Organizing the Model with Packages (Part 1 – SysML Concepts)

Section Objectives

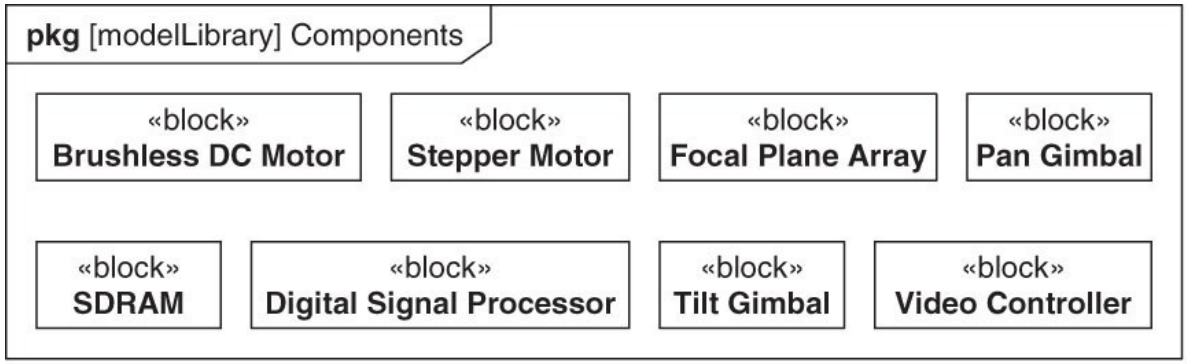
- In this Section, you will learn
  - How to organize a model using Packages
  - How to depict model organization using a Package Diagram

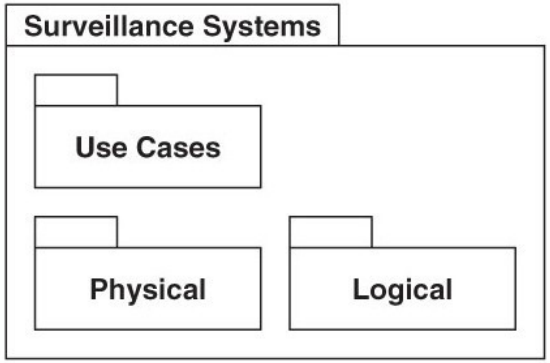
Overview

- This section will discuss:
  - Package Concepts
    - Purpose of Packages and Package Diagrams
    - Package Nesting
    - Model Organization
    - Model Library
    - Containment Relationship
    - Namespace
    - Views and Viewpoints
  - Package Modeling for In-Class Project

Purpose of Packages and Package Diagrams

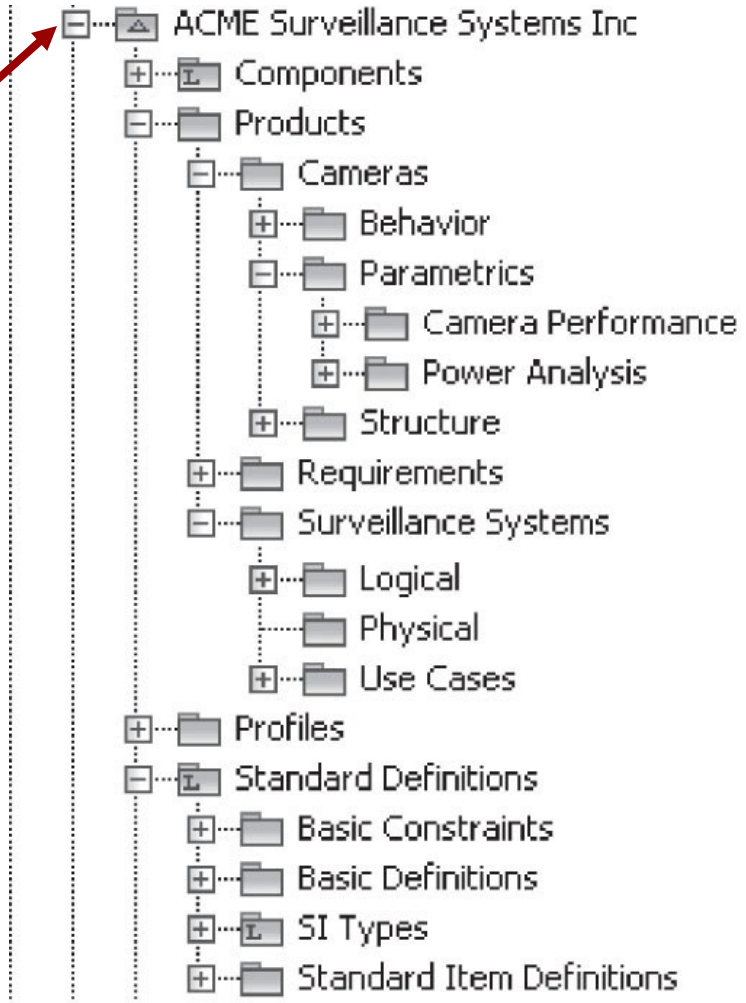
- Packages are used for model organization in SysML
  - Similar to Folders in Windows
  - Serve as containers for model elements, diagrams, and other packages
- Package Diagrams provide a graphical depiction of the model organization and/or package content





Package Nesting

- Packages can be organized into a hierarchy (packages within packages)
- A Model is the top-level package in a nested package hierarchy



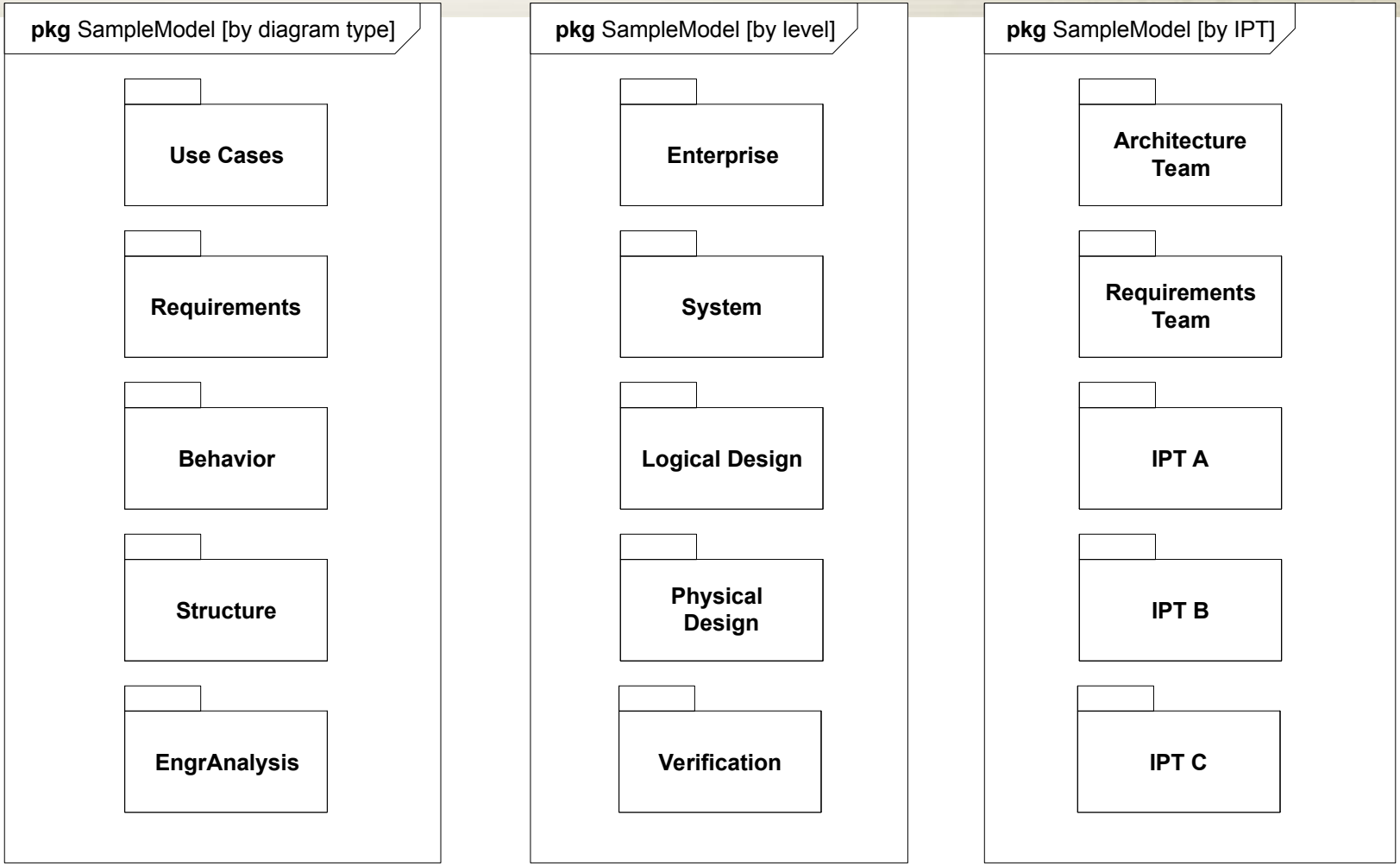
Motivations for Model Organization

- Re-use
  - Model Library – Good model organization supports re-use of model elements
- Access Control
  - Within a modeling tool, packages can be assigned write privileges to particular users
- Navigation
  - Good model organization impacts how easily elements and diagrams can be found
- Configuration Management
  - Within a modeling tool, packages can be locked while a user makes changes
- Data Exchange
  - Packages can be imported and exported between models

Model Organization Methods

- Possible ways to organize a model's packages:
  - By system hierarchy (e.g. system level, subsystem level, component level)
  - By process life cycle where each model subpackage represents a stage in the process (e.g. requirements analysis, system design)
  - By teams that are working on the model (e.g. Requirements Team, Integrated Product Team 1, 2)
  - By the type of model elements contained in it (e.g. requirements, behavior, structure)
  - By model elements that are likely to change together
  - By model elements organized to support reuse (e.g. model libraries)
  - A combination of the preceding

Example Model Organization Methods



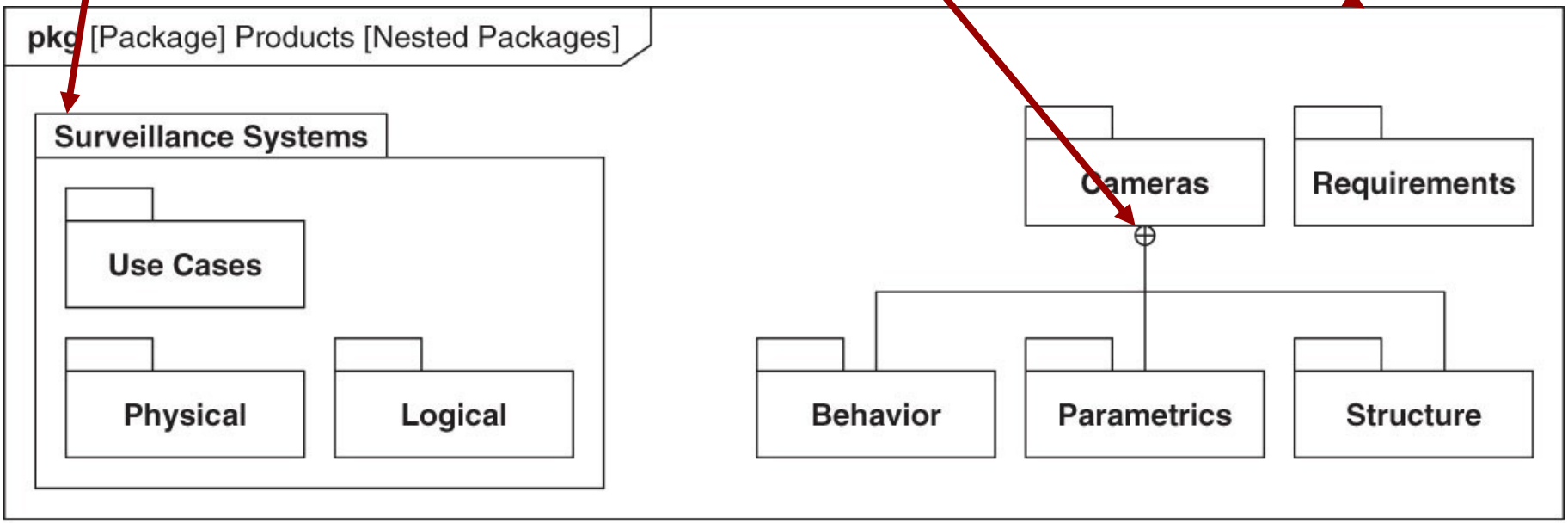
By Diagram Type | By Hierarchy | By IPT

Model Library

- Model Library – a set of packages that contain reusable model elements
- Any packageable model element can be included in a model library:
  - Blocks
  - Value Types
  - Activities
  - Constraint Blocks
- Best Practice: Create an element once, place it in the Model Library, and use it on multiple diagrams

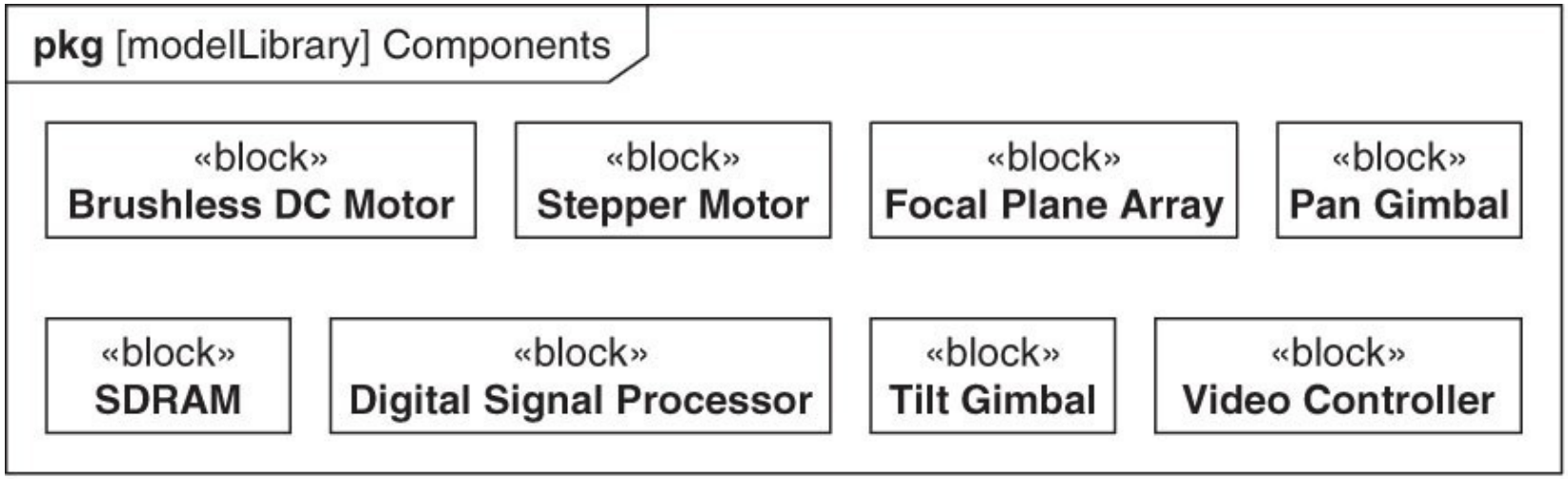
Package Diagram Containment Relationship

- Depicts Package Hierarchy
- Three techniques (displayed below)
  - Packages contained within 'frame' of parent package
  - Packages contained within a package
  - Crosshair pointing to the parent package



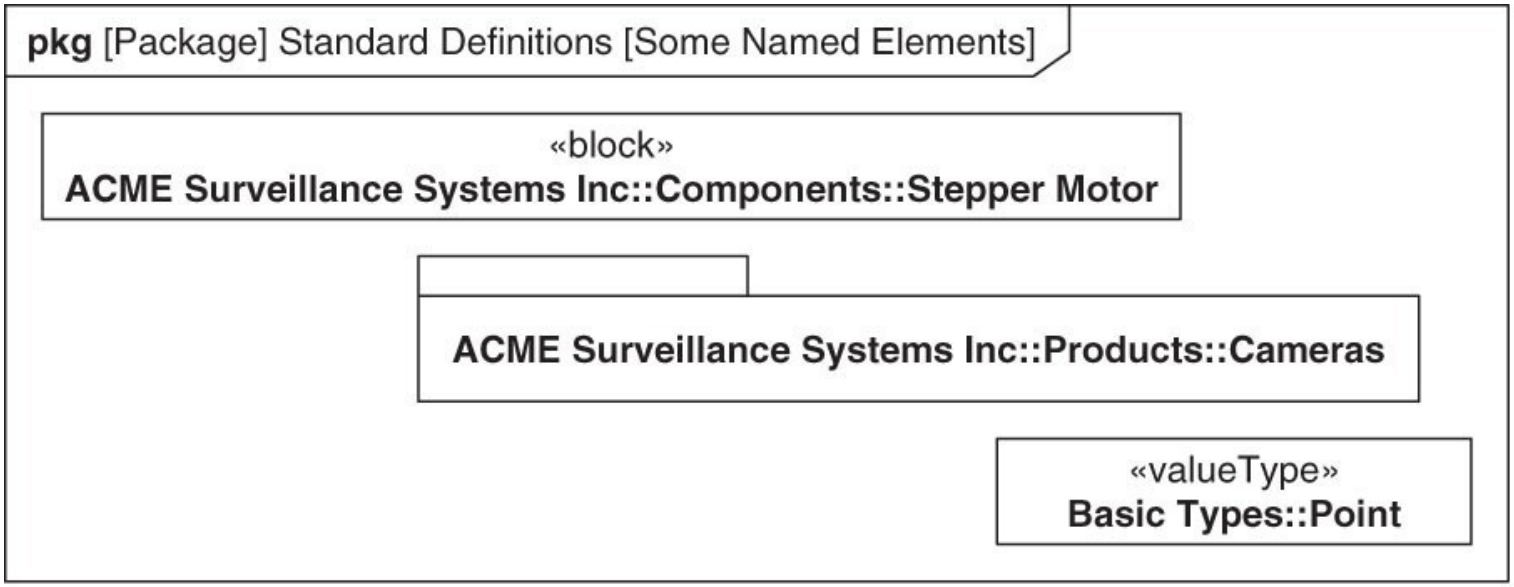
Containment Relationship (cont'd)

- The Containment Relationship can also be used to show which model elements are contained in a Package
- Model Elements may include: use cases, requirements, blocks, activities, etc.



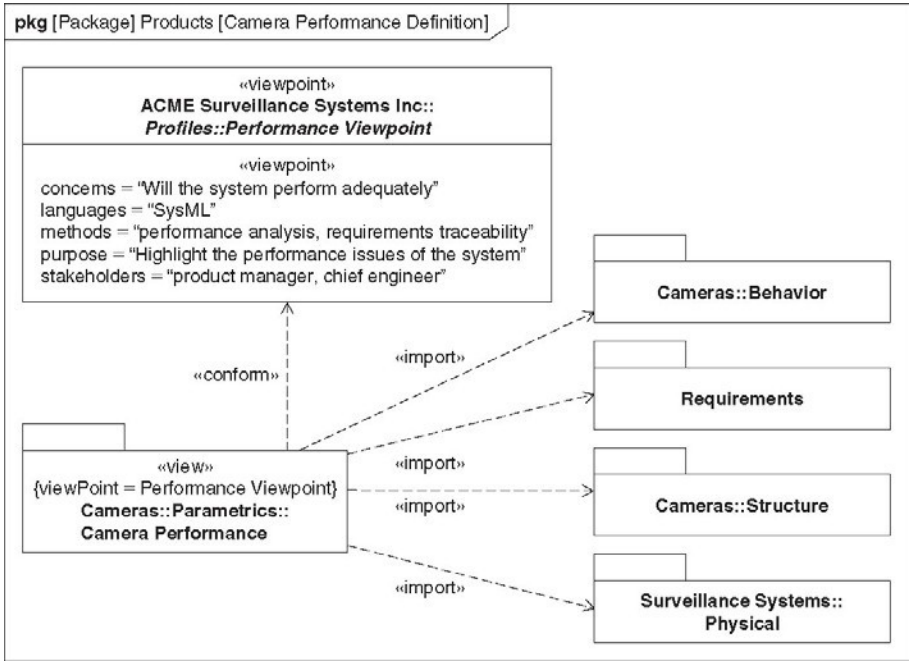
Namespace

- A package is the Namespace for all elements within the package
- Model elements can be contained in one package but used in a diagram that is contained in a different package
- A Qualified Name is used to clarify where the model element resides in the package structure



Views and Viewpoints

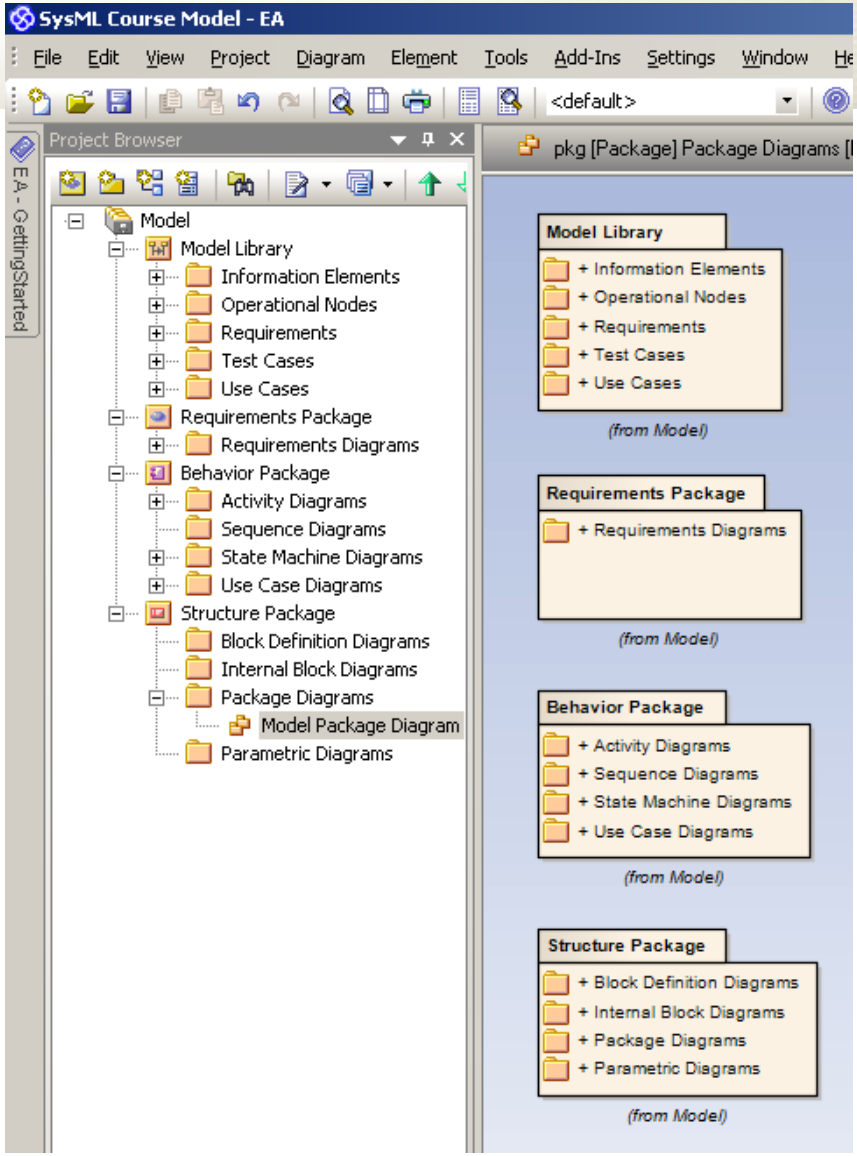
- **Viewpoint – describes a perspective of interest of one or more stakeholders (e.g. system performance)**
- **Viewpoints include properties that identify:**
  - Purpose of the viewpoint
  - Interested stakeholders
  - Concerns of the stakeholders
  - Language used to present the view
  - Methods used to establish the view
- **View – a package that *conforms* to (or supports) the viewpoint**
- **Views *import* model elements that contain relevant information**
  - Views provide the model information that address stakeholder concerns



Package Modeling for In-Class Project

- Set-up High-Level Packages in Project Browser
  - Model Library
  - Requirements
  - Behavior
  - Structure
- For each High-Level Package, add Lower-Level packages for each diagram type, (e.g. for Structure, add):
  - Block Definition Diagrams
  - Internal Block Diagrams
  - Package Diagrams
  - Parametric Diagrams
- Create a Package Diagram to depict the Model Organization

Package Organization for Parking Garage Gate



Summary

- Packages are used for Model Organization
- Package Diagrams are used to depict how the model is organized
- Packages can contain:
  - Other packages
  - Model elements
  - Model diagrams
- Models may be organized using a variety of methods
- Package names are used as the Namespace for the elements that reside in the package
- Views are a type of package that import model elements in order to support a particular viewpoint