



UML Overview

Cliff Kettemborough

crk@jpl.nasa.gov

What is UML?

Unified Modeling Language

- **Notation Language for specifying, visualizing, constructing and documenting object-oriented software**
 - Successor to the modeling languages found in Booch, OOSE/Jacobson, OMT and others
 - Integrates the best of each
- **Includes 9 diagrams for different views of a system model**
 - Define a common meta-model that is independent of a particular programming language
 - Selected to facilitate an Object-oriented Analysis and Design (OOAD) process
 - Selected to encourage use-case driven, architecture-centric, and iterative and incremental development process
- **Non-proprietary standard adopted by OMG standard**
 - Developed by Rational
 - Companies who have adopted UML: Dec, HP, IBM, Microsoft, MCI, TI, Unisys
- **Industry support**
 - Products from Rational, iLogix, Artisan, Visual Object Modelers, Object International, etc.
 - UML training and reference materials are widely available
 - UML model interchange is under development by OMG

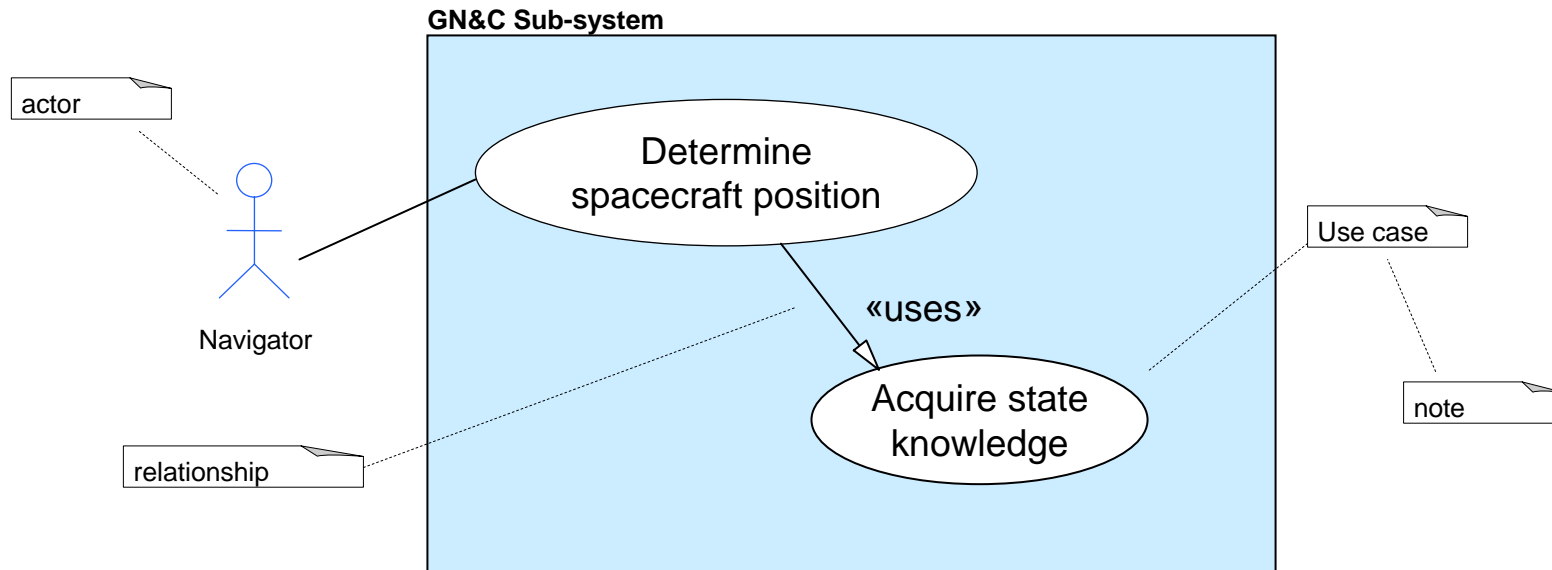
The 9 UML Diagrams

Each diagram captures a different view of the model

Use Case	expresses user requirements and capabilities required from the model
Sequence	captures message flows in a scenario
Collaboration	captures the message flows between collaborating objects in a scenario
Statechart	captures the dynamic behavior in a scenario
Activity	captures the concurrent activities in a scenario
Class	expresses the static structure of the model
Object-Model	expresses relationship between instantiated classes in model
Component	captures the work units and development dependencies in the model
Deployment	captures process allocation of the model

What Is a Use Case?

“A use case is a system function that returns an observable result to an actor without revealing internal system structure” — Bruce Douglass



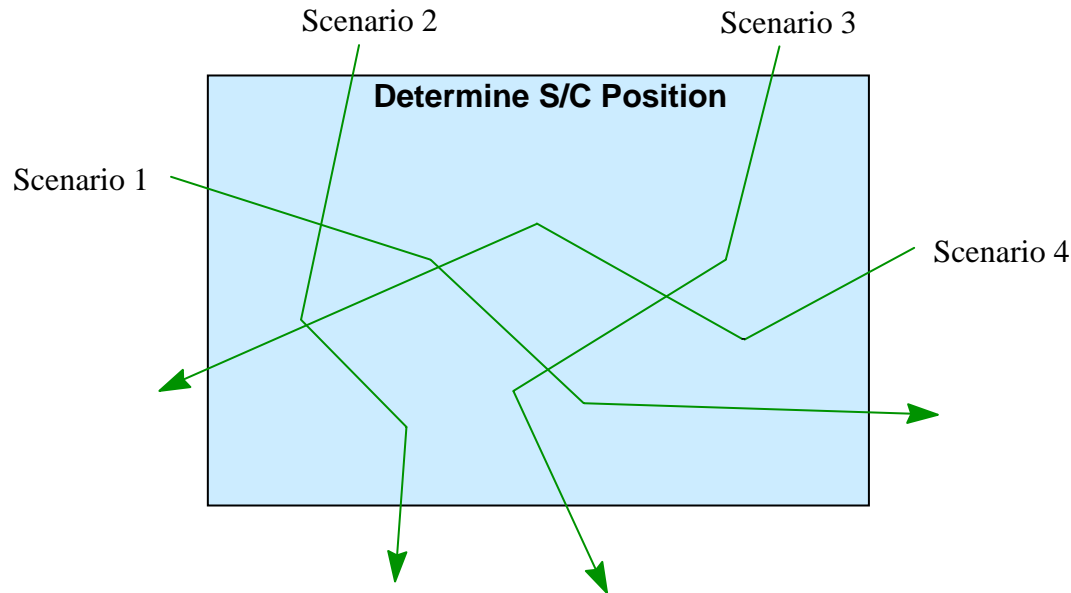
Intent of a use case:

- Describes how a user interacts with the system.
- Expresses requirements in ordinary language.
- Unifies a set of behaviors under a single idea.

A use case does not describe implementation.

Connecting Scenarios to Use Cases

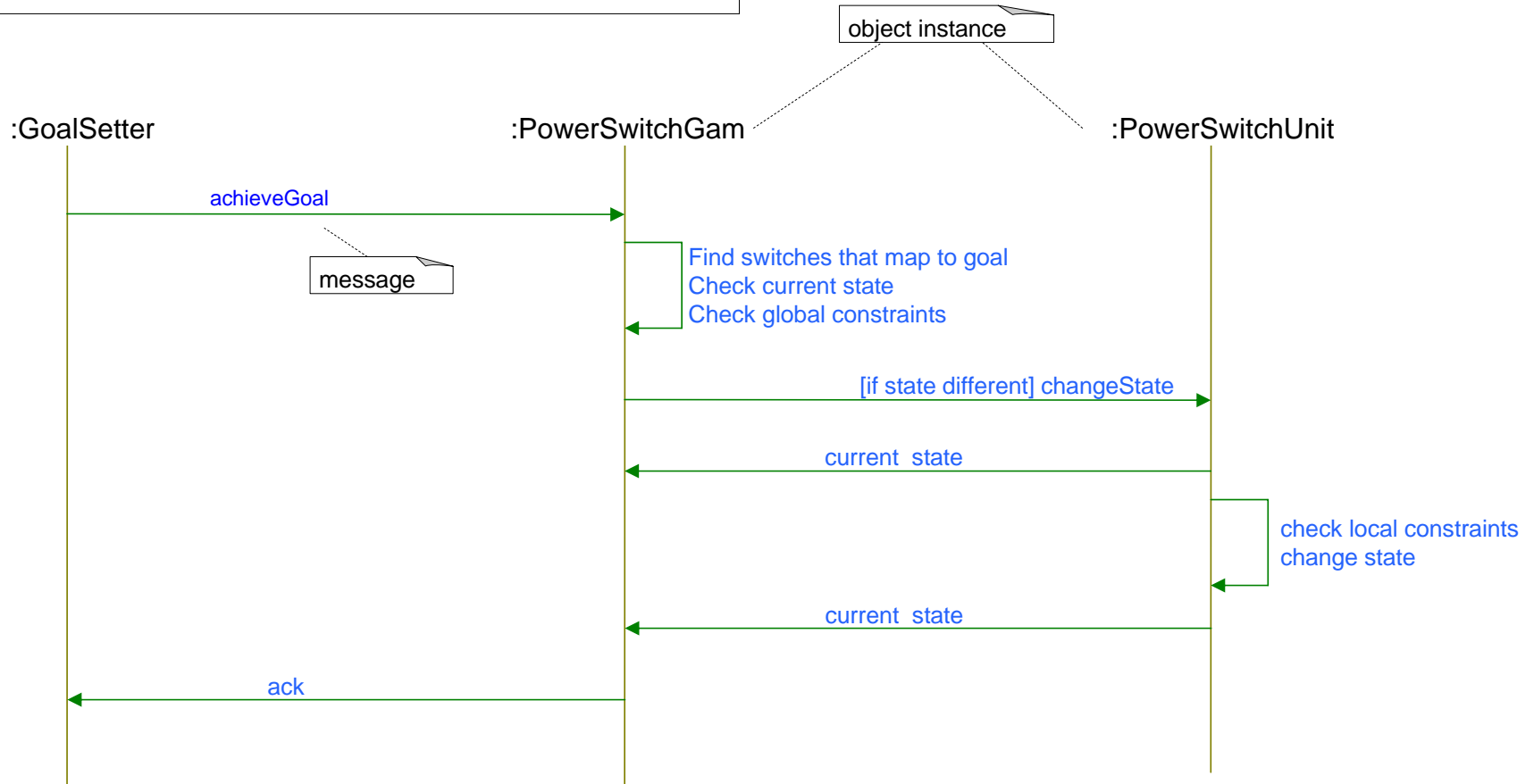
A scenario is a particular path through the abstract and general description provided by the use case. — Pierre-Alain Muller



A **use case** describes an abstract function of the system.
 A **scenario** describes a specific implementation of a use case.

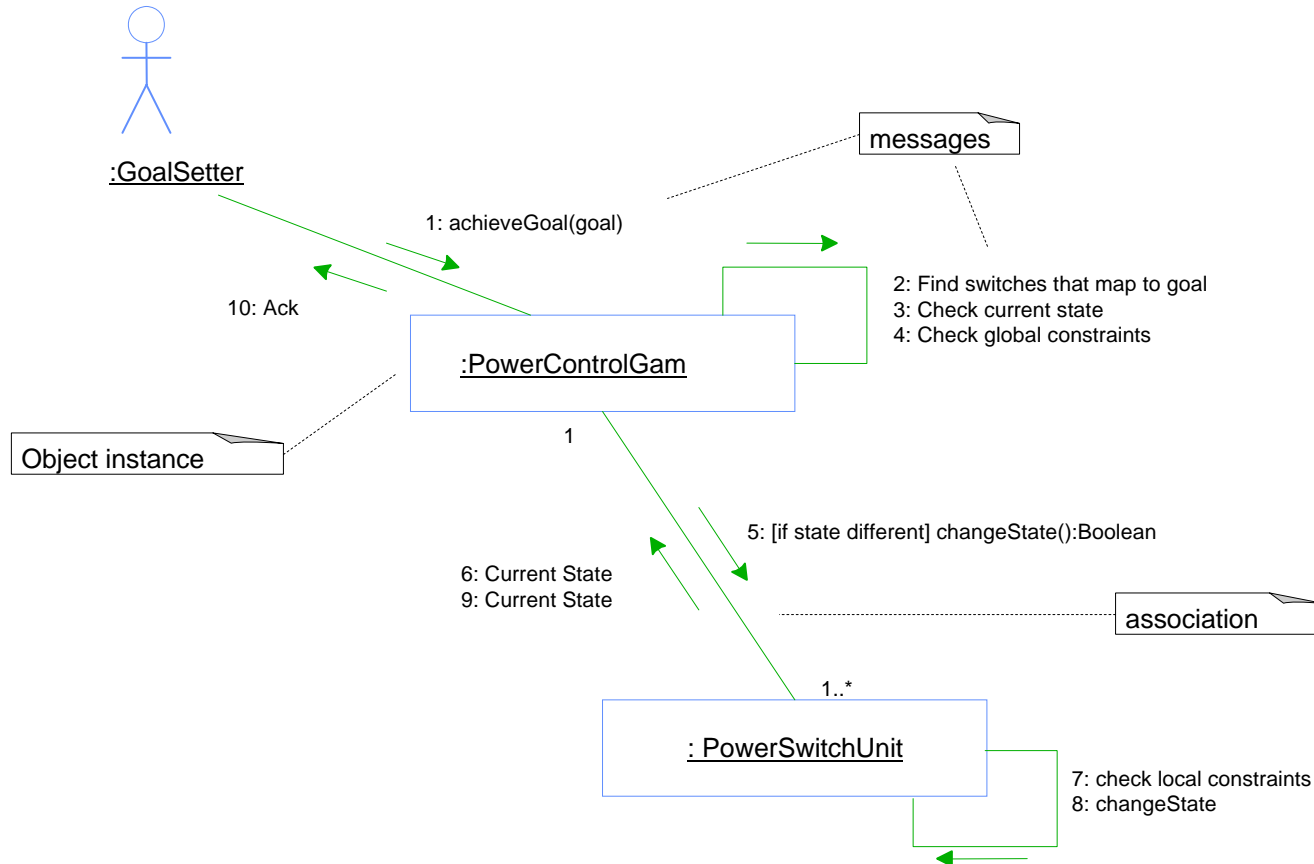
Example Sequence Diagram

Context: Power control
Use Case: Set a goal on power
Scenario: PowerSwitchGam nominal goal setting

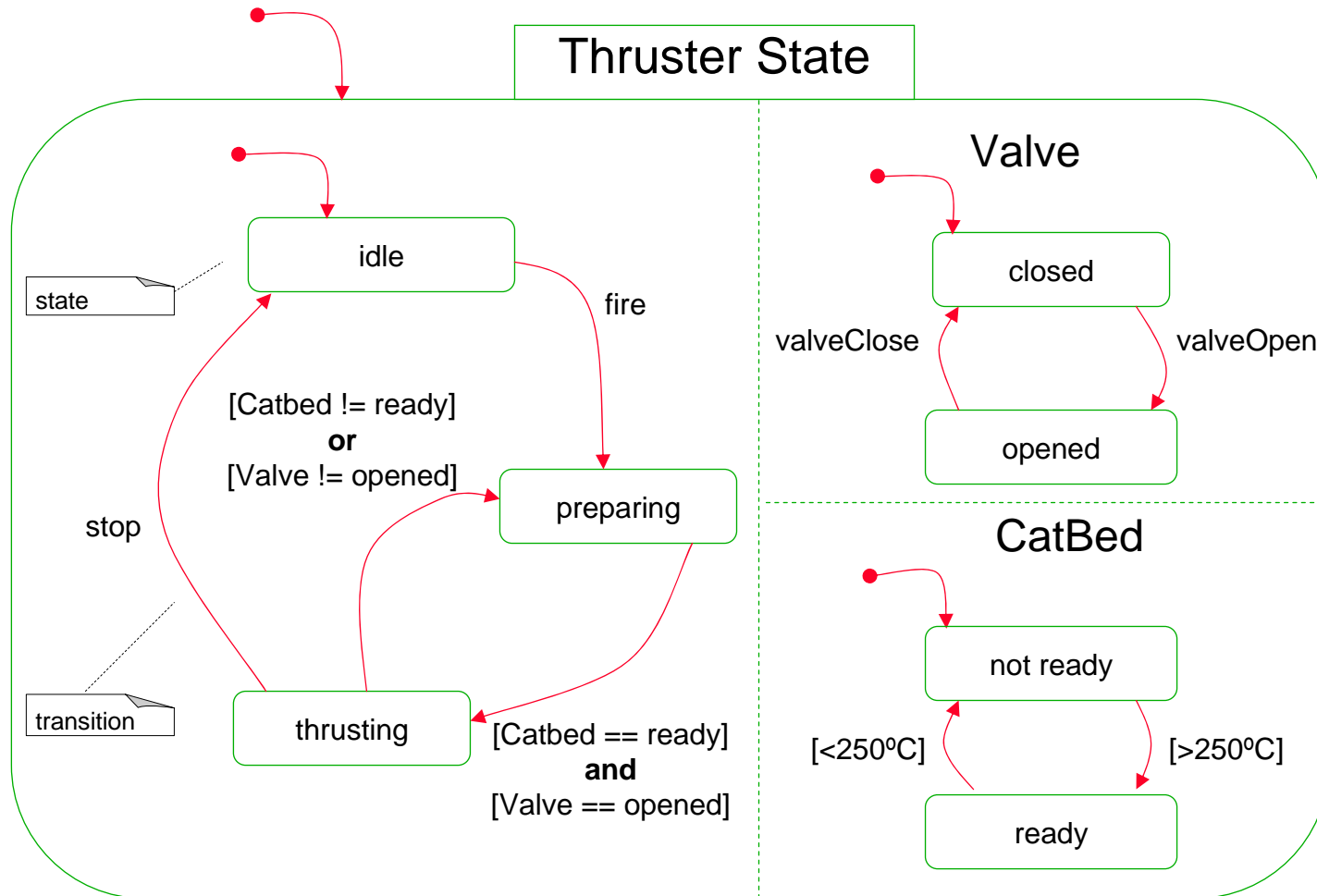


Example Collaboration Diagram

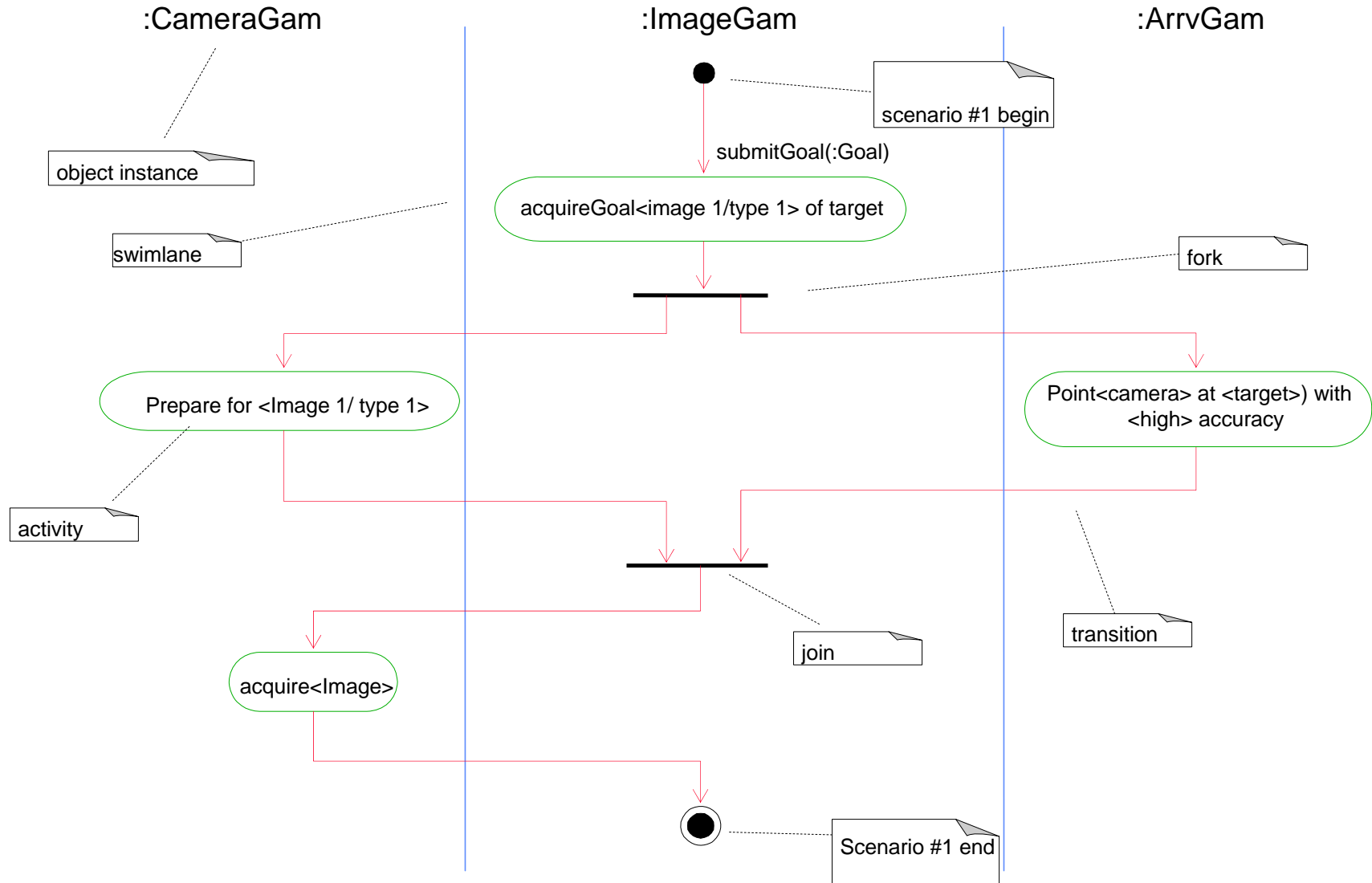
Context: Power control
Use Case: Set a goal on power
Scenario: PowerSwitchGam nominal goal setting



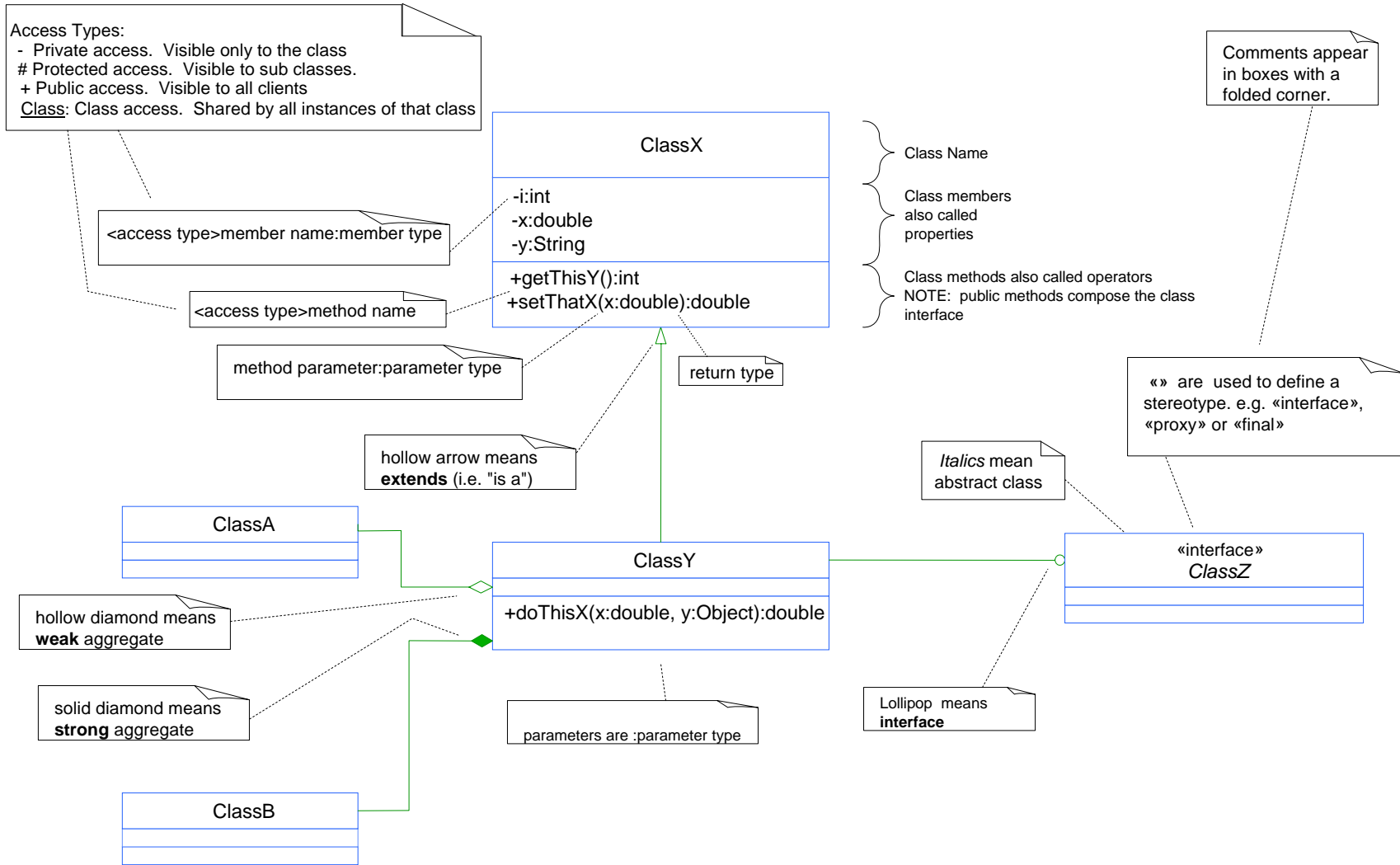
Example Statechart



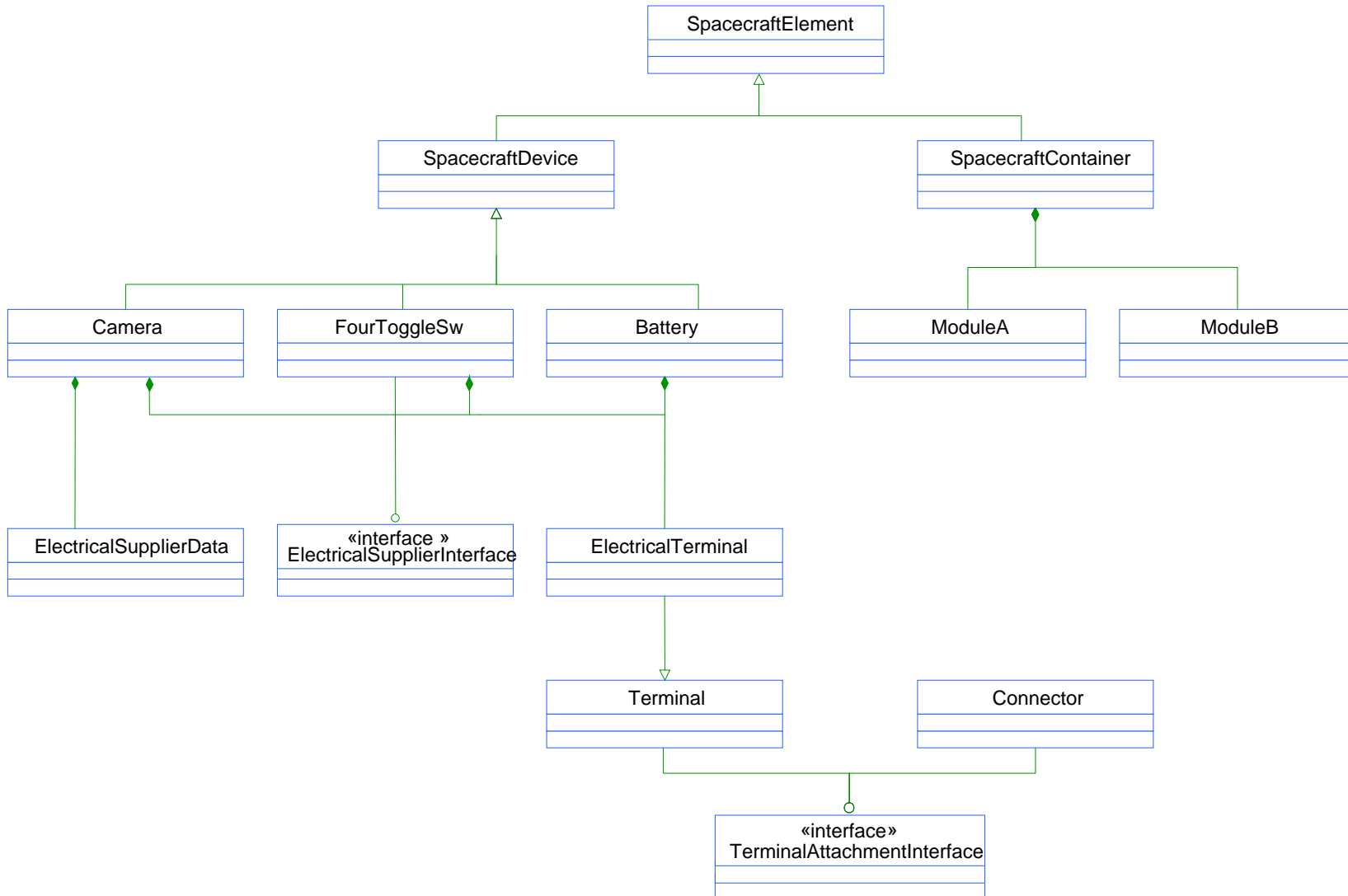
Example Activity Diagram



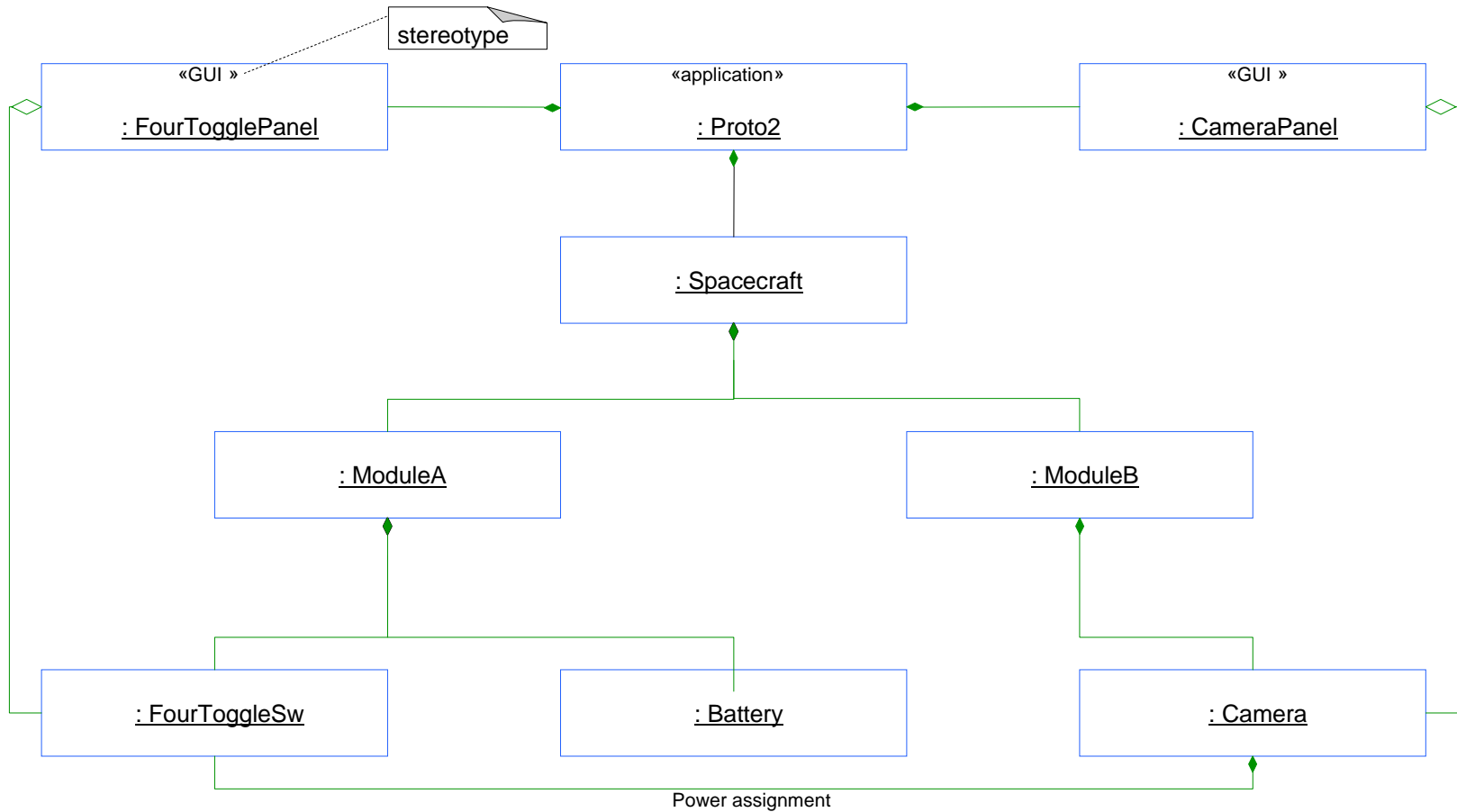
Reference Class Diagram



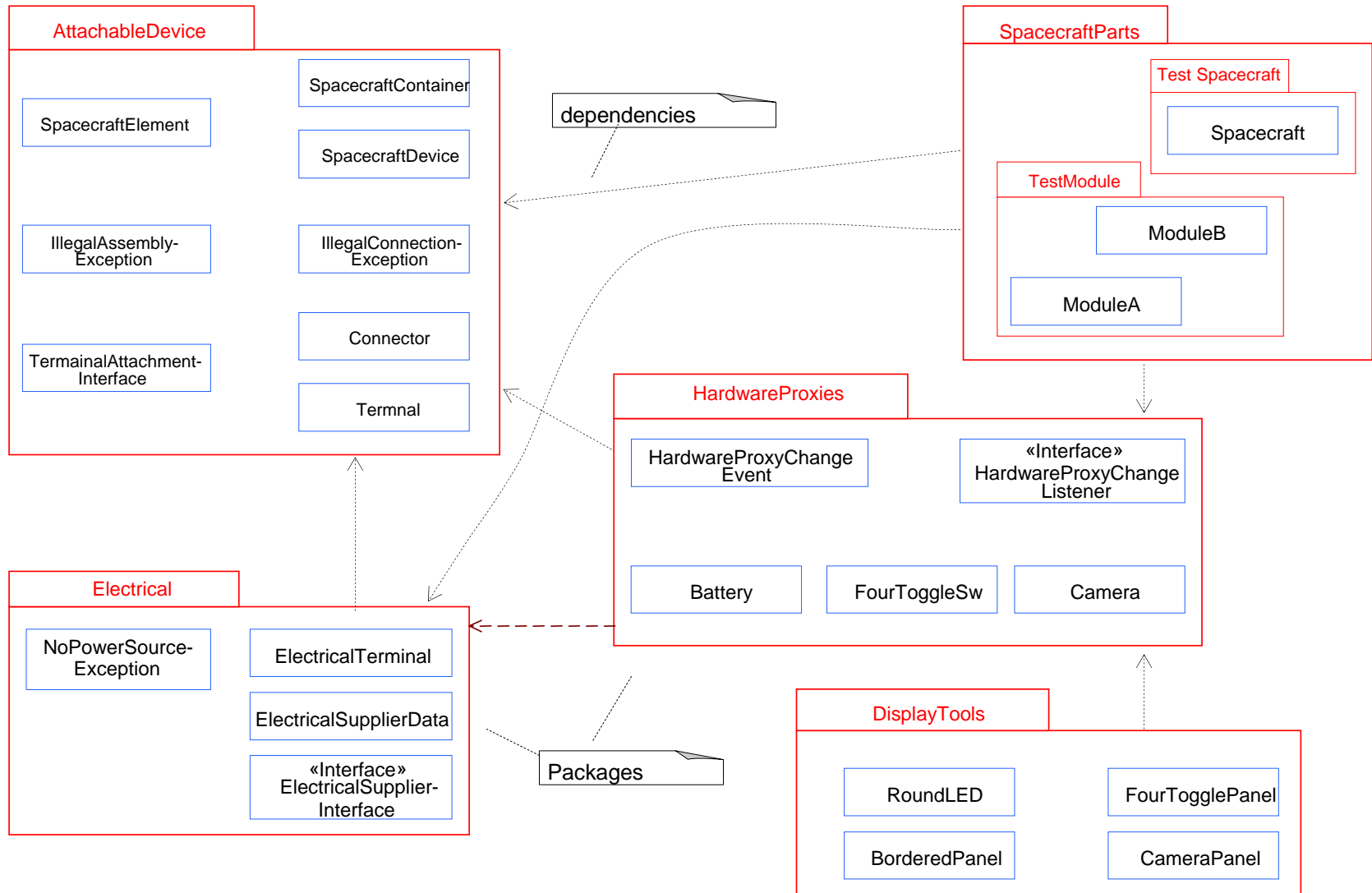
Example Class Diagram



Example Object Model Diagram

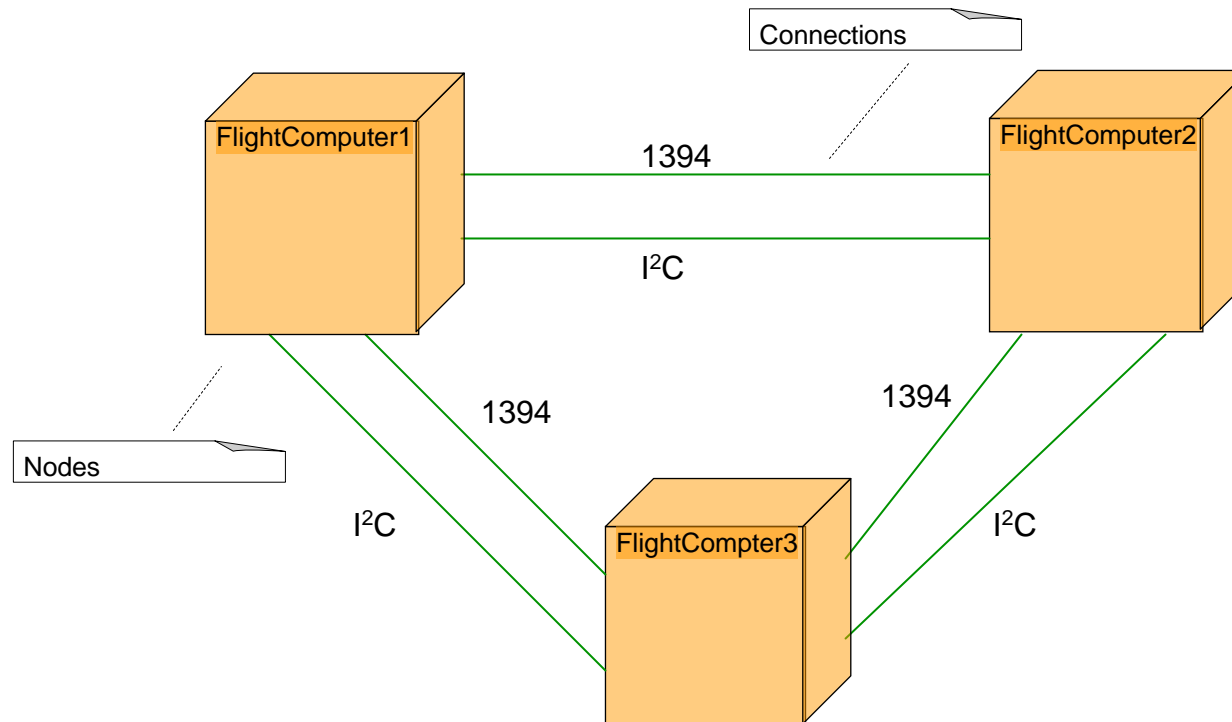


Example Package Diagram with Dependencies



Example Deployment Diagram

Deployment Diagram



Using UML for Test

- Assignments to implementation teams include a scenario — scenarios define a verification test.
- Since scenarios include requirement specifications (as constraints) — scenarios define validation tests.
- Since use cases define functionality that arise from a collaboration of objects — use cases define end-to-end tests.
- Since use cases capture the entire functionality of the system — use cases are used to determine when the project is complete.

UML Roadmap

Stage 1: No UML/OOAD modeling tool (i.e Visio or Canvas)

- Capture design independent of implementation in a well-defined common notation

Stage 2: Use UML/OOAD tool

- Capture design independent of implementation in a well-defined common notation
- Rapid prototyping and code generation
- Round-trip of tool generated code

Stage 3: Next generation UML/OOAD tool

- Capture design independent of implementation in a well-defined common notation
- Rapid prototyping and code generation
- Round-trip of tool arbitrary code
- Accurate documentation of implemented code
- Interoperability between tools

UML/OOAD tools candidates discussed in Development infrastructure section

Summary of UML uses

- **Requirements capture**
 - Identify and describe system function with use cases
 - Elaborate use cases with scenarios
 - Document scenarios as sequence, collaboration, activity or statechart diagrams
- **Testing**
 - Verify implementation with use cases and scenarios
- **Inter-team communication**
 - Describe architectural design patterns
 - Describe project assignments
 - Describe planned implementation in design walkthroughs
 - Describe implementation in completion reviews
 - Describe test scenarios for verification and validation
- **Documentation**
 - Capture details of design for developers from projects
 - Capture process for ISO compliance