# Activity: Architecture Modeling

### Importance

Understanding models and views that can capture an architecture is very important in software architecture design. Model views have to also be consistent among the different views.

### Learning Objectives

- To understand about common notations used for architectural modeling that include the UML and box and line drawings.
- To understand how architectural views are related to each other.

### Success Criteria

- To be able to create both UML and natural language descriptions for some simple architectures.

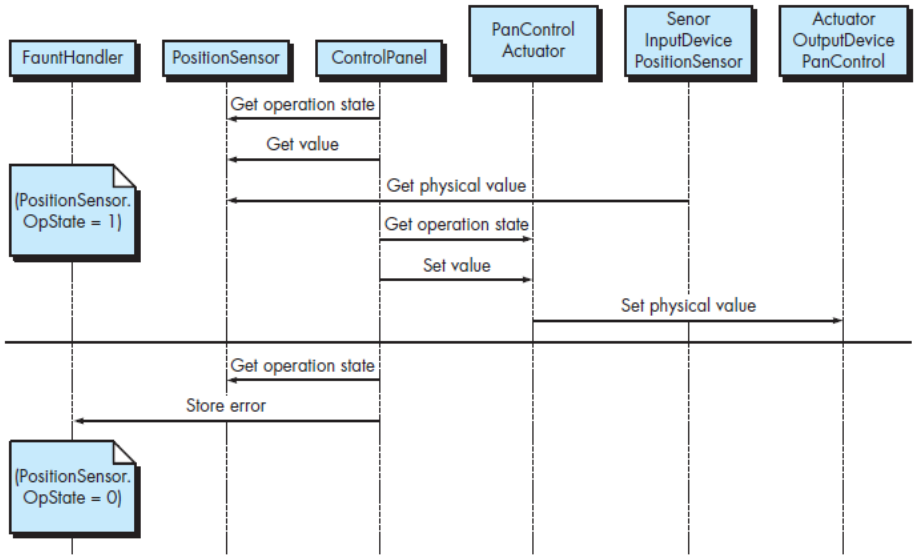
### Resources

- Course notes on Software Architectural models
- Online resources on UML package diagrams such as
    - https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-package-diagram/

### Exercises

Below is a sequence diagram of an Actuator-Sensor pattern realization for a SafeHome application that controls the positioning of a security camera.

1. From this diagram it is possible to define a compatible component diagram, i.e. the components are the entities in the sequence diagram and messages must be sent via interfaces. Draw a component diagram that has a consistent view with the sequence diagram.




2. Read through the cash register requirements posted in Blackboard with this exercise and using the natural language notation write descriptions of the components envisioned in this system. Note: Components are similar to Classes except that they may consist of several classes interacting through a common Interface. As such you can leverage the noun-phrases procedure typically used to extract Classes from a textual description of a scenario, where nouns are used to define Classes. In a similar fashion verbs in scenarios can help identify communication between components.
3. One could argue that creating a component diagram for the Cash Register system is a bit premature and as such a UML Package diagram is more suitable. Use the knowledge acquired from question 2 to create a UML Package representation of the Cash Register system and perceived dependencies.