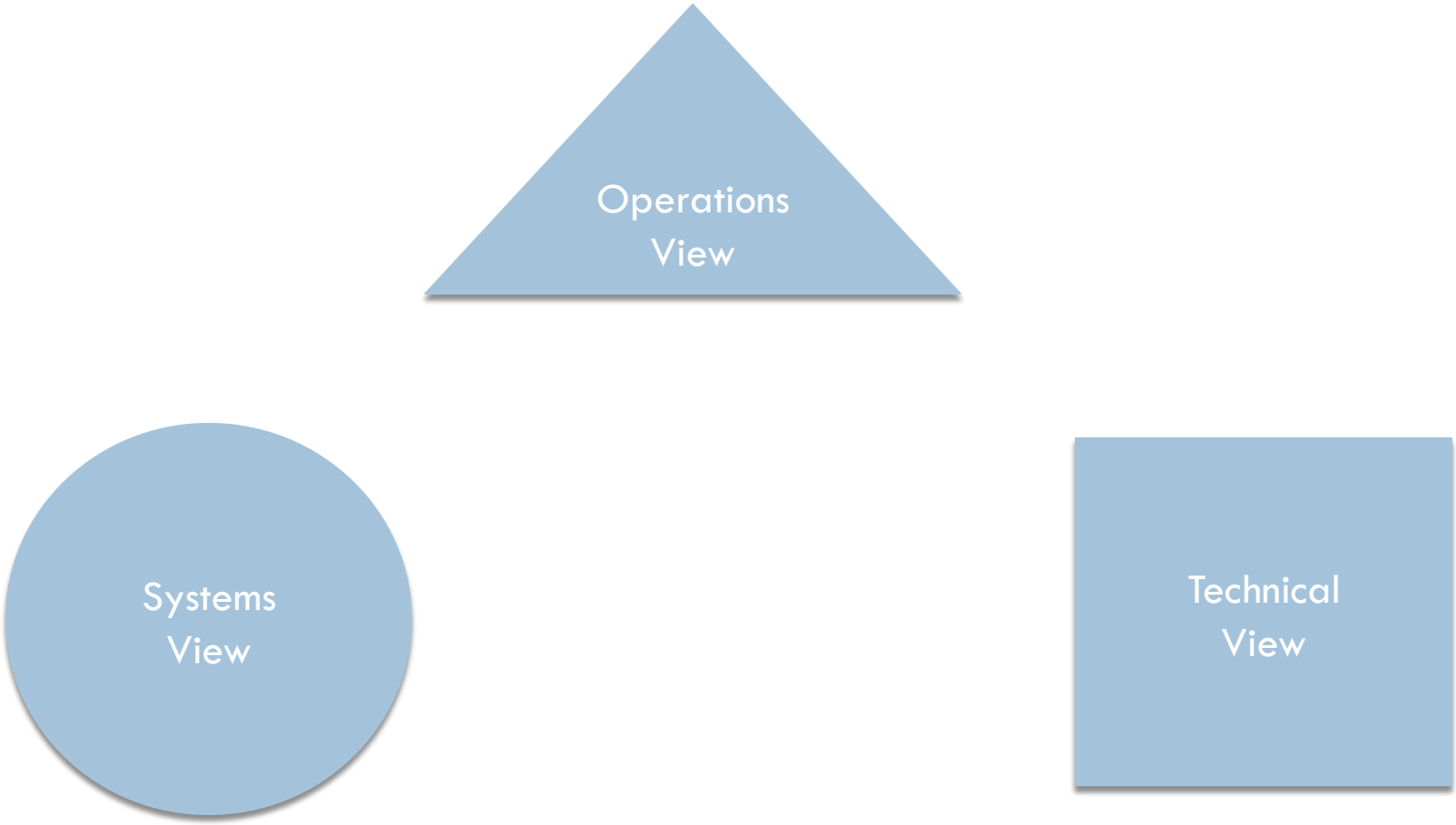
Benefits of Enterprise Architecture



Benefits of Enterprise Architecture

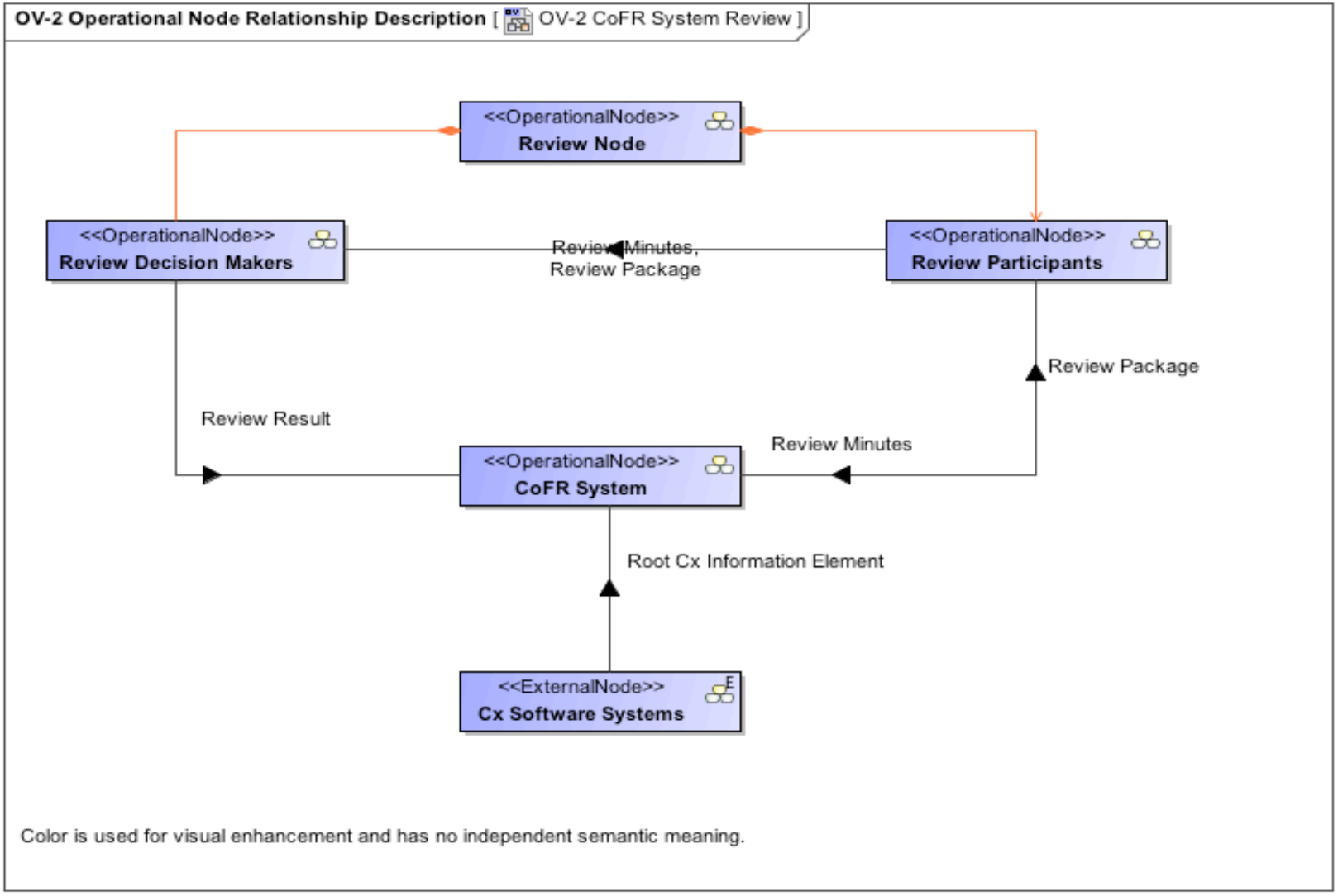


Catalog of DoDAF views



DoDAF Example of Models

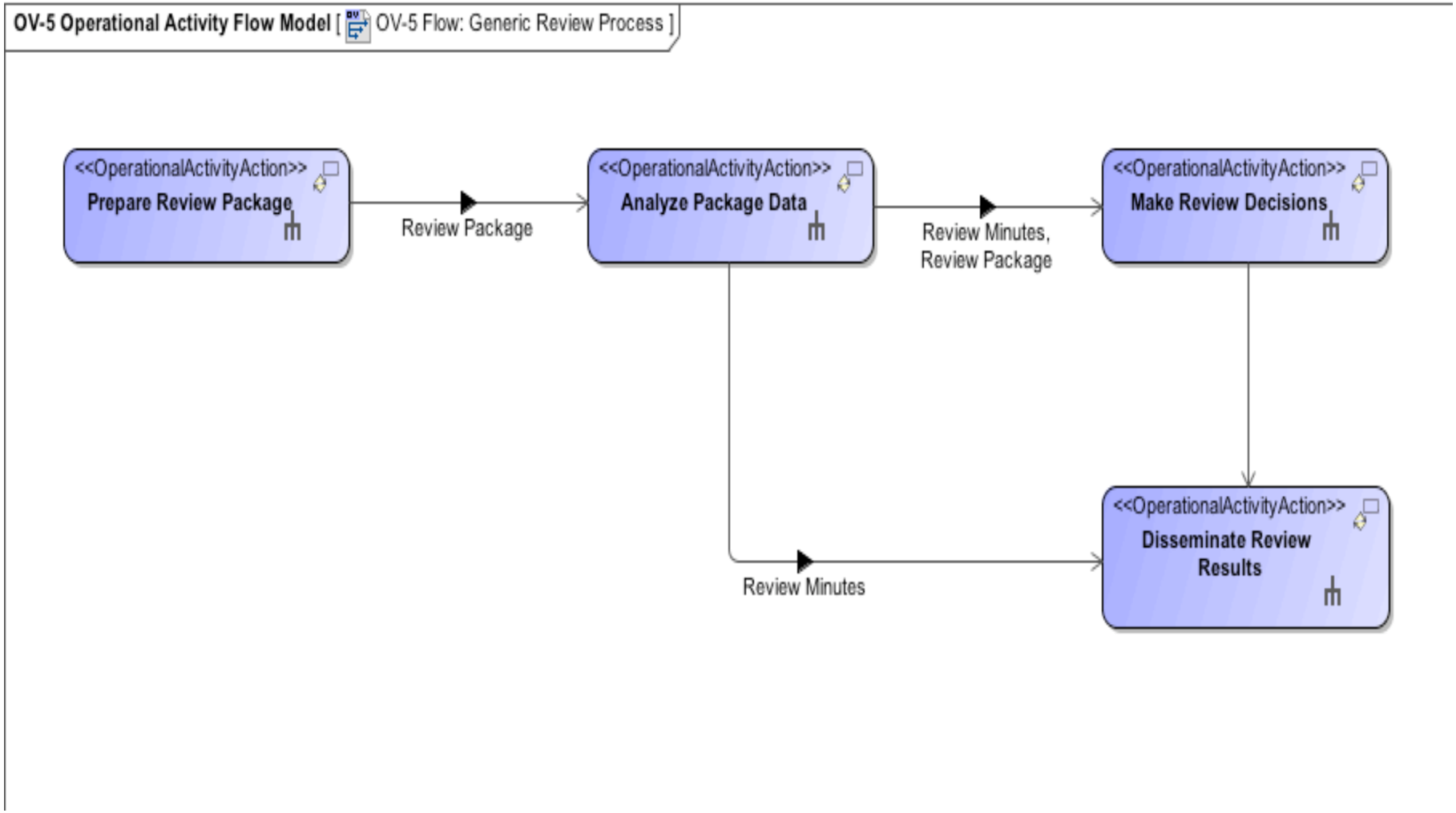
OV-2


OV-2 Operational Node Relationship Description [OV-2 CoFR System Review]
<<OperationalNode>>
Review Node
<<OperationalNode>>
Review Decision Makers
Review Minutes,
Review Package
<<OperationalNode>>
Review Participants
Review Package
Review Result
Review Minutes
<<OperationalNode>>
CoFR System
Root Cx Information Element
<<ExternalNode>>
Cx Software Systems
Color is used for visual enhancement and has no independent semantic meaning.

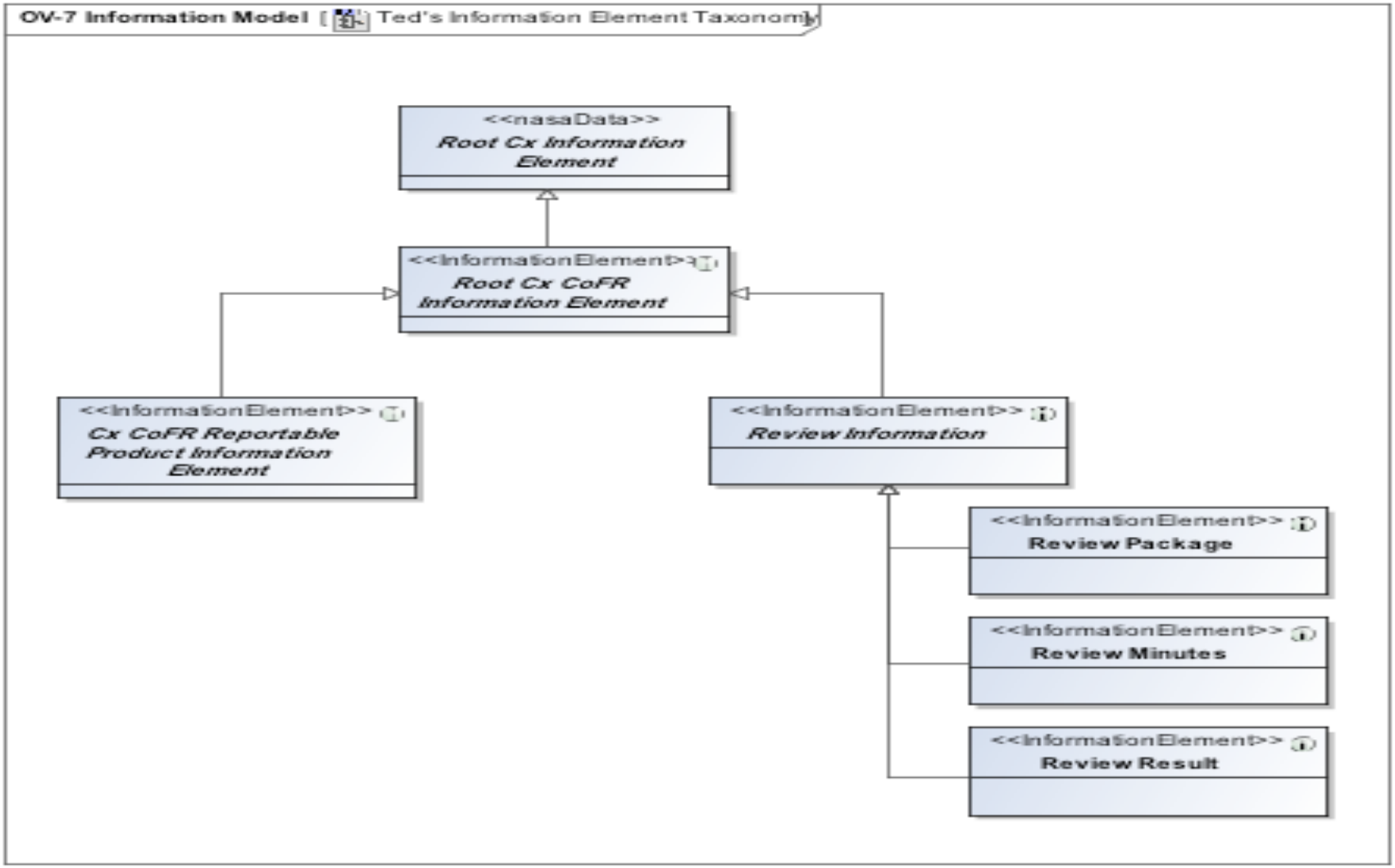
OV-3

#	Needline Ide... Number	Operational E... Identification...	Operational Exchange Type	Operational Exchange Item Name	Sending Node	Receiving Node	Sending Operational Activity	Receiving Operational Activity
1			Information Exchange	Review Package	CoFR System	Review Participants	Prepare Review Package	Analyze Package Data
2			Information Exchange	Review Minutes Review Package	Review Participants	Review Decision M...	Analyze Package Data	Make Review Decisions
3			Information Exchange	Review Minutes	Review Participants	CoFR System	Analyze Package Data	Disseminate Review Results

OV-5


OV-5 Operational Activity Flow Model [OV-5 Flow: Generic Review Process]
<<OperationalActivityAction>>
Prepare Review Package
Review Package
<<OperationalActivityAction>>
Analyze Package Data
Review Minutes,
Review Package
<<OperationalActivityAction>>
Make Review Decisions
Review Minutes
<<OperationalActivityAction>>
Disseminate Review
Results

OV-7


OV-7 Information Model [Ted's Information Element Taxonomy]
<<nasaData>>
Root Cx Information Element
<<InformationElement>>
Root Cx CoFR Information Element
<<InformationElement>>
Cx CoFR Reportable Product Information Element
<<InformationElement>>
Review Information
<<InformationElement>>
Review Package
<<InformationElement>>
Review Minutes
<<InformationElement>>
Review Result

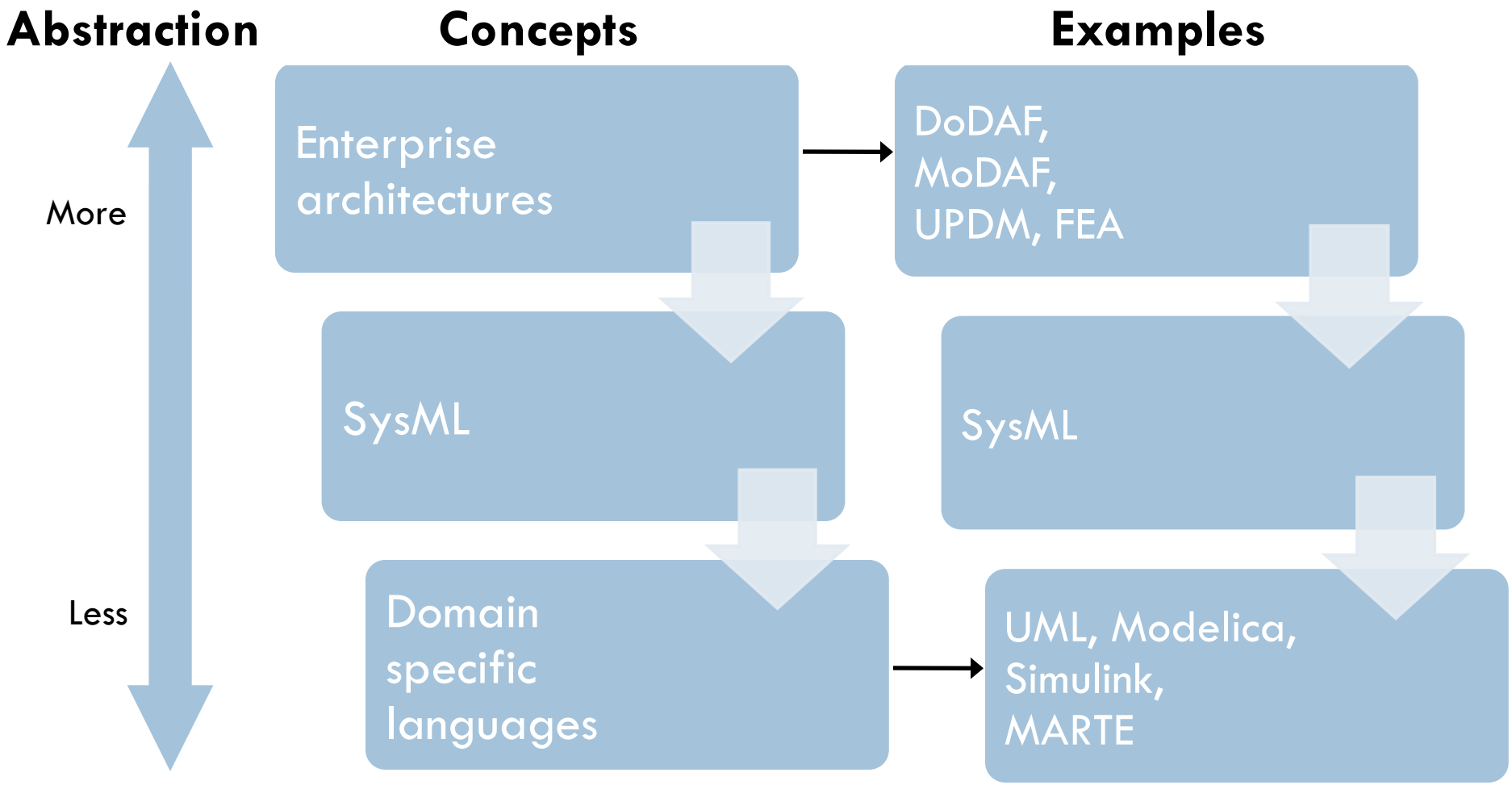
Views into model

Ranges and Types of Modeling Tools

Attributes	Simple Drawing Tools		Enterprise Modeling Suites
Most commonly purchased by	Individual	Department	Organization
Easy to learn and use	Yes[1]	No	No
Semantic consistency imposed by tool?	No	Yes	Yes
Data based repository for model elements	No	Yes	Yes
Framework support	No	Yes	Yes
Diagrams	Yes	Yes	Yes
Matrices (tables and/or reports)	No	Yes	Yes
Able to operationalize data	No	No	Yes
Examples	PowerPoint Visio Gliffy Canvas	Enterprise Architect MagicDraw	Casewise Telelogic Rational
Purchase cost	$100,s	$1000s	$10,000s

[1]Yellow cells indicate the most prominent differentiating advantage for that class of software - From Enterprise Architecture Presentation by Lisa Murphy, Ian Sturken, Andy Schain

SysML and other modeling constructs



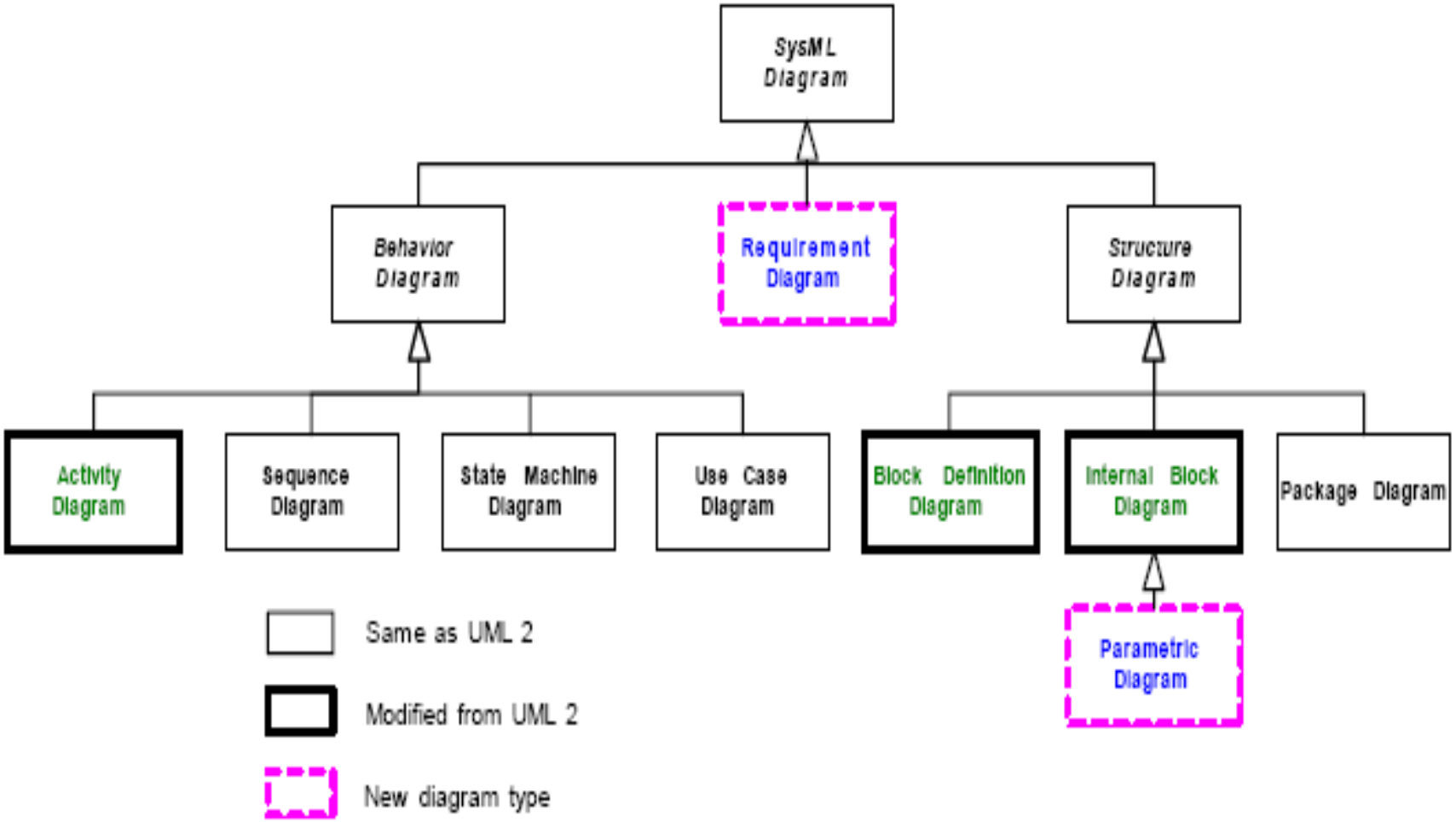
Additional SysML and UPDM semantics

- SysML structure and semantics
- UPDM structure and semantics

SysML Structure and Semantics

SysML may be considered in terms of its diagrams, reports and grammar.

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

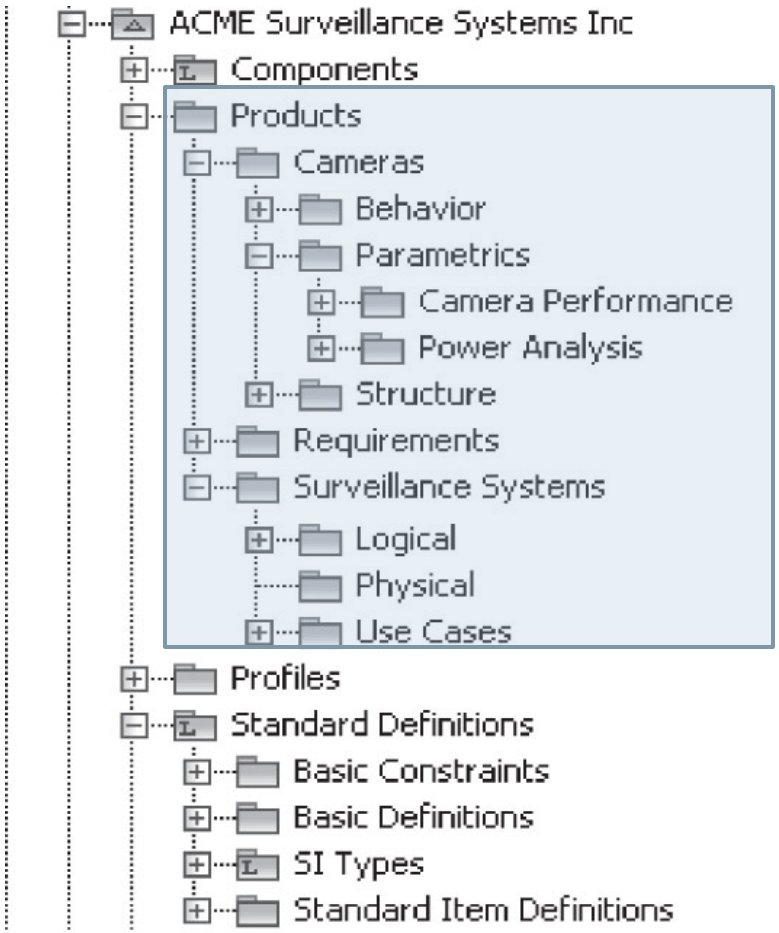


FIGURE 5.4

Browser view of the model's package hierarchy.

Source: A Practical Guide to SysML., copyright © 2008 by Elsevier, Inc. All rights reserved.

Packages shown graphically

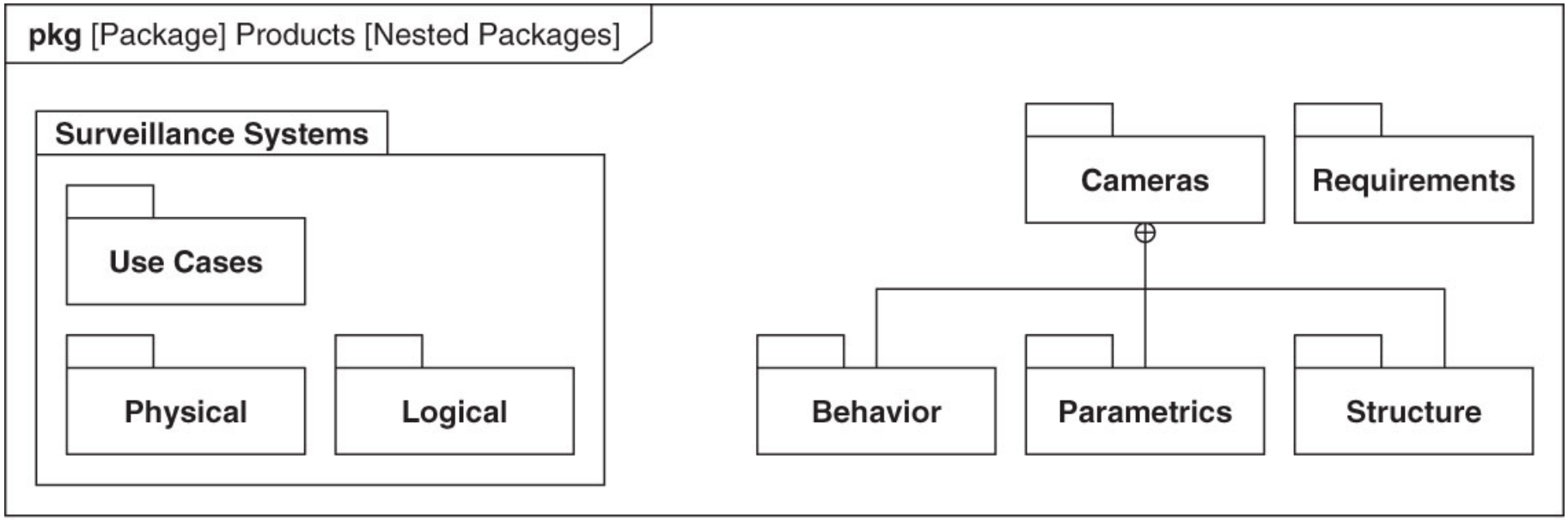


FIGURE 5.1

An example package diagram.

Source: A Practical Guide to SysML., copyright © 2008 by Elsevier, Inc. All rights reserved.

Dependences and stereotypes

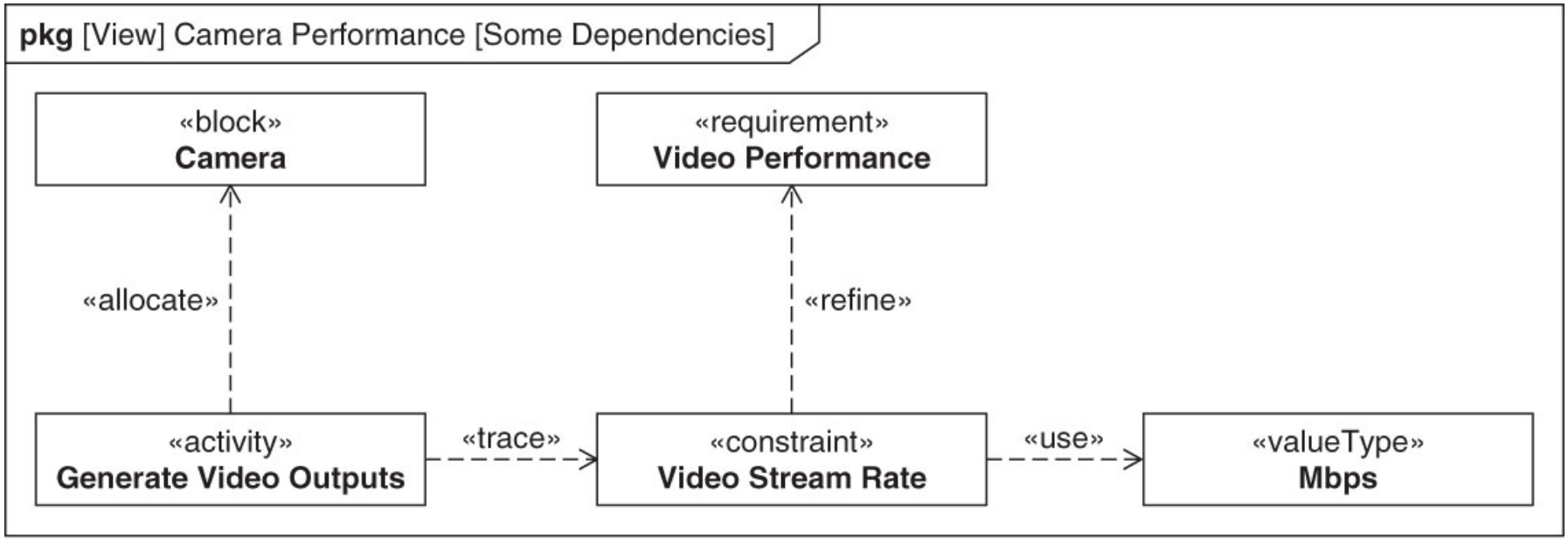


FIGURE 5.10

Example of dependencies in the camera performance view.

Source: A Practical Guide to SysML., copyright © 2008 by Elsevier, Inc. All rights reserved.

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

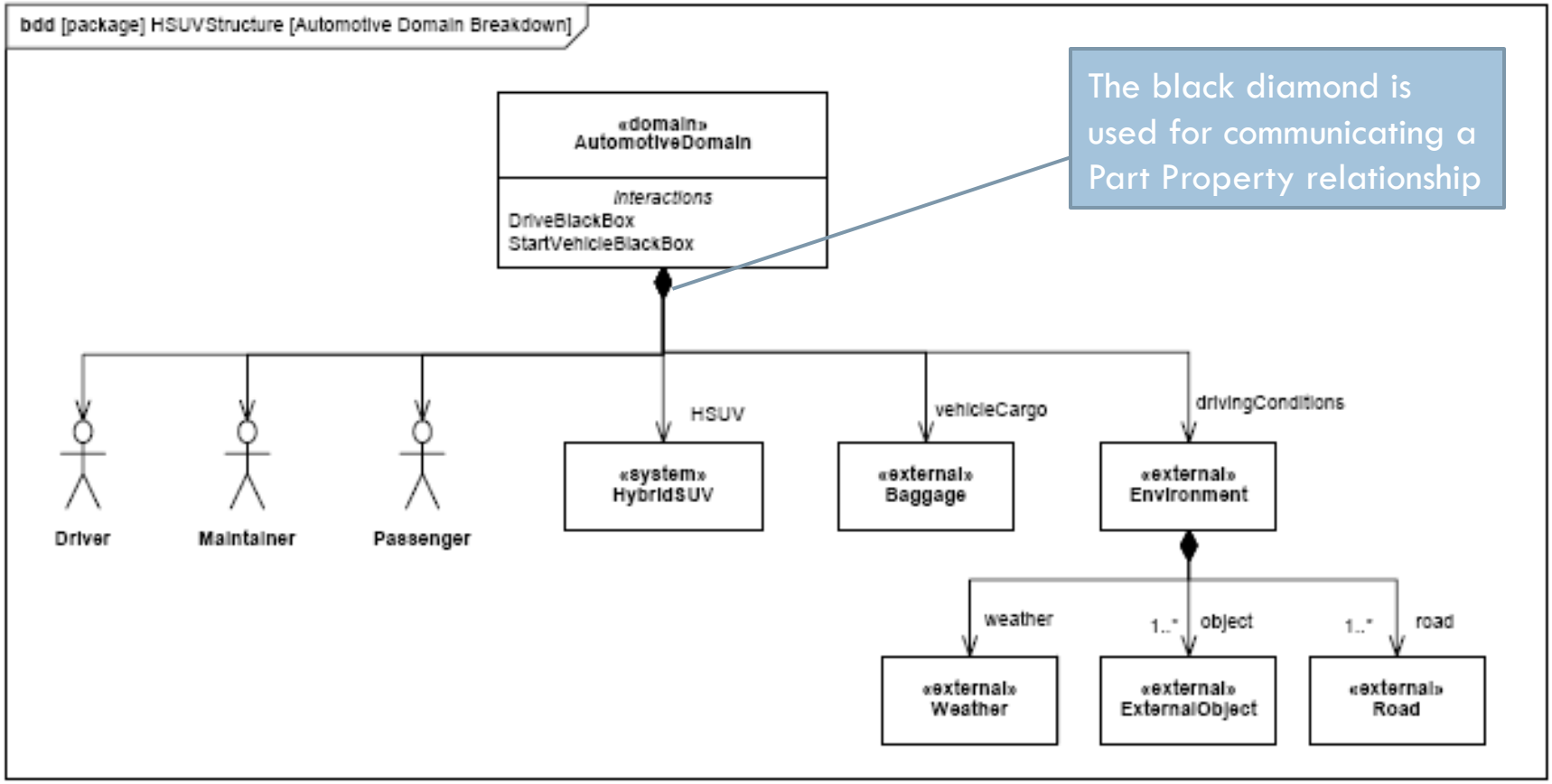


Figure B.15 - Defining the Automotive Domain (compare with Figure B.4) - (Block Definition Diagram)

Source: OMG Specification

Block definition diagram—Association

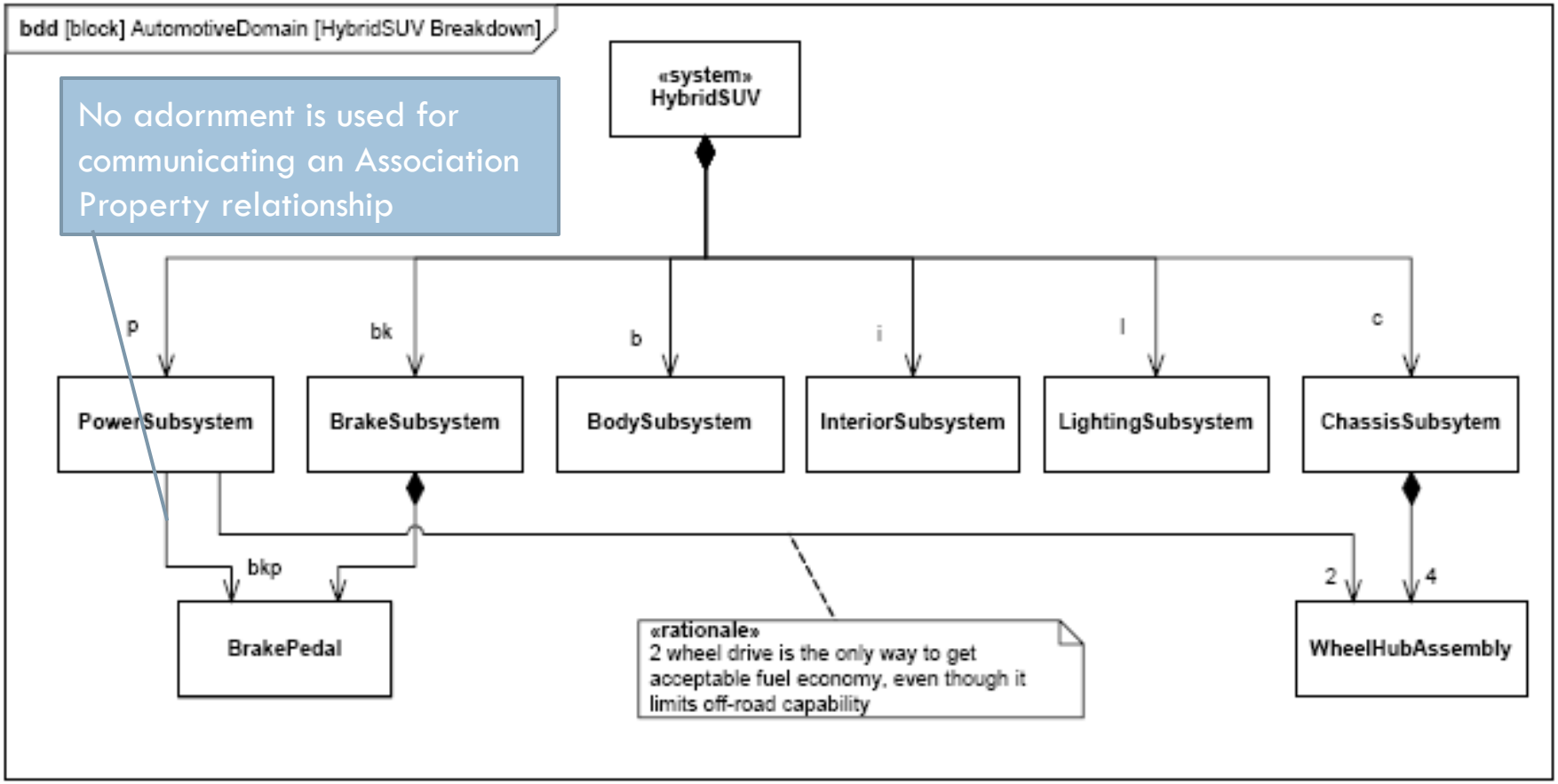


Figure B.16 - Defining Structure of the Hybrid SUV System (Block Definition Diagram)

Source: OMG Specification

Block definition diagram—Reference

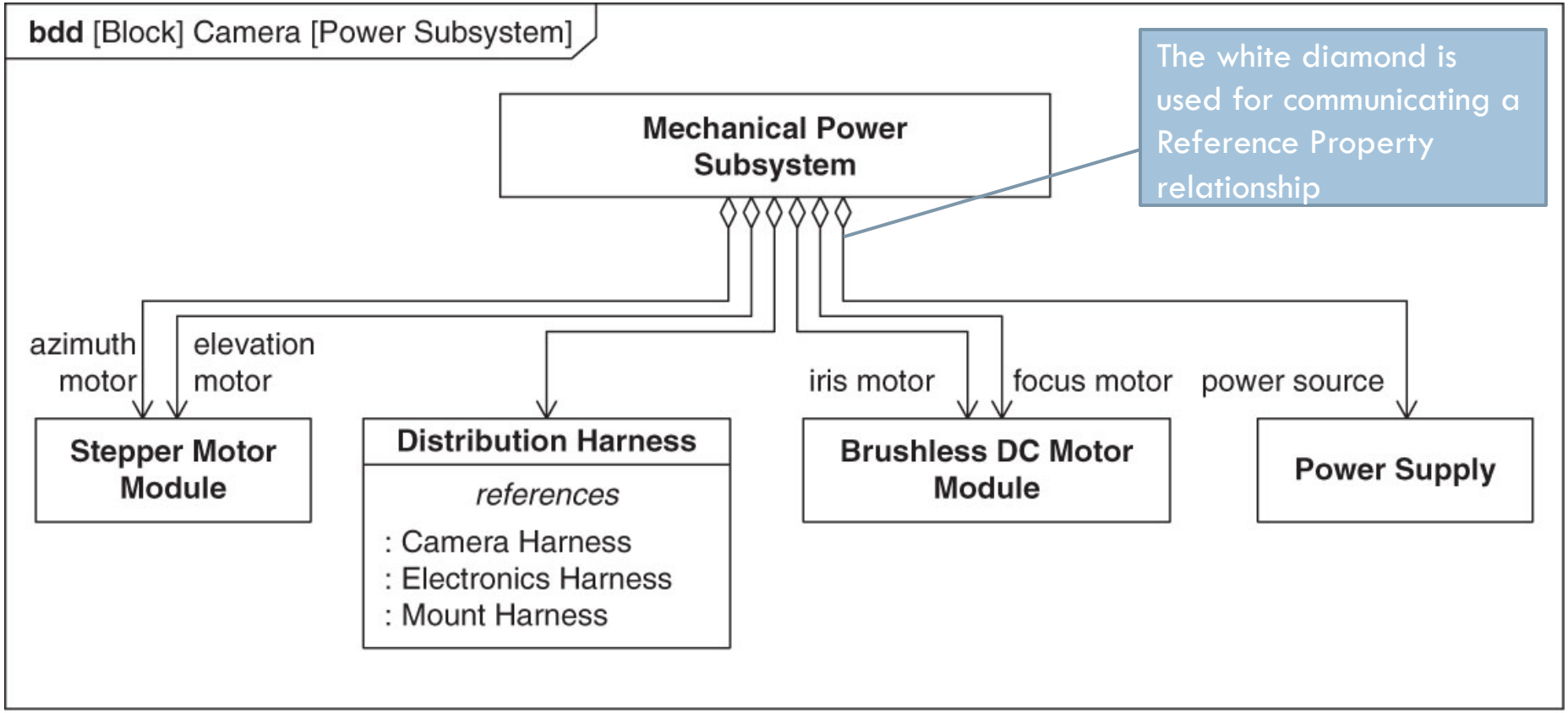


FIGURE 6.10

A reference association on a block definition diagram.

Source: A Practical Guide to SysML., copyright © 2008 by Elsevier, Inc. All rights reserved.

Block compartments

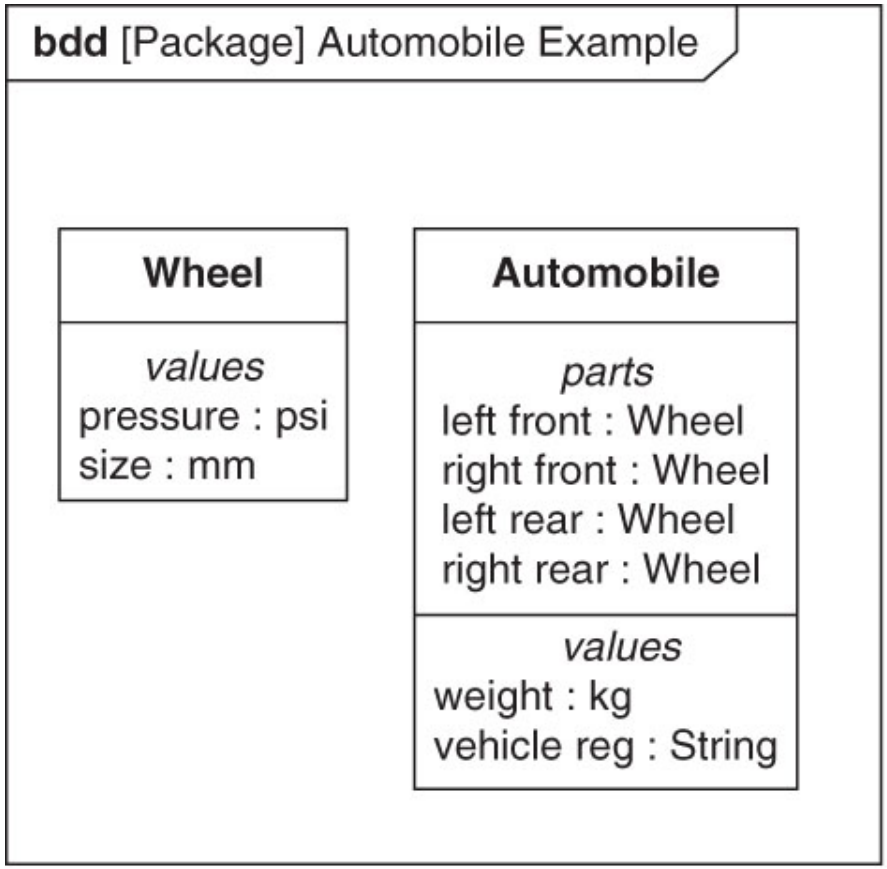


FIGURE 6.4

An automobile with four wheels described as separate parts.

Source: A Practical Guide to SysML., copyright © 2008 by Elsevier, Inc. All rights reserved.

Blocks and inheritance

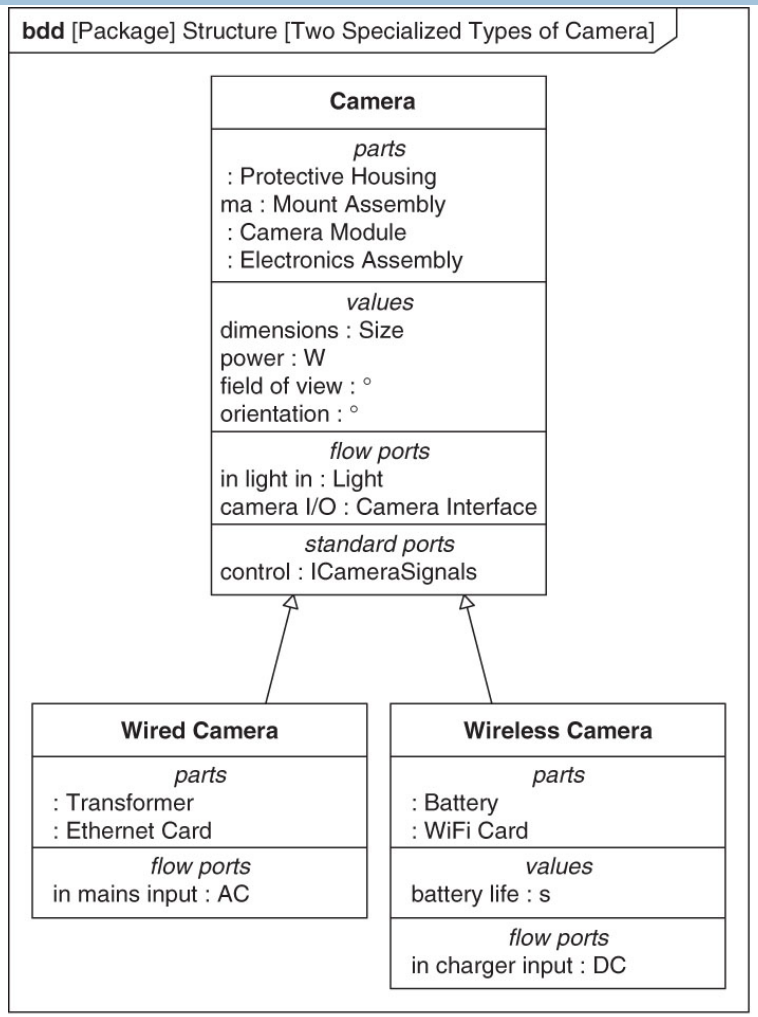
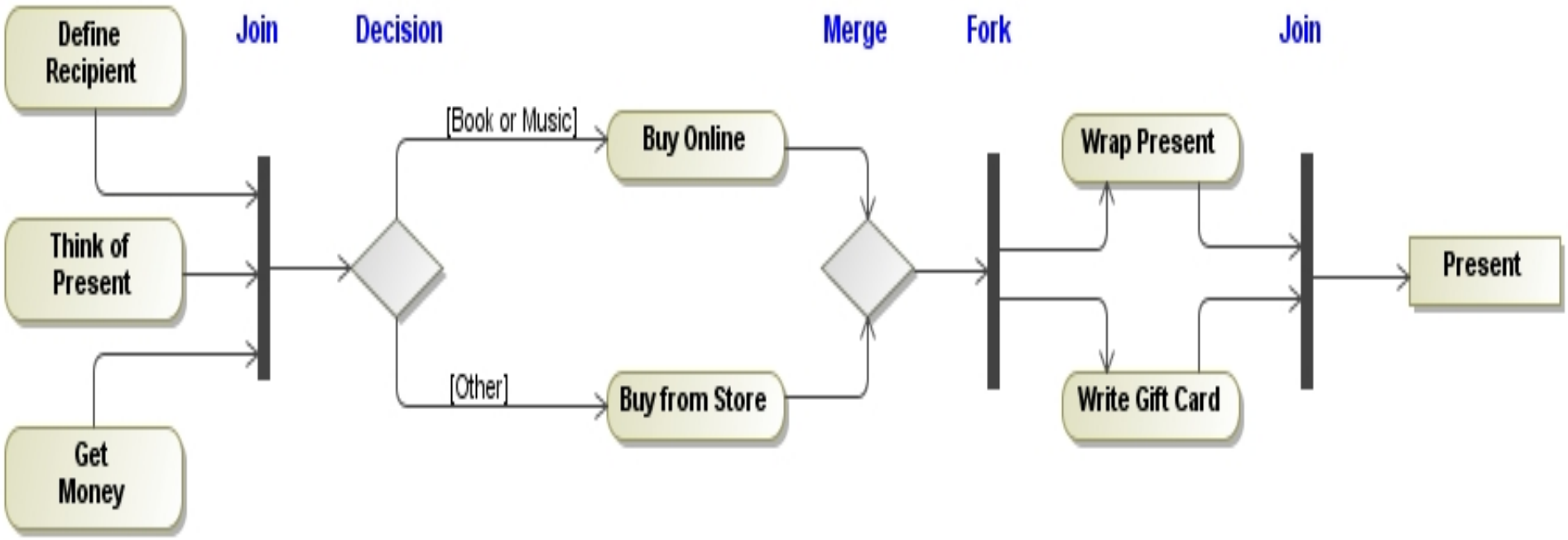


FIGURE 6.35

Example of block specialization.

Source: A Practical Guide to SysML., copyright © 2008 by Elsevier, Inc. All rights reserved.

Activity Diagrams



Sample activity diagram 2

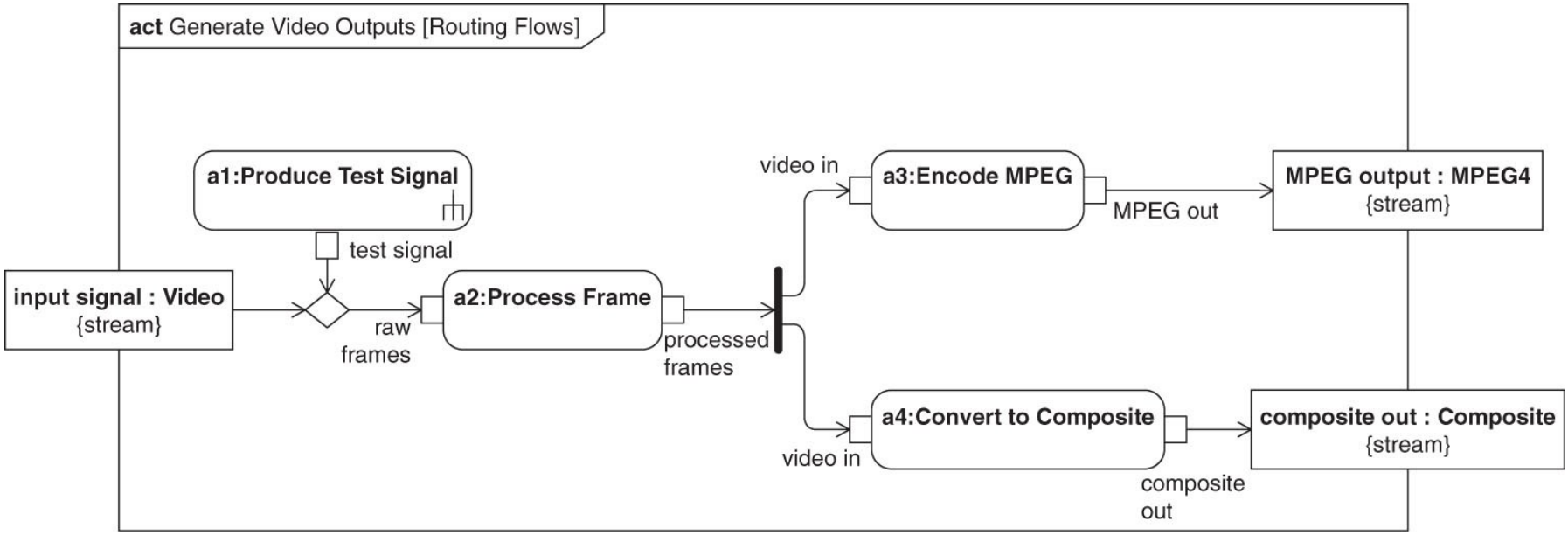
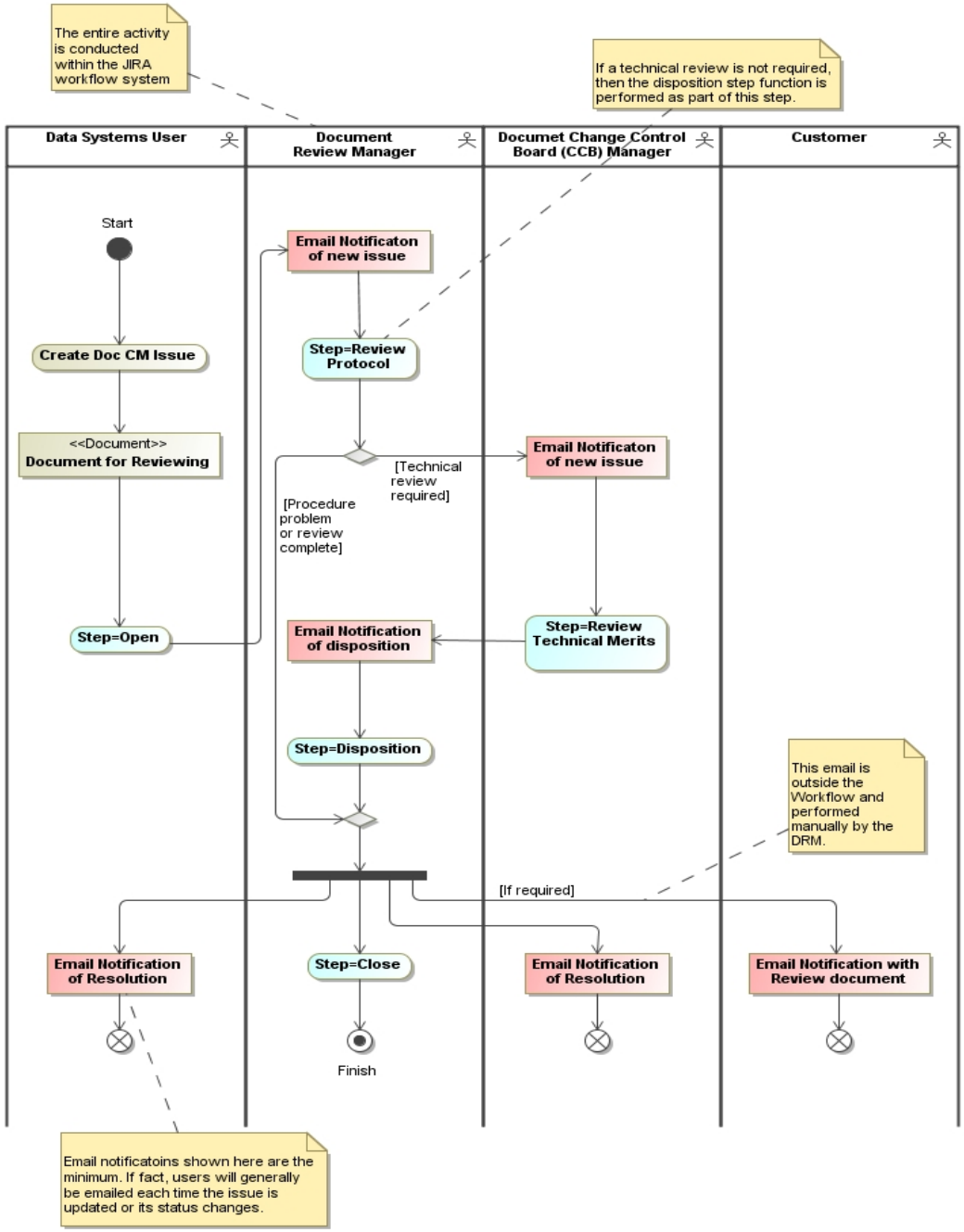


FIGURE 8.1

An example activity diagram.

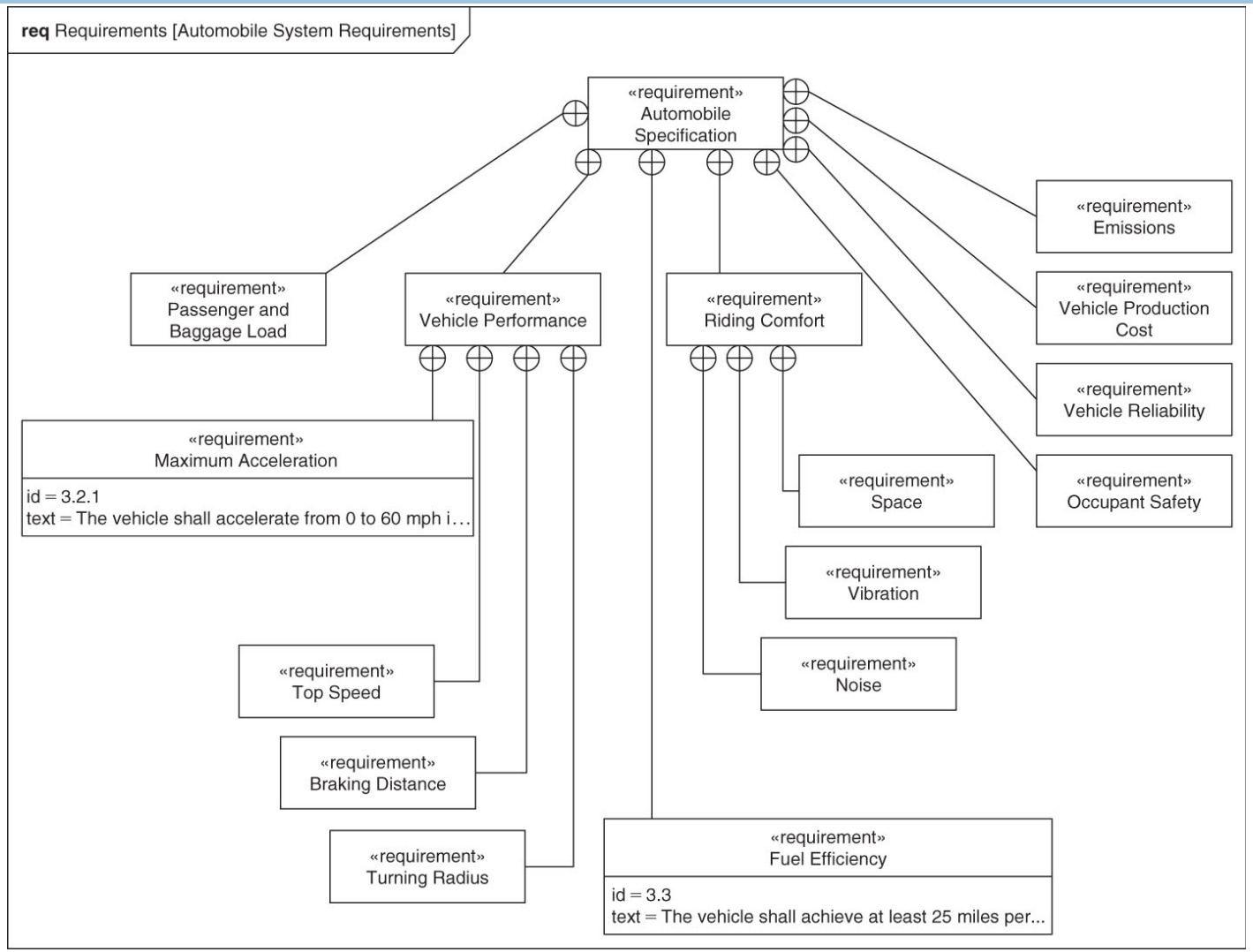
Source: A Practical Guide to SysML., copyright © 2008 by Elsevier, Inc. All rights reserved.


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
Create Doc CM Issue
<<Document>>
Document for Reviewing
Step=Open
Email Notificaton of new issue
Step=Review Protocol
[Technical review required]
[Procedure problem or review complete]
Email Notificaton of new issue
Step=Review Technical Merits
Email Notification of disposition
Step=Disposition
This email is outside the Workflow and performed manually by the DRM.
[If required]
Email Notification of Resolution
Step=Close
Finish
Email Notification of Resolution
Email Notification with Review document
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 <u>how requirements relate to each other</u>.
- Provide concise and unambiguous information showing <u>how requirements relate to other model elements</u>—the project lifecycle.

Requirements diagram—Example



Source: A Practical Guide to SysML., copyright © 2008 by Elsevier, Inc. All rights reserved.

Requirements dependences

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

Refine, satisfy &dervieReqt

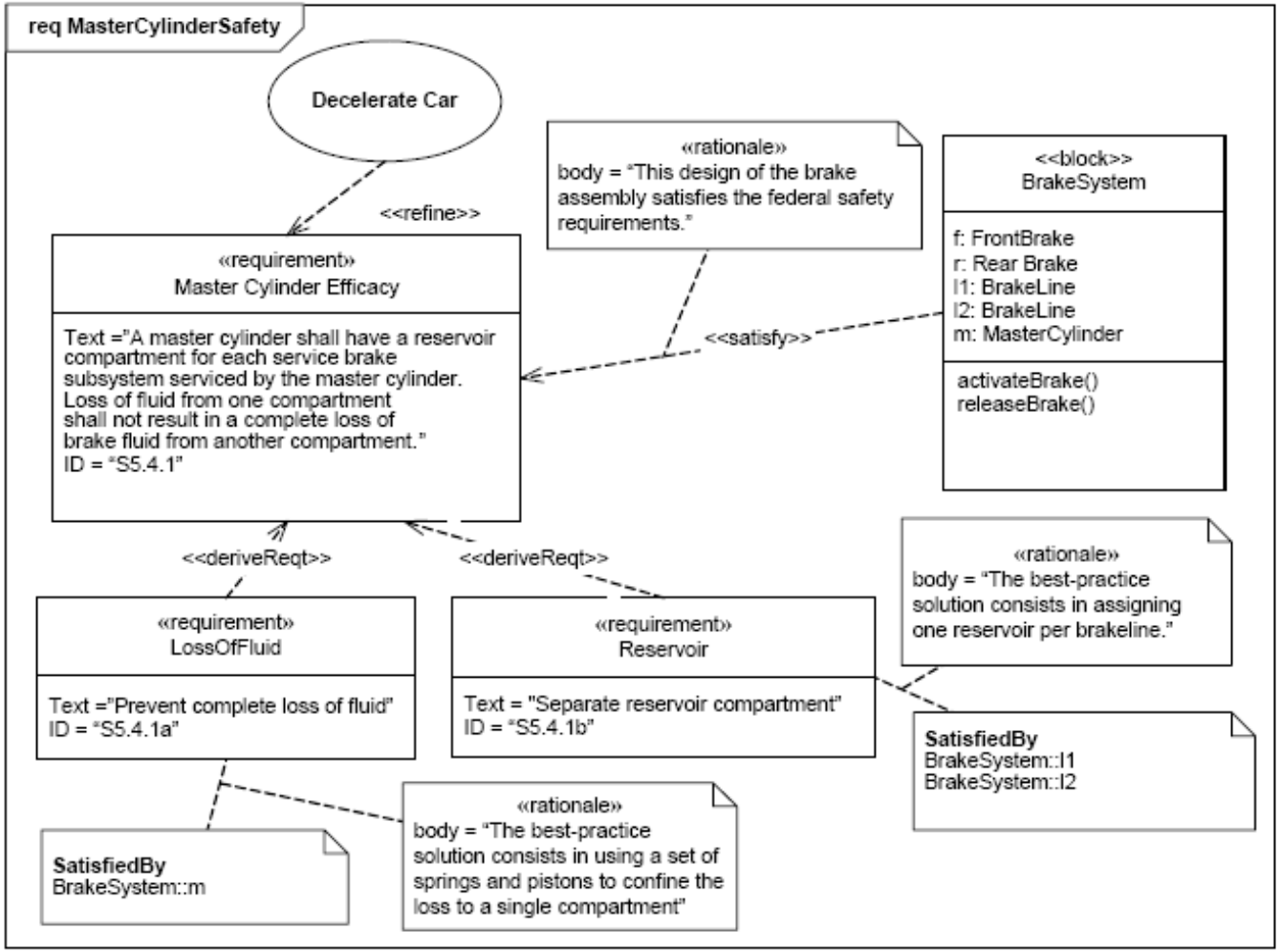


Figure 16.4 - Links between requirements and design

Source: OMG Specification

Copy

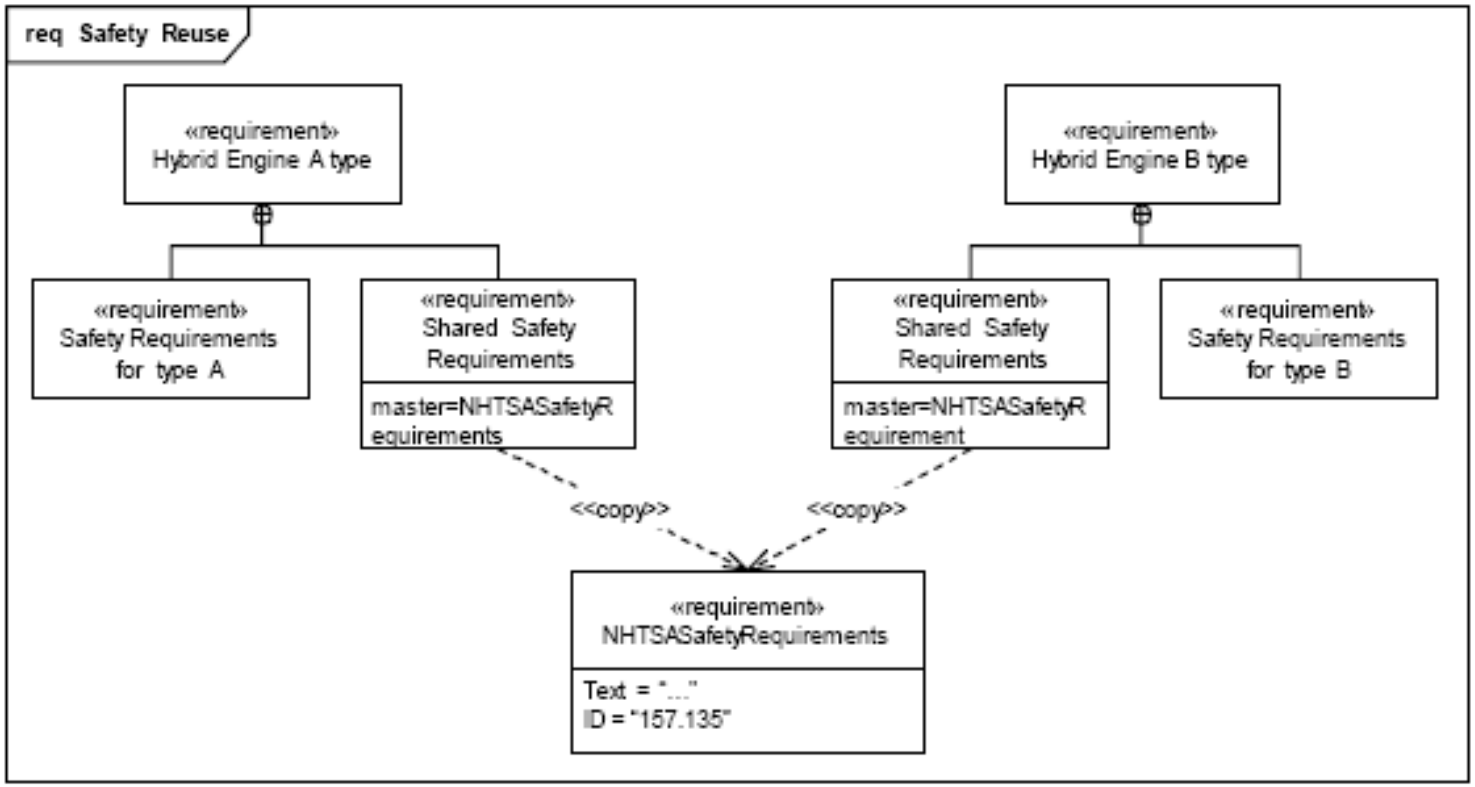
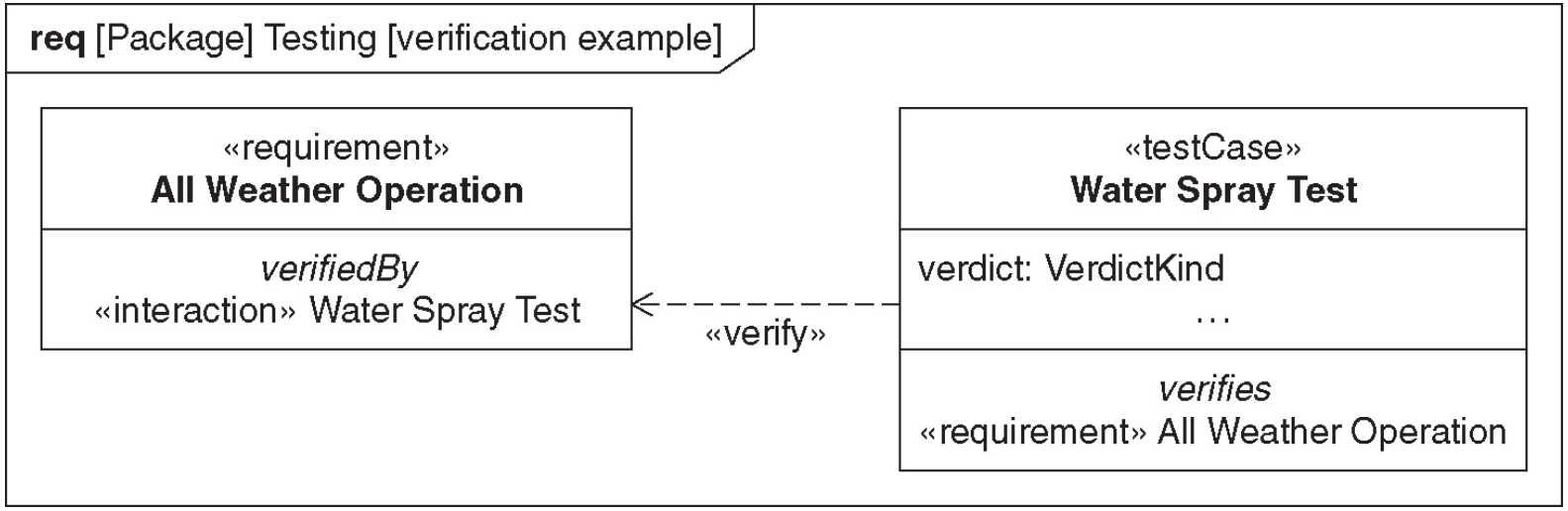


Figure 16.6 - Use of the copy dependency to facilitate reuse

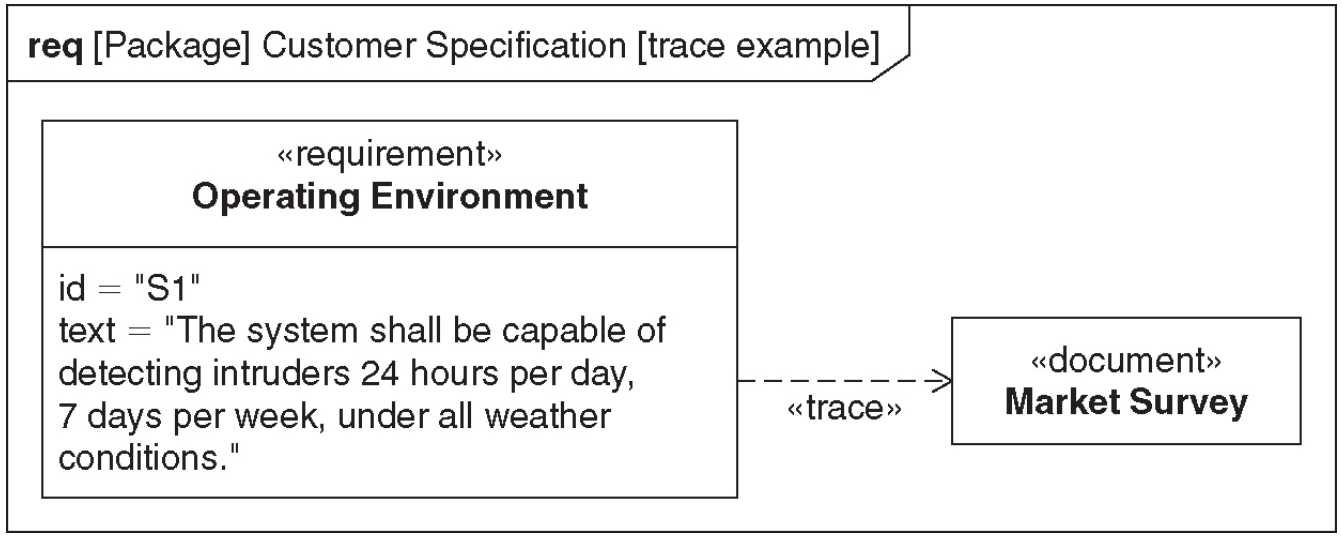
Source: OMG Specification

Verify



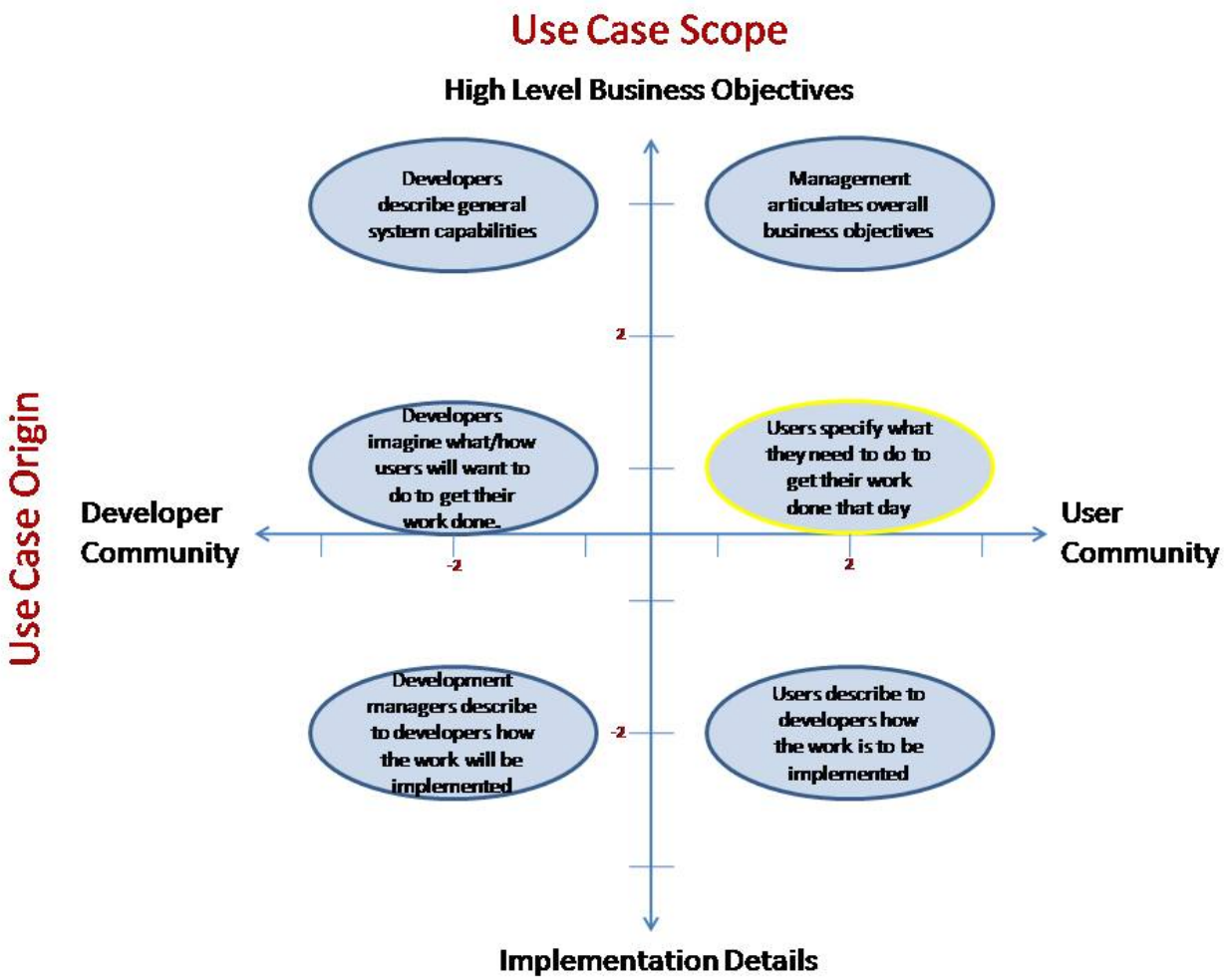
Source: A Practical Guide to SysML., copyright © 2008 by Elsevier, Inc. All rights reserved.

Trace



Source: A Practical Guide to SysML., copyright © 2008 by Elsevier, Inc. All rights reserved.

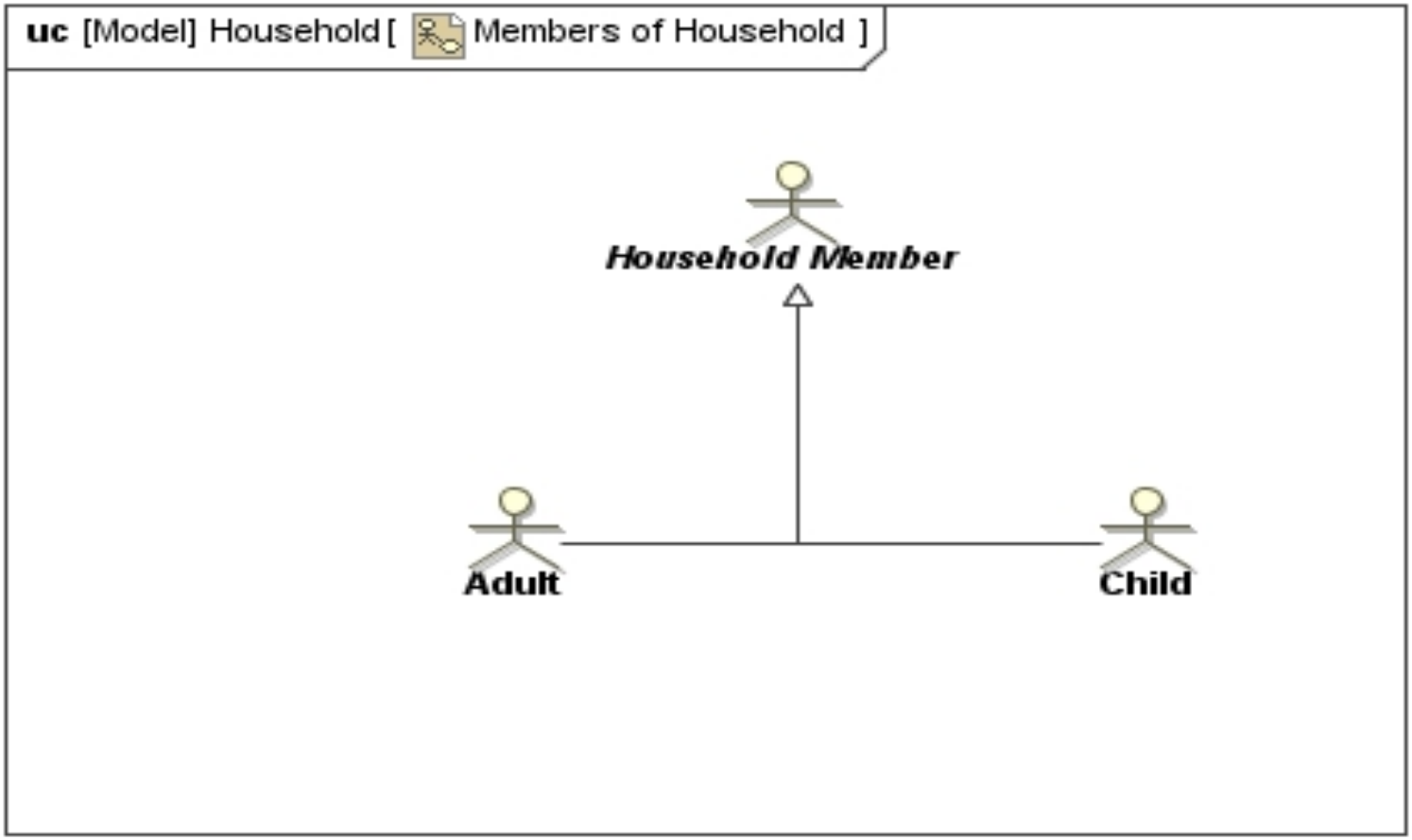
Use case scope



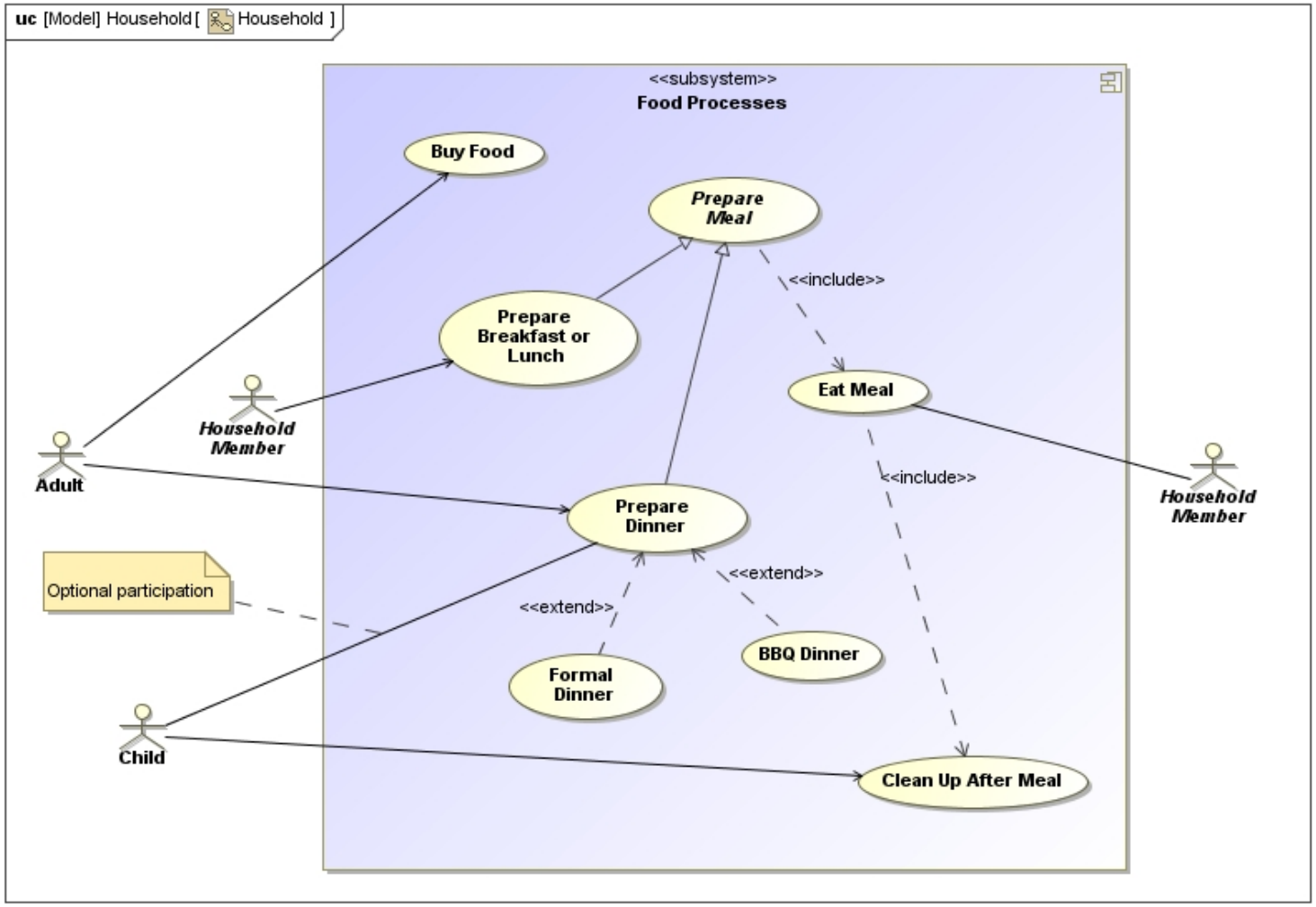
Use cases and actors

- 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.

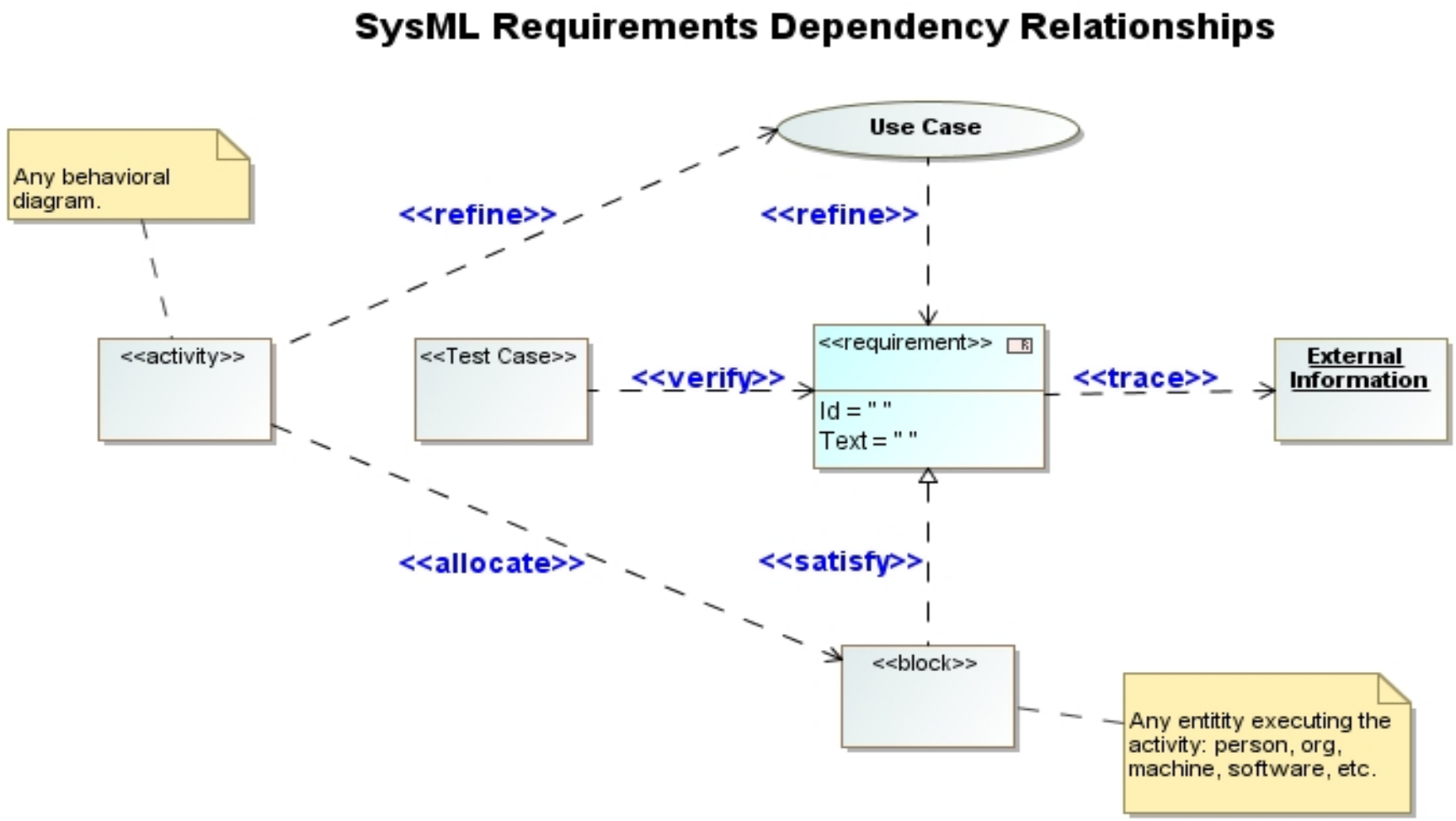
Actors for a household model

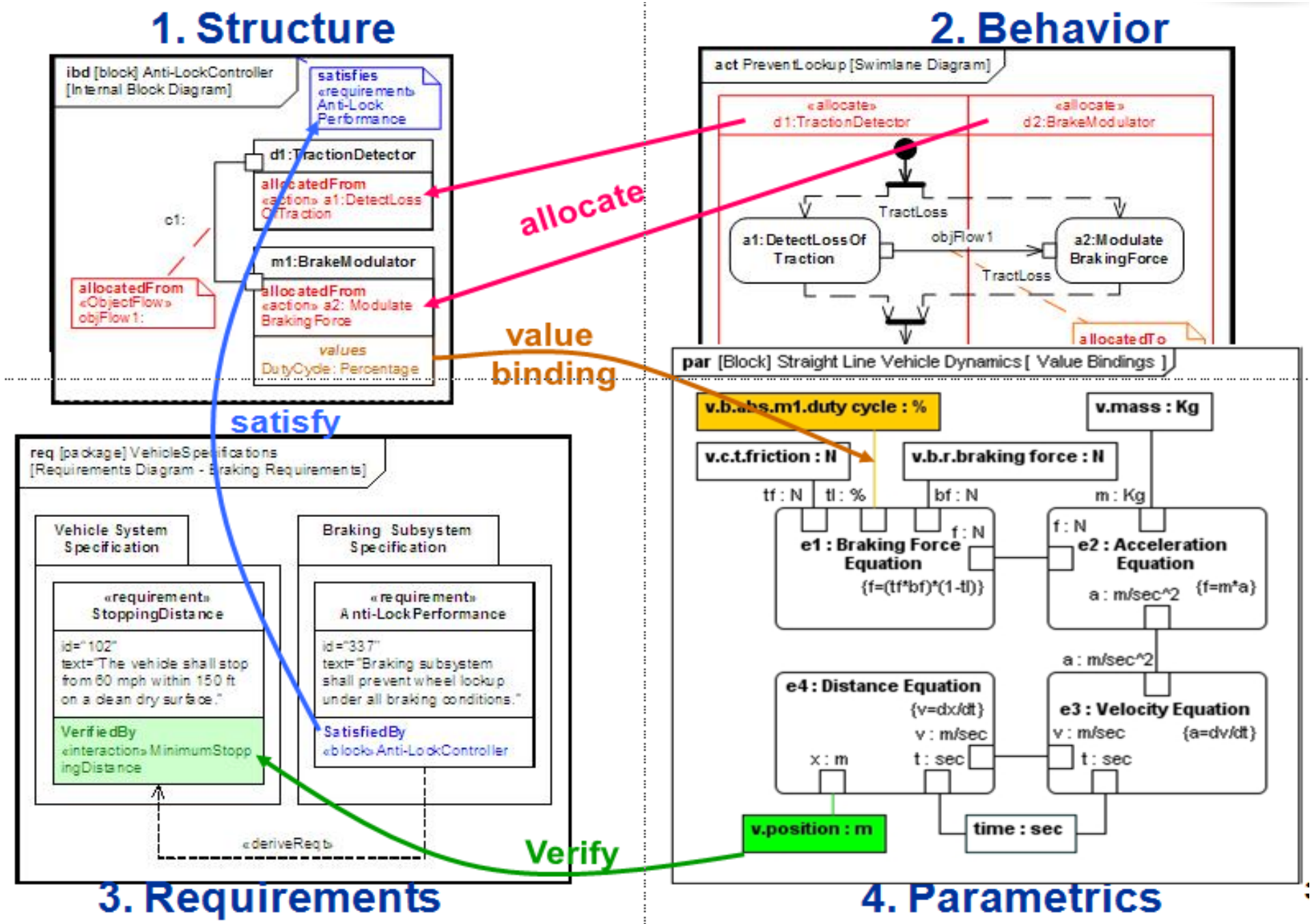


Use cases for household food processes



SysML requirements relationships





Source: OMG Specification

UPDM Structure And Semantics

To be done by Ian?

UPDM Overview

OV2 Elements

- Operational Nodes
- Needline
- Information Element

OV3 Elements

- Operational Activity
- Needline
- Item Name
- Operational Node

OV4 Elements

OV5 Elements

OV7 Elements

Use Cases?

Requirements?

Summary

- Key Features
- Conclusion
- A Historical Perspective, video clip

End of Part I

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.

Conclusion

- **Why you <u>don't</u> want to model**
  - Modeling is hard
  - Modeling tools are difficult
  - Modeling will likely require cultural changes
- **Why you <u>do</u> want to model**
  - It increases the <u>rate</u> of communications
  - It increases the <u>precision</u> of communications
  - It reduces <u>tacit</u> information
  - It promotes a <u>common understanding</u> 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.

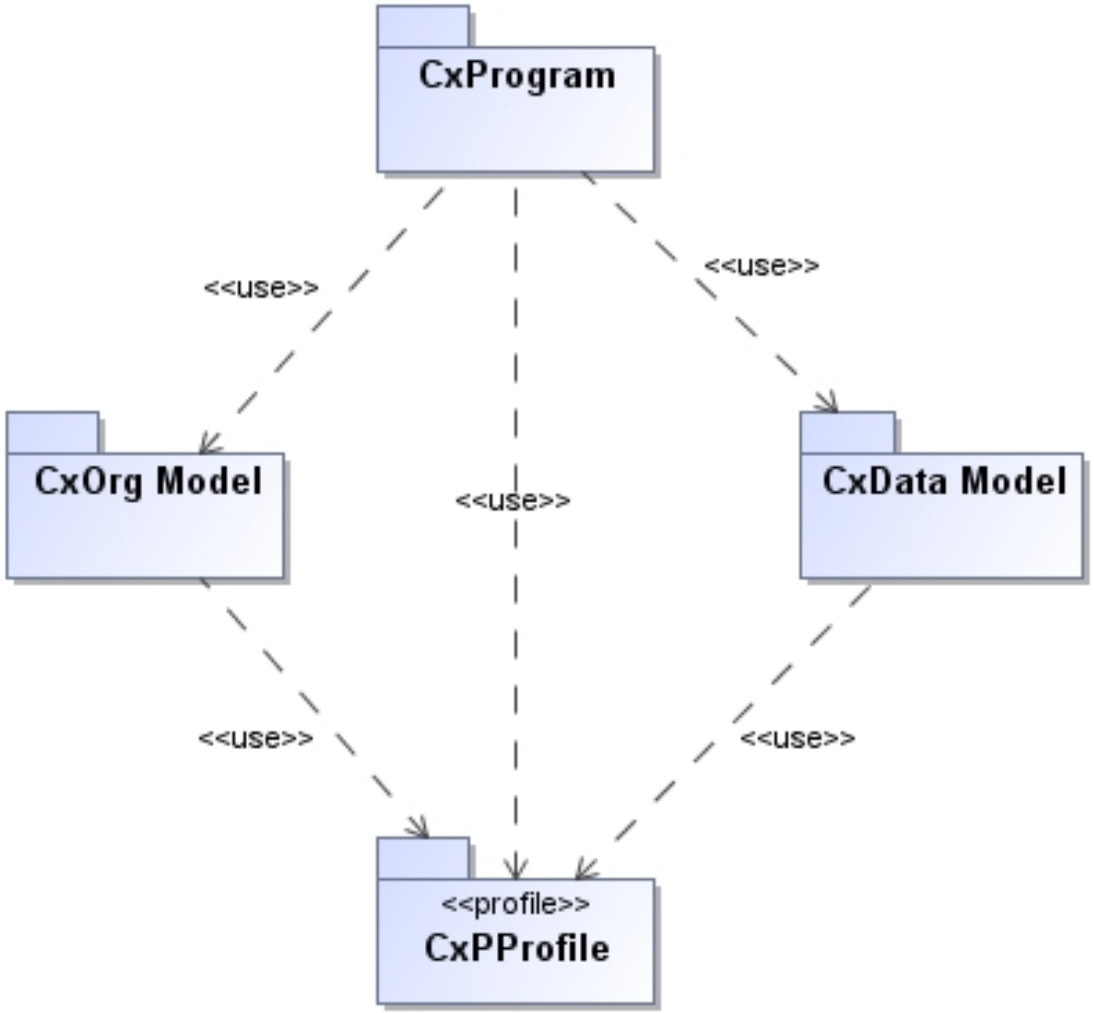
A historical perspective

Demonstration

- Ares
- Mission Information System (~CoFR)
- FireSat

Ares Demonstration

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 :

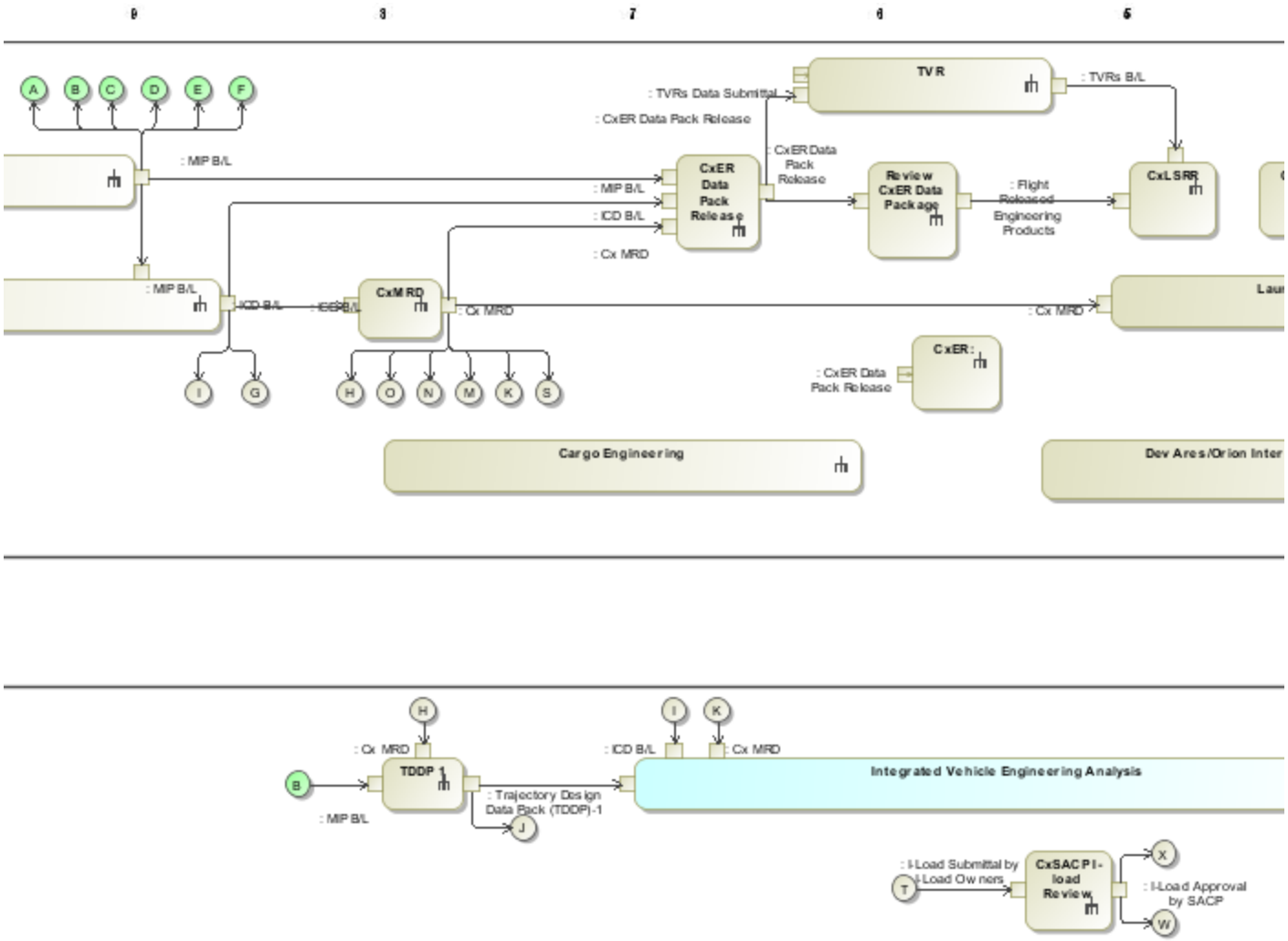
MPPF

Offline Processing

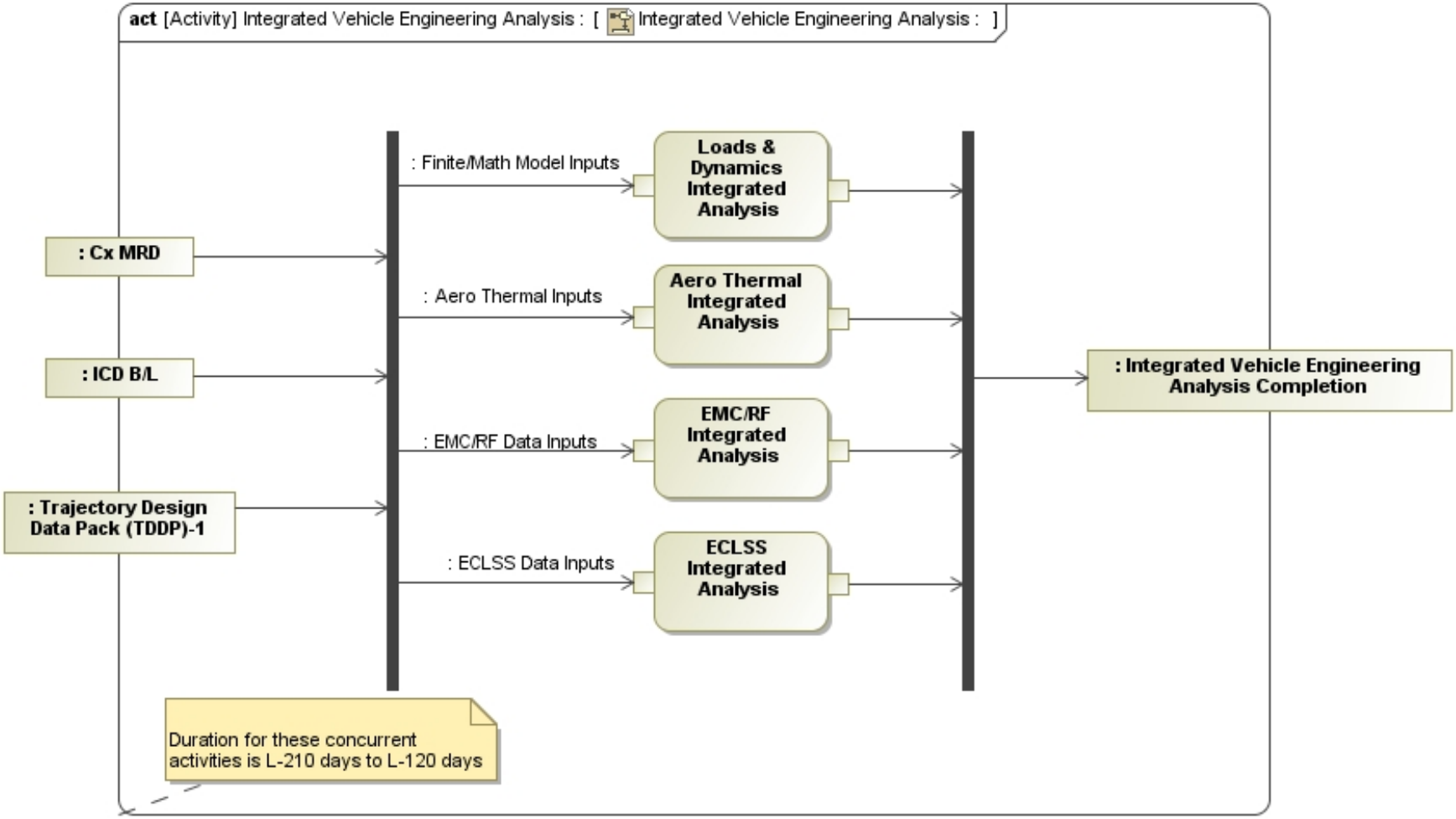
CM/SM Stack Integration

VAB Integrated Operations

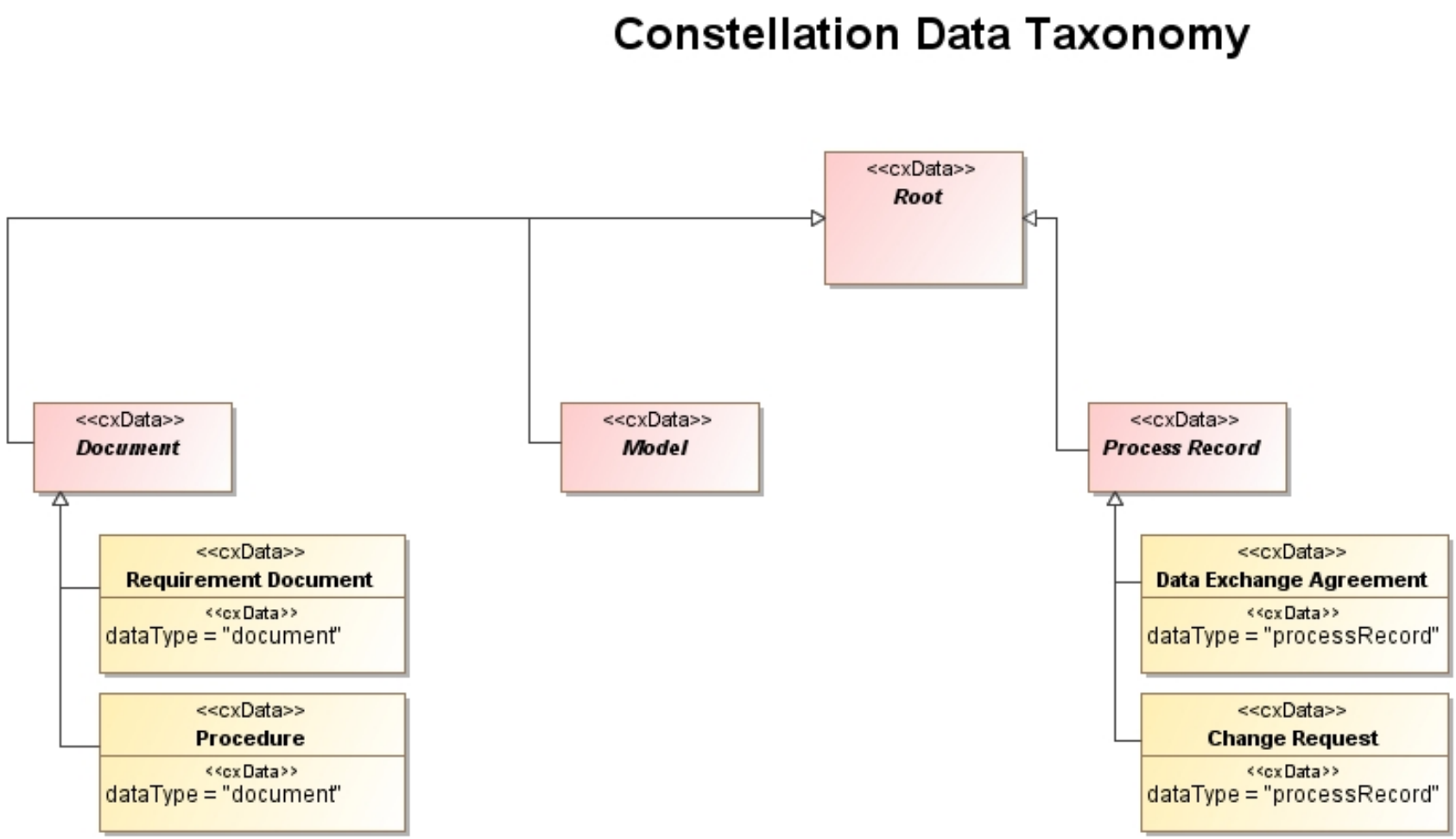
Generic Mission Template



Integrated Vehicle Engineering Analysis



Constellation Data Taxonomy

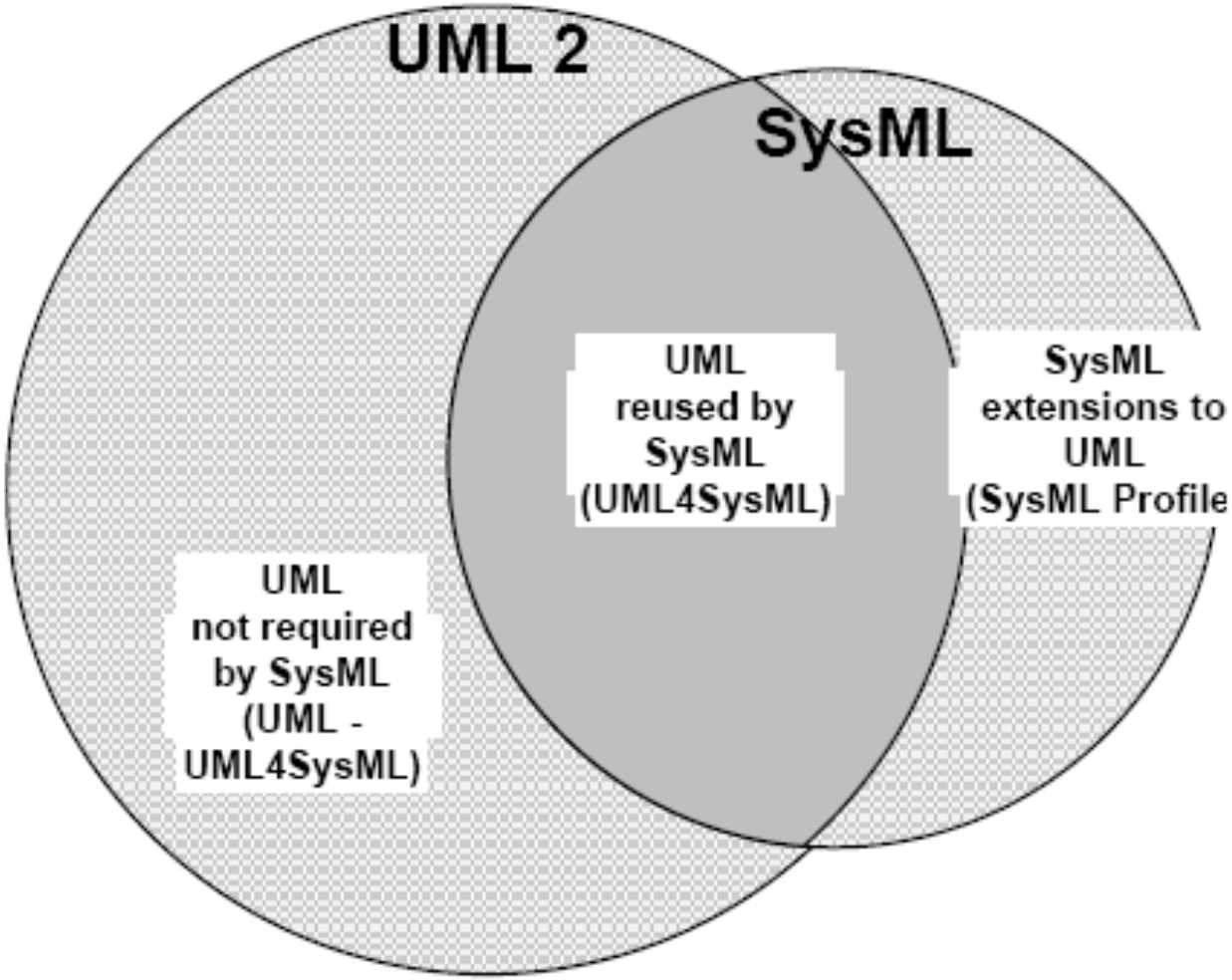


MIS Demonstration

FireSat Demonstration

Backup

Relationship between UML and SysML



Source: OMG Specification

UML/SysML metamodel

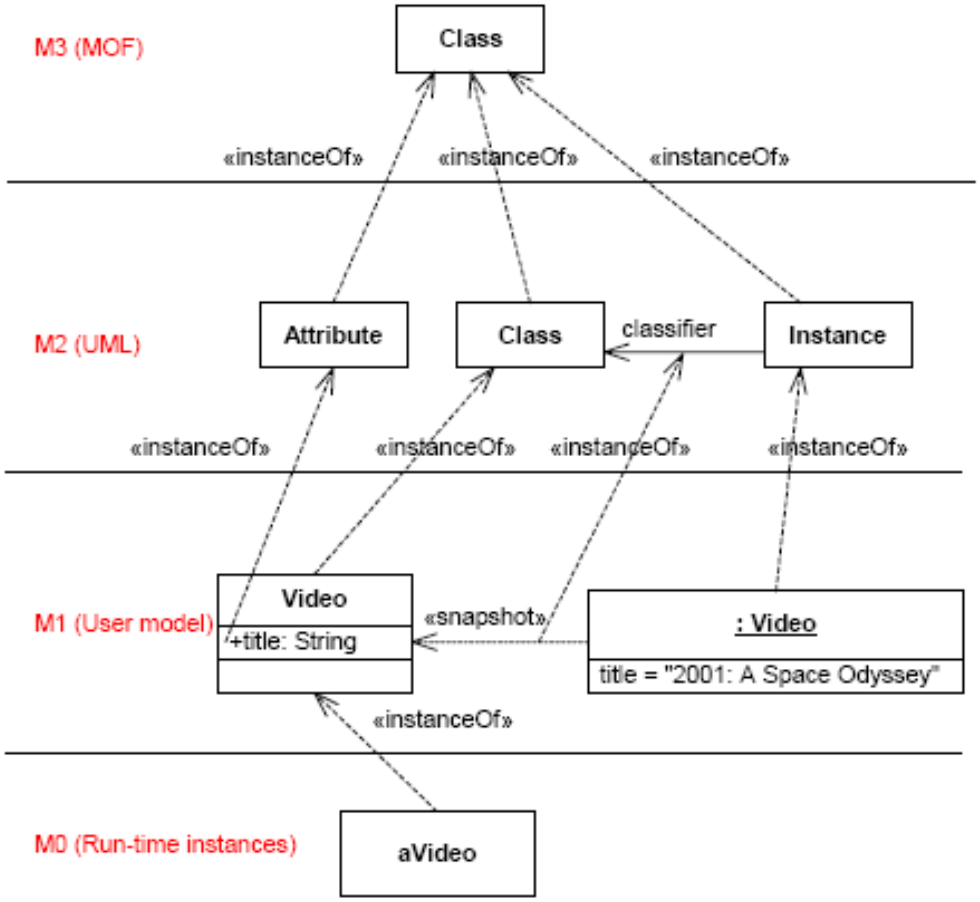


Figure 7.8 - An example of the four-layer metamodel hierarchy

Source: OMG Specification

Copyright Information

OMG Source material is reprinted with permission. Object Management Group, Inc. (C) OMG. 1997 - 2006.