Systems Engineering

What is Systems Engineering?

- Systems Engineering = definition, specification, and high-level architecture of a system which is to be realized with multiple disciplines, typically including electronics, mechanical, software, and possible chemical engineering.

- Primary activities:
  - Capturing, specifying and validating the requirements of the system as a whole
  - Specification of the high-level subsystem architecture
  - Definition of the subsystem interfaces and functionality
  - Mapping the system requirements onto the various subsystems
  - Decomposing the subsystems into the various disciplines – electronic, mechanical, software and chemical – and defining the abstract interfaces between those aspects

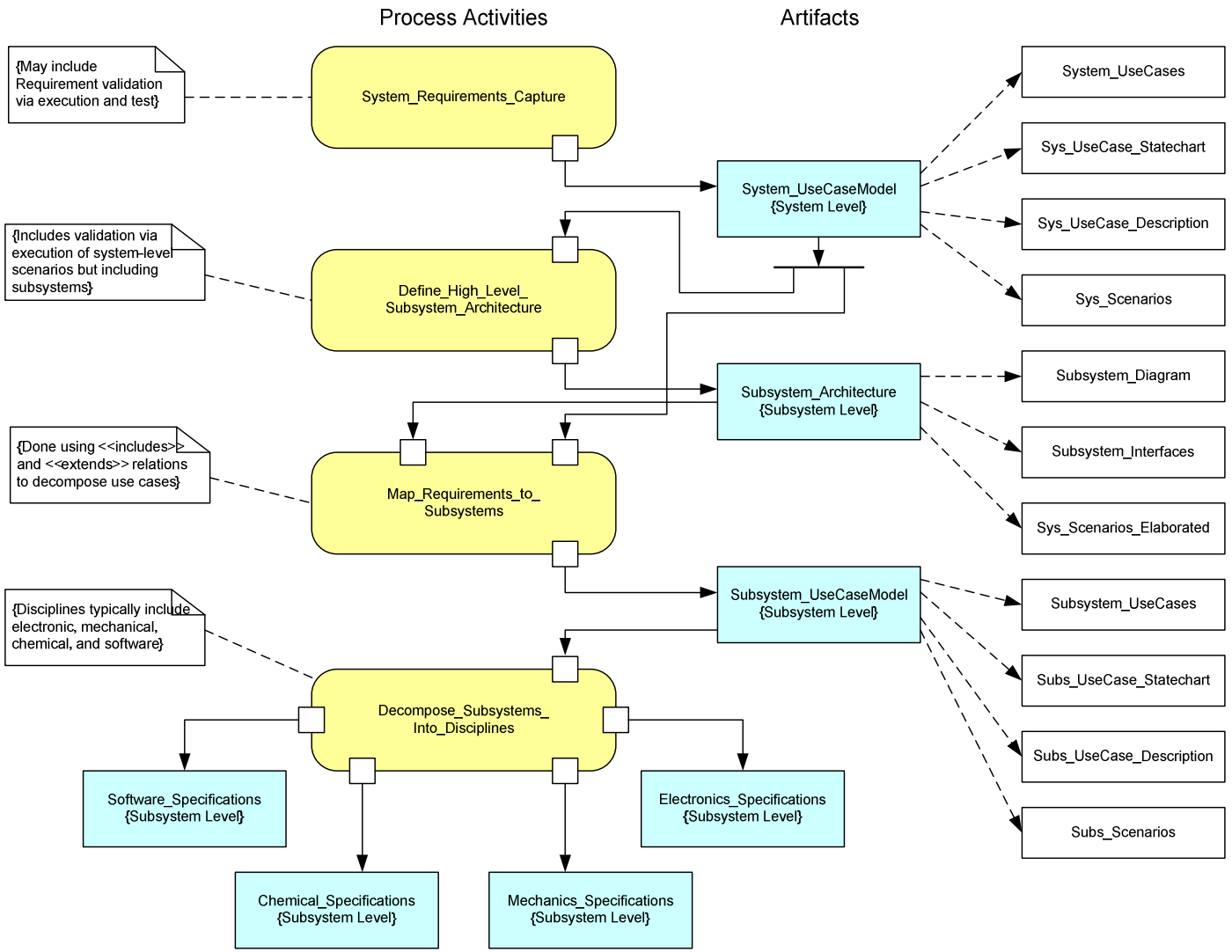
Role of the systems engineer

- In all of these activities, the systems engineers are not concerned with the design of the discipline-specific aspects of the software or the electronics, but ARE concerned with the specification of what those design aspects must achieve and how they will collaborate.

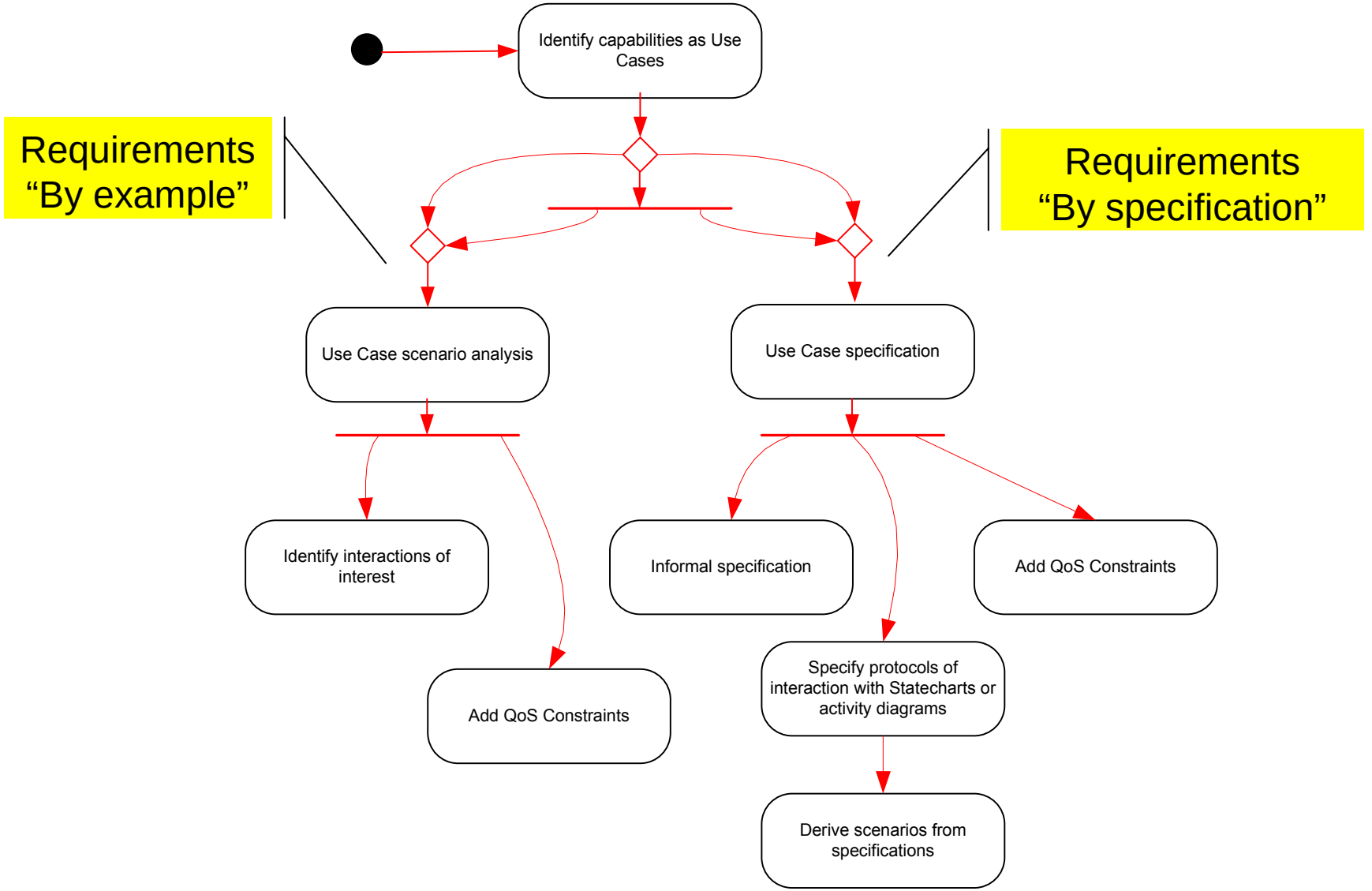
Systems Engineering

According to B.P. DOUGLASS

Systems Engineering Workflow Overview



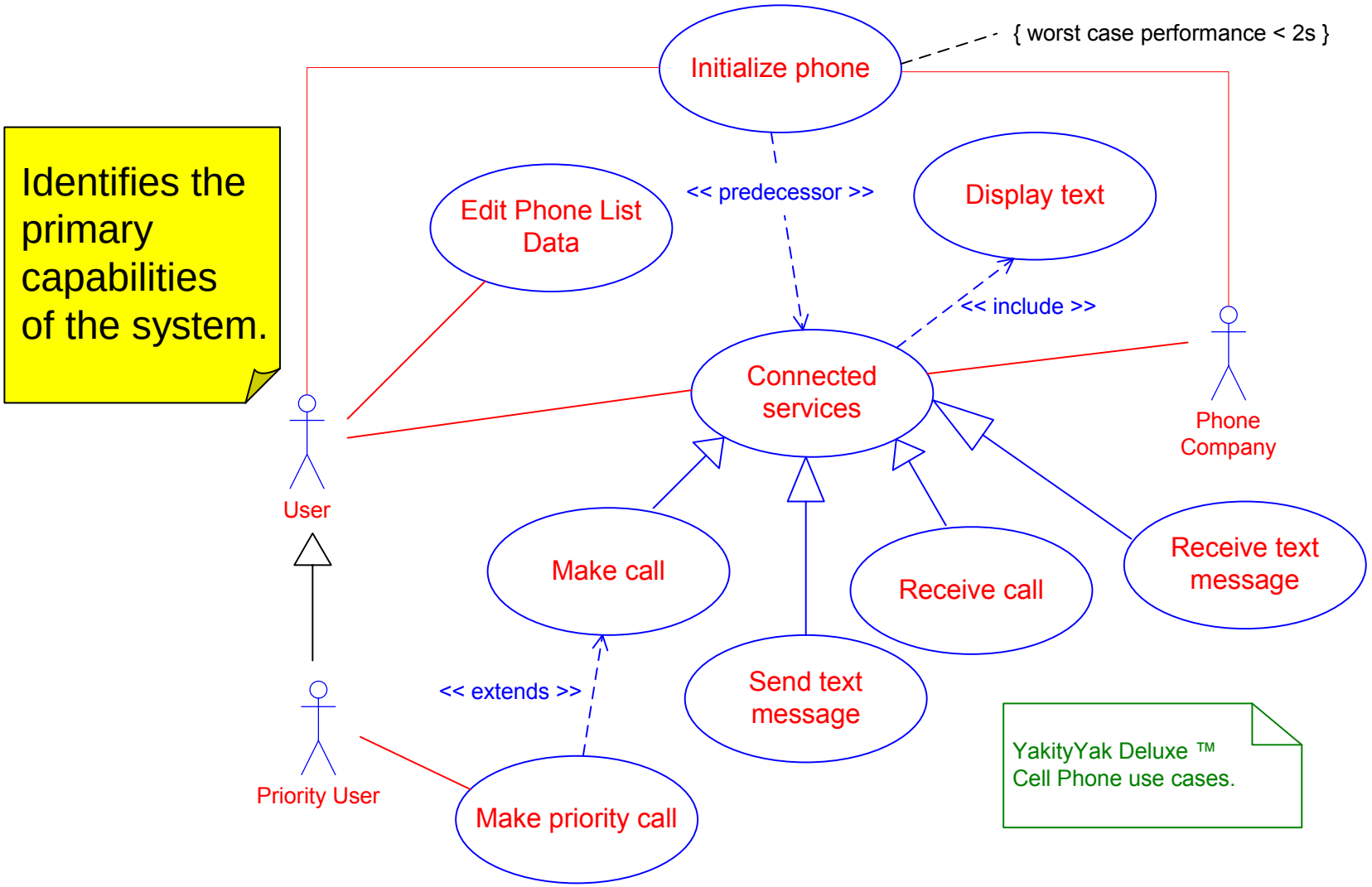
System Requirements Capture



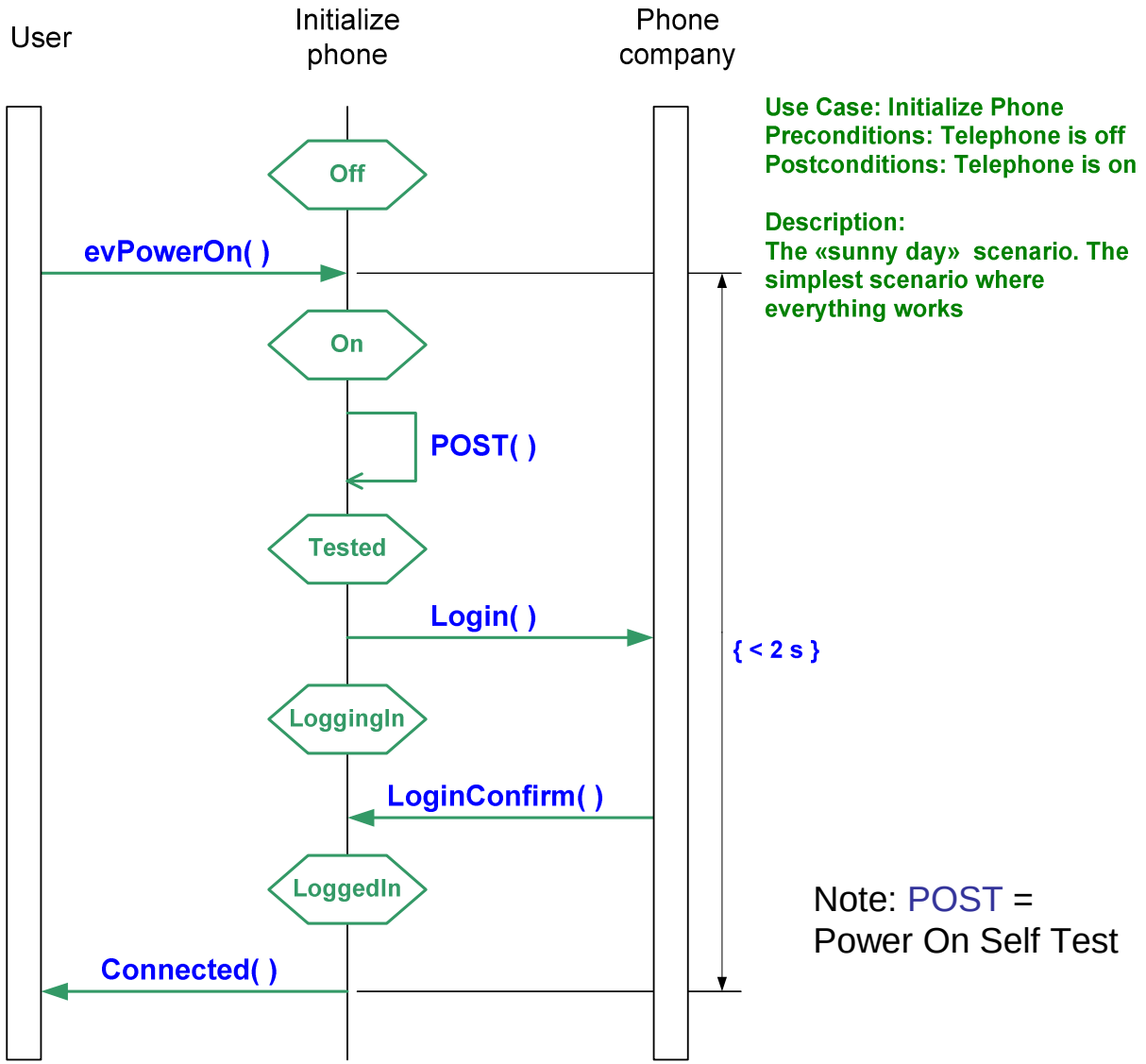
Use Cases

- A use case usually provides a textual description outlining its intent and purpose.
- Use Case Description Format:
  - Name
  - Purpose
    - May be written informally (“The purpose of the capability is to…”)
  - Description
    - May be written semi-formally (“The system shall…”)
    - May be informal
    - May be a hyperlink off to a separate document
  - Preconditions
    - What is true *prior* to the execution of the capability?
  - Postconditions
    - What does the system guarantee to be true *after* the execution of the use case?
  - Constraints
    - Additional QoS requirements or other rules for the use case

Use Case Diagram

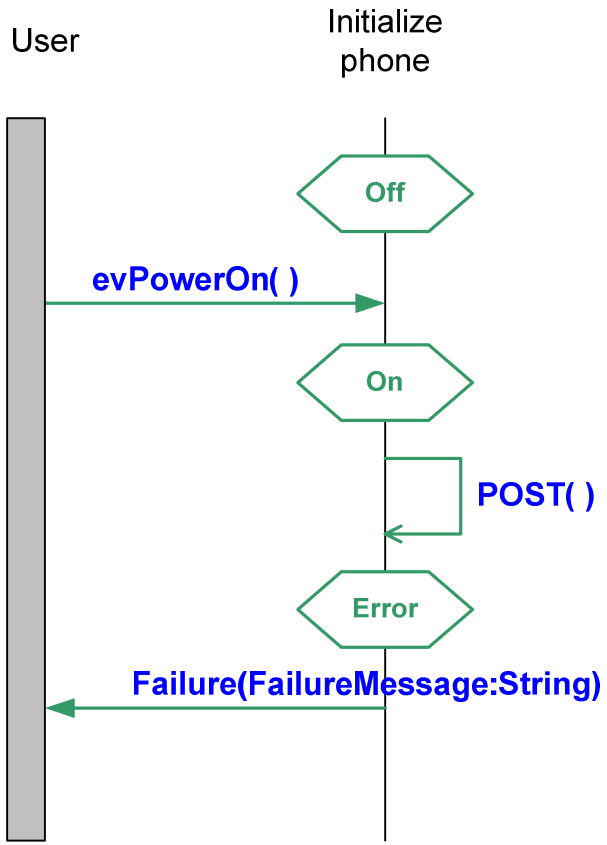


Initialize phone scenario



Note: POST = Power On Self Test

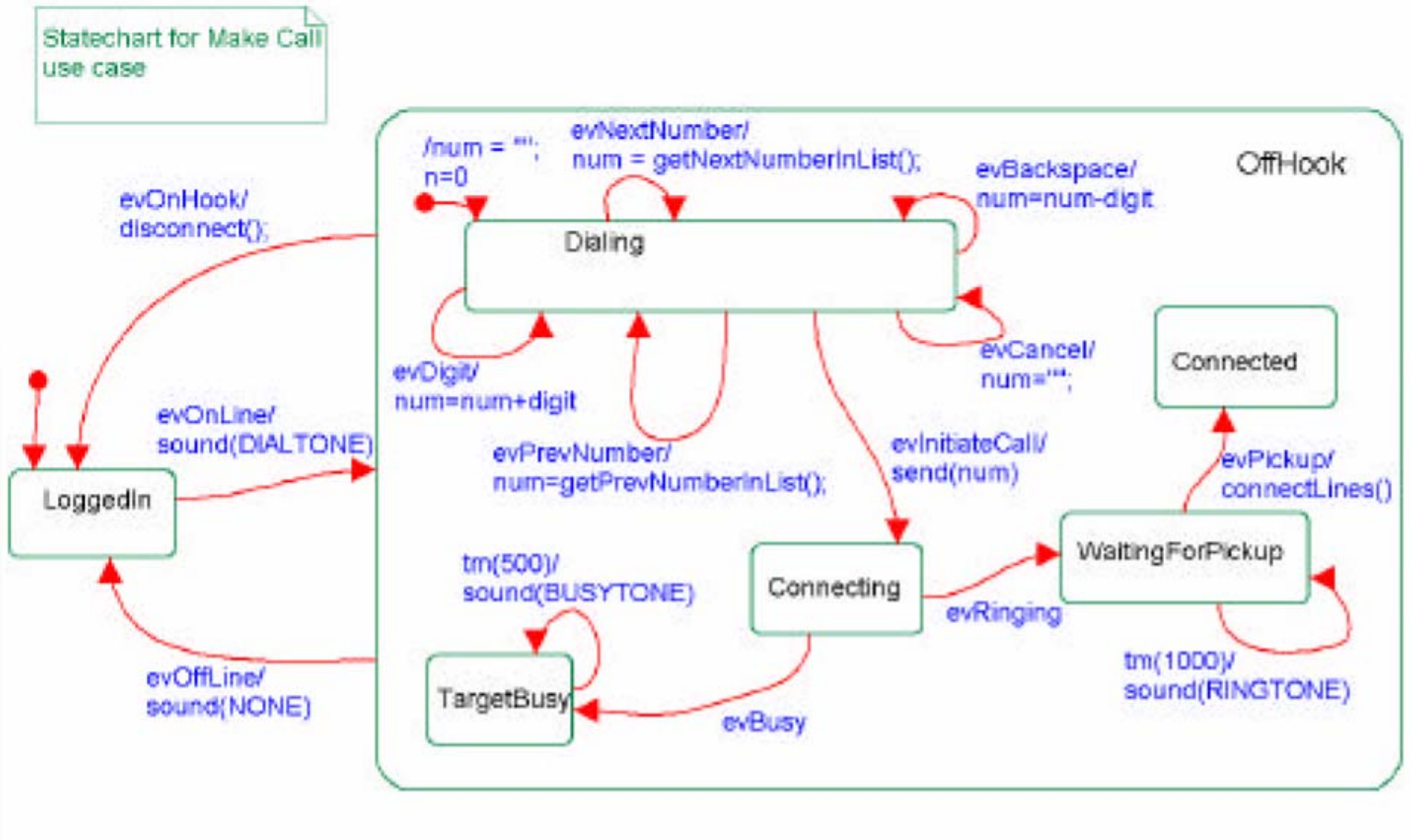
Initialize phone scenario (2)



Use Case: Initialize Phone
Preconditions: Telephone is off
Postconditions: Telephone is on

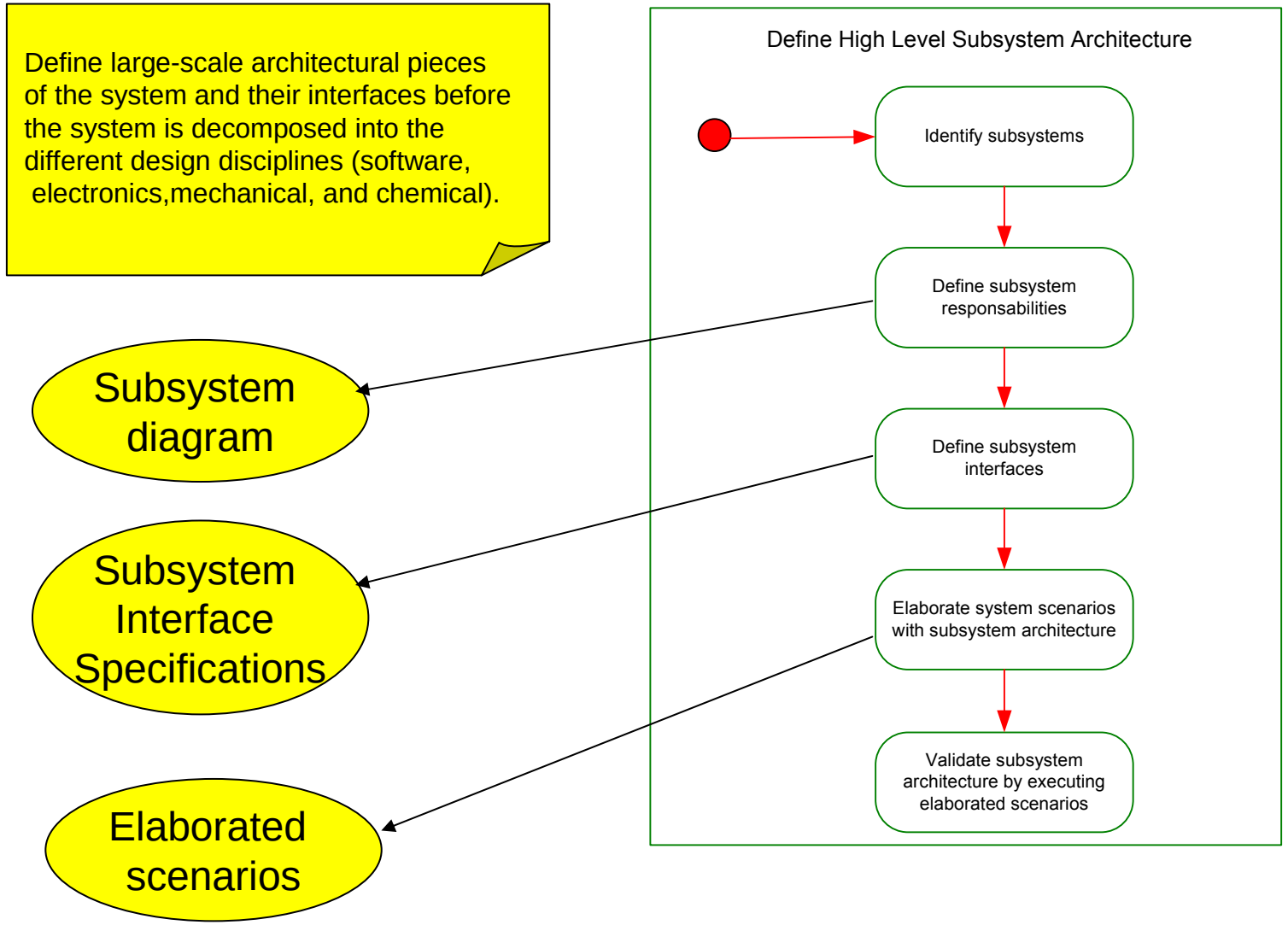
Description:
POST fails. Error message is displayed on the phone display

Make Call Use Case Statechart

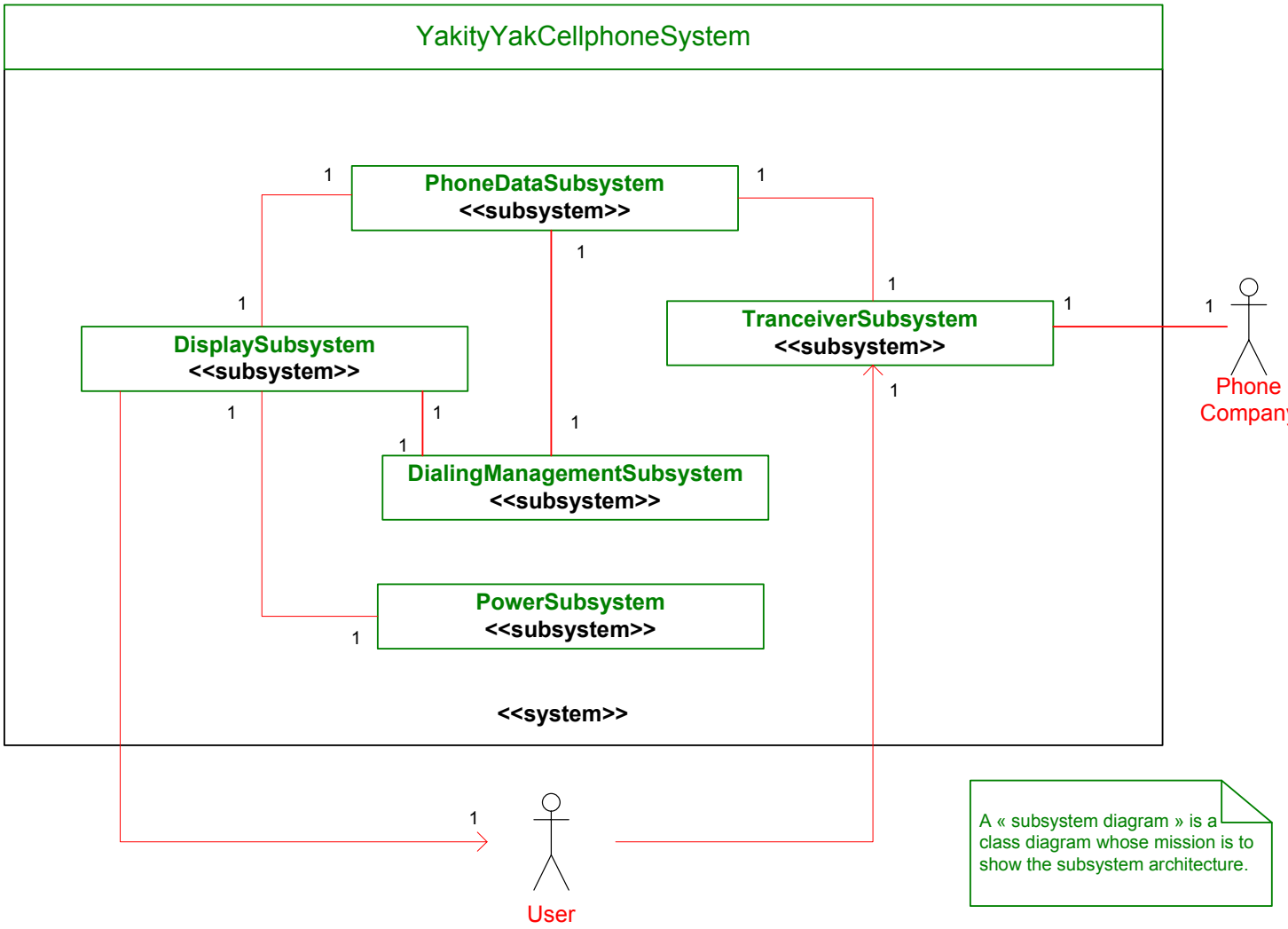


System architecture workflow

Define large-scale architectural pieces of the system and their interfaces before the system is decomposed into the different design disciplines (software, electronics,mechanical, and chemical).



Cellphone subsystem diagram

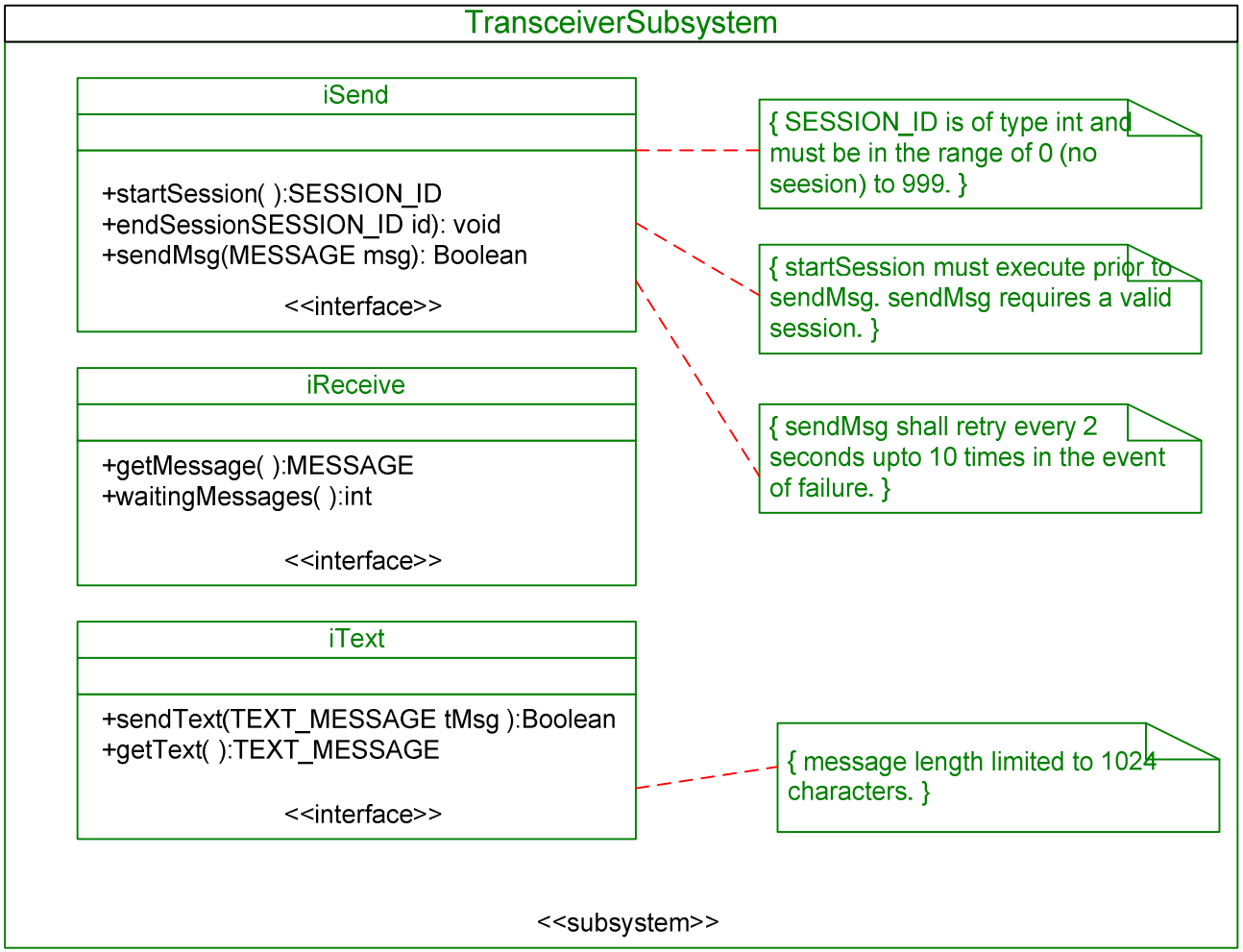


Interface Specification

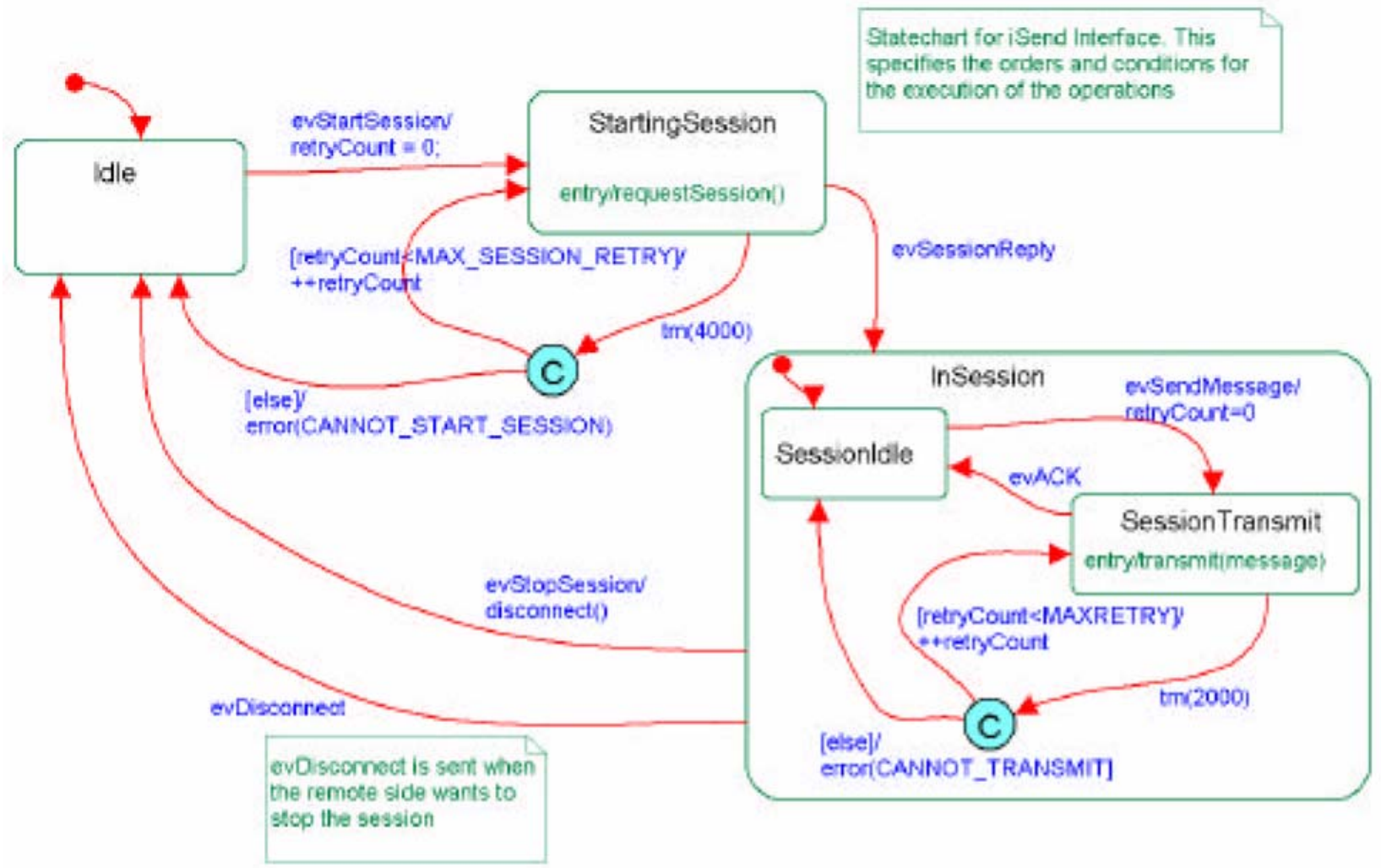
Subsystem Interface Specification Description Format:

- Operation Name and parameter list
- Parameter Description (for each parameter)
  - type
  - range
- Description
- Preconditions
  - What is true *prior* to the execution of the operation?
- Postconditions
  - What does the system guarantee to be true *after* the execution of the operation?
- Constraints
  - Additional QoS requirements or other rules for the operation (including performance requirements, if appropriate)

Transceiver interface



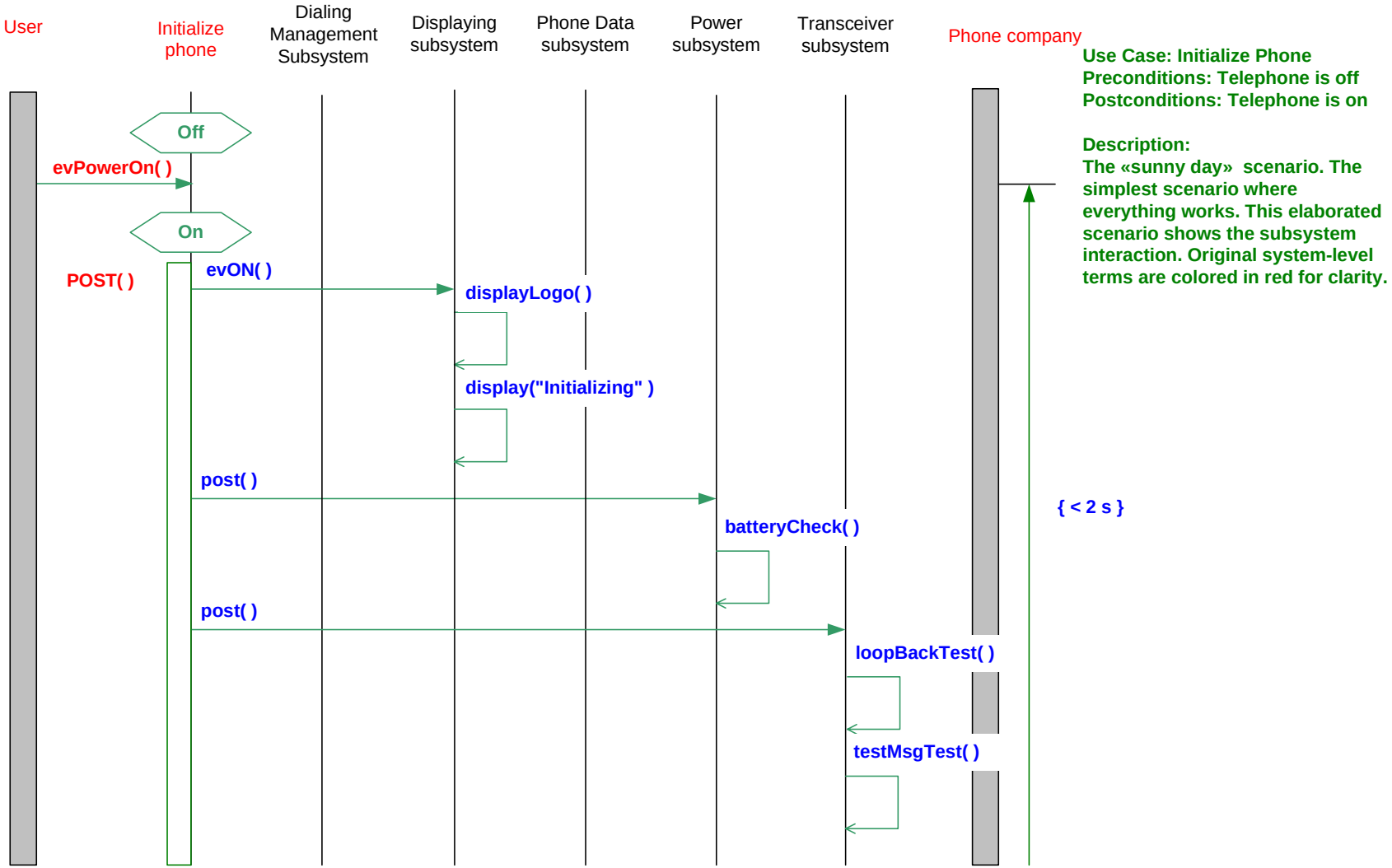
iSend interface statechart



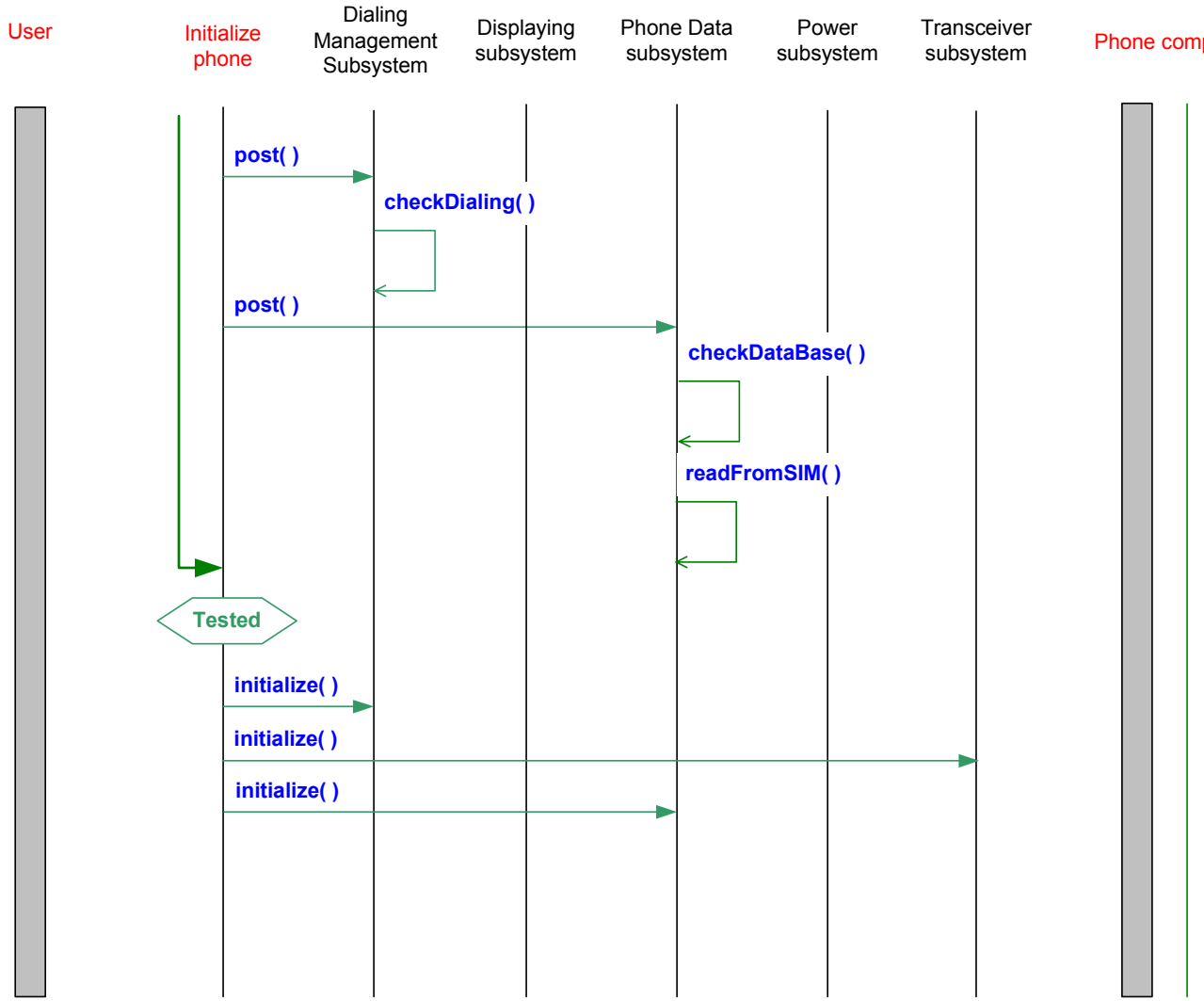
Architecture validation

- Scenarios capture example traces of execution of the system. The UML sequence diagrams are the most popular way to capture scenarios. As we create a subsystem architecture and define the interfaces among this architectural components, we must ask ourselves if we have done it properly. The *only* way to truly answer that question is to *validate* the architecture via execution, and that is exactly what scenarios allow you to do.
- The point of a subsystem architecture, ultimately, is the efficient realization of the system requirements, so the validation of the architecture must take the requirements into account.

Elaborated Scenario (1)



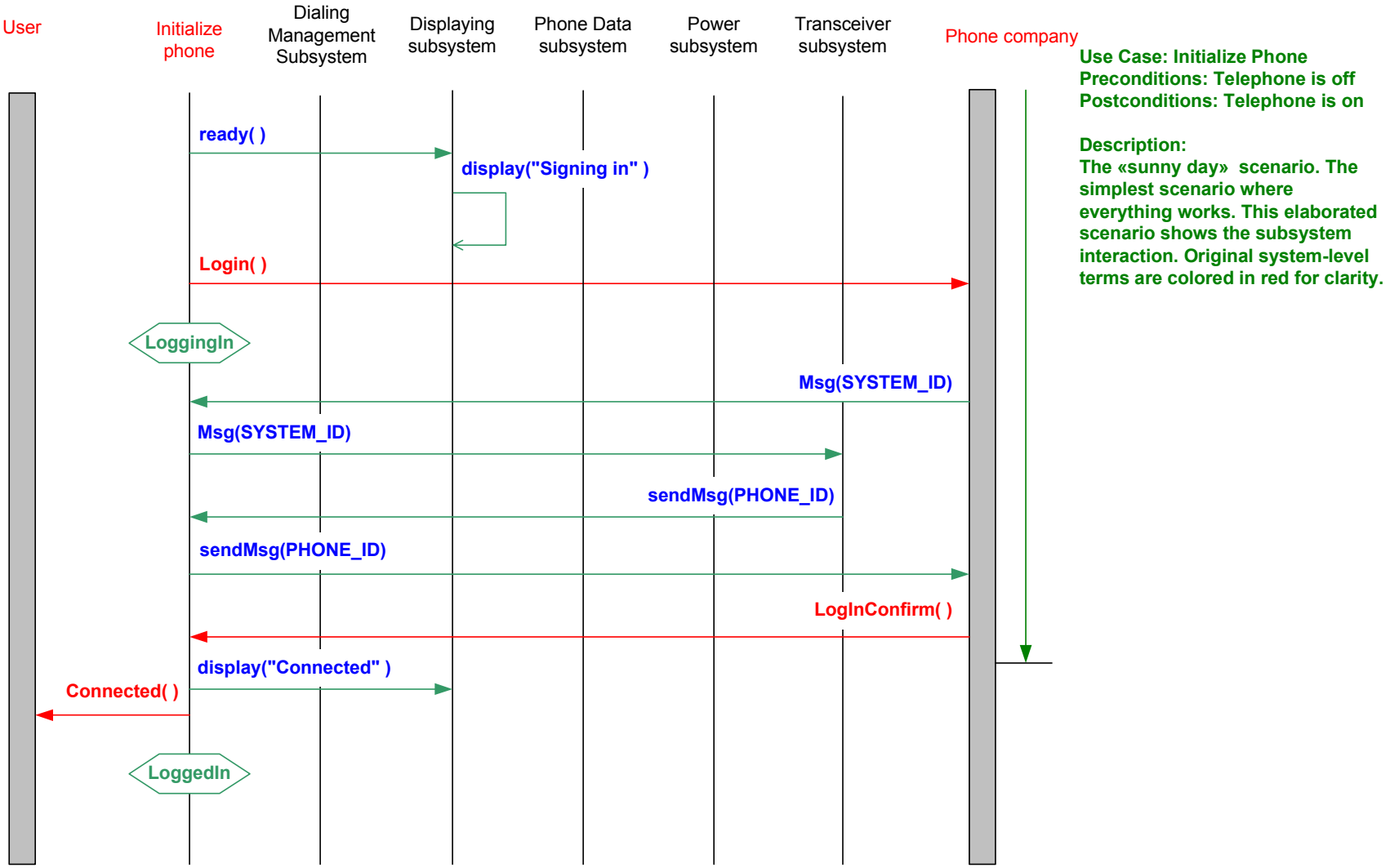
Elaborated Scenario (2)



Use Case: Initialize Phone
Preconditions: Telephone is off
Postconditions: Telephone is on

Description:
The «sunny day» scenario. The simplest scenario where everything works. This elaborated scenario shows the subsystem interaction. Original system-level terms are colored in red for clarity.

Elaborated Scenario (3)



Use Case: Initialize Phone
Preconditions: Telephone is off
Postconditions: Telephone is on

Description:
The «sunny day» scenario. The simplest scenario where everything works. This elaborated scenario shows the subsystem interaction. Original system-level terms are colored in red for clarity.

Mapping requirements to subsystems

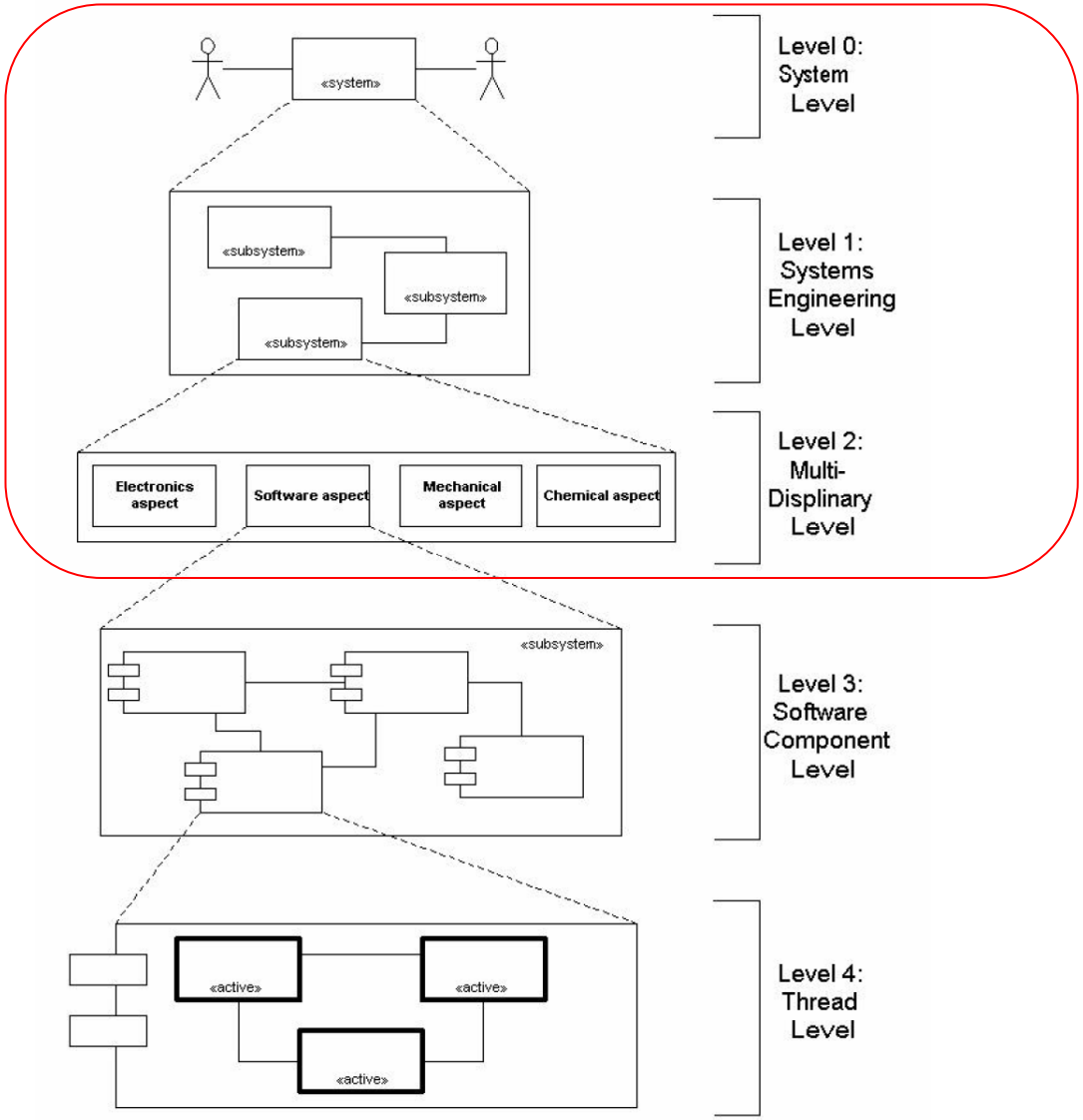
- **Goal**: Provide the subsystem development teams with detailed requirements specific to their subsystems, and a well-defined means for plugging their subsystem into the overall system architecture in such a way that the completed system meets all its functional and QoS requirements.
- **Primary Artifacts**: For each subsystem the following is producted: a Subsystem Use Case Model, consisting a set of use cases (and associated actors), each of which identifies or defines
  - Use Case Description
  - Use Case statechart or activity diagram (optional)
  - Use Case Scenarios

Mapping requirements

UML Diagrams and Model Elements Utilized:

- Use case Diagram, containing
  - Use Cases
  - Actor (including “internal actors,” i.e. peer subsystems)
  - Relations among model elements
    - Associations between actors and use cases
    - Generalization of actors and use cases
    - Dependency among use cases
  - Constraints
- Text Description
  - May be done in Description field associated with Use case
  - May be done in a separate documentation tool (e.g. word)
- Statechart associated with a single subsystem-level use case
- Activity diagram associated with a subsystem-level single use case
- Sequence diagram, each depicting a single scenario of a subsystem-level single use case

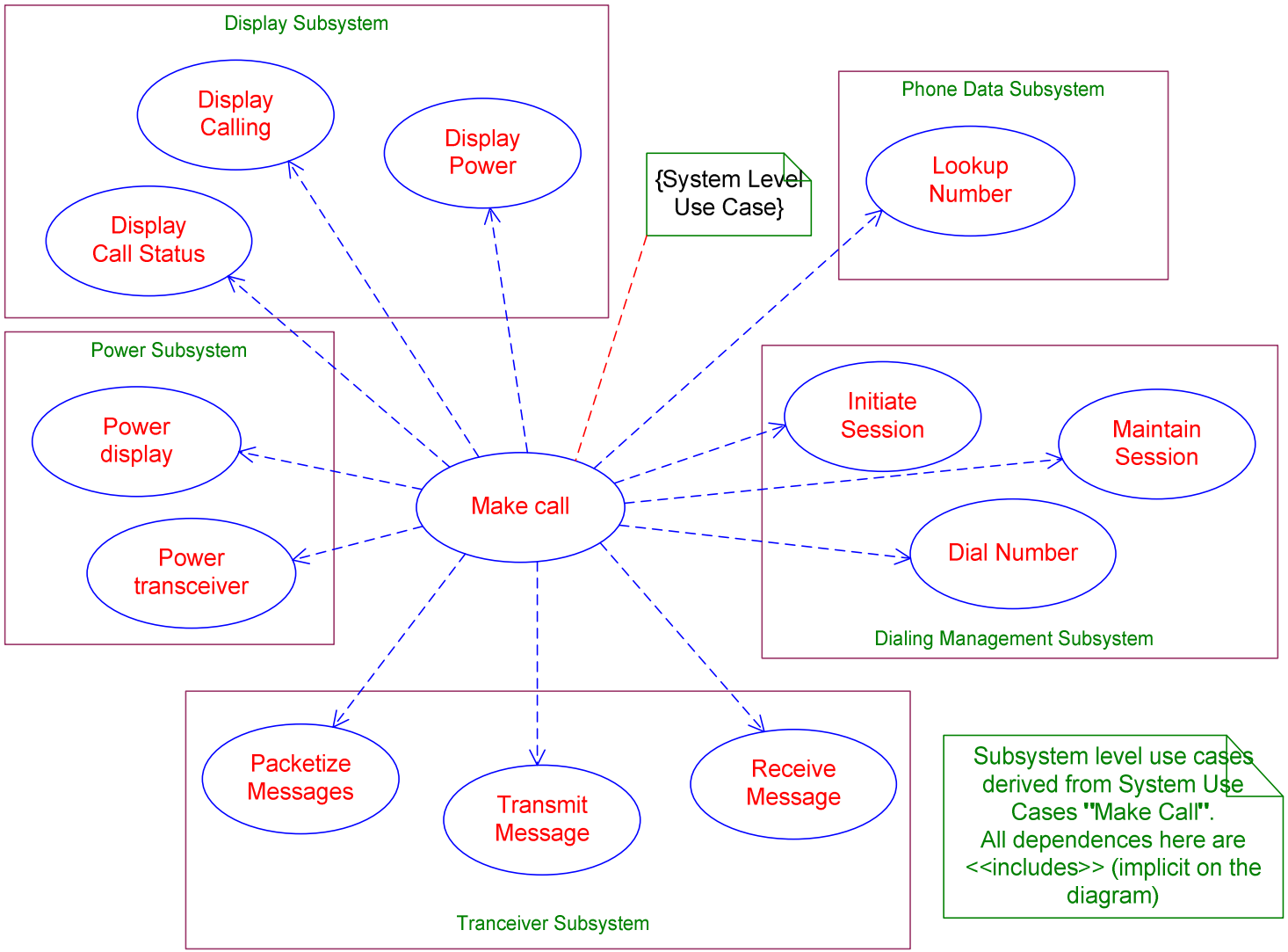
Hierarchical architecture model



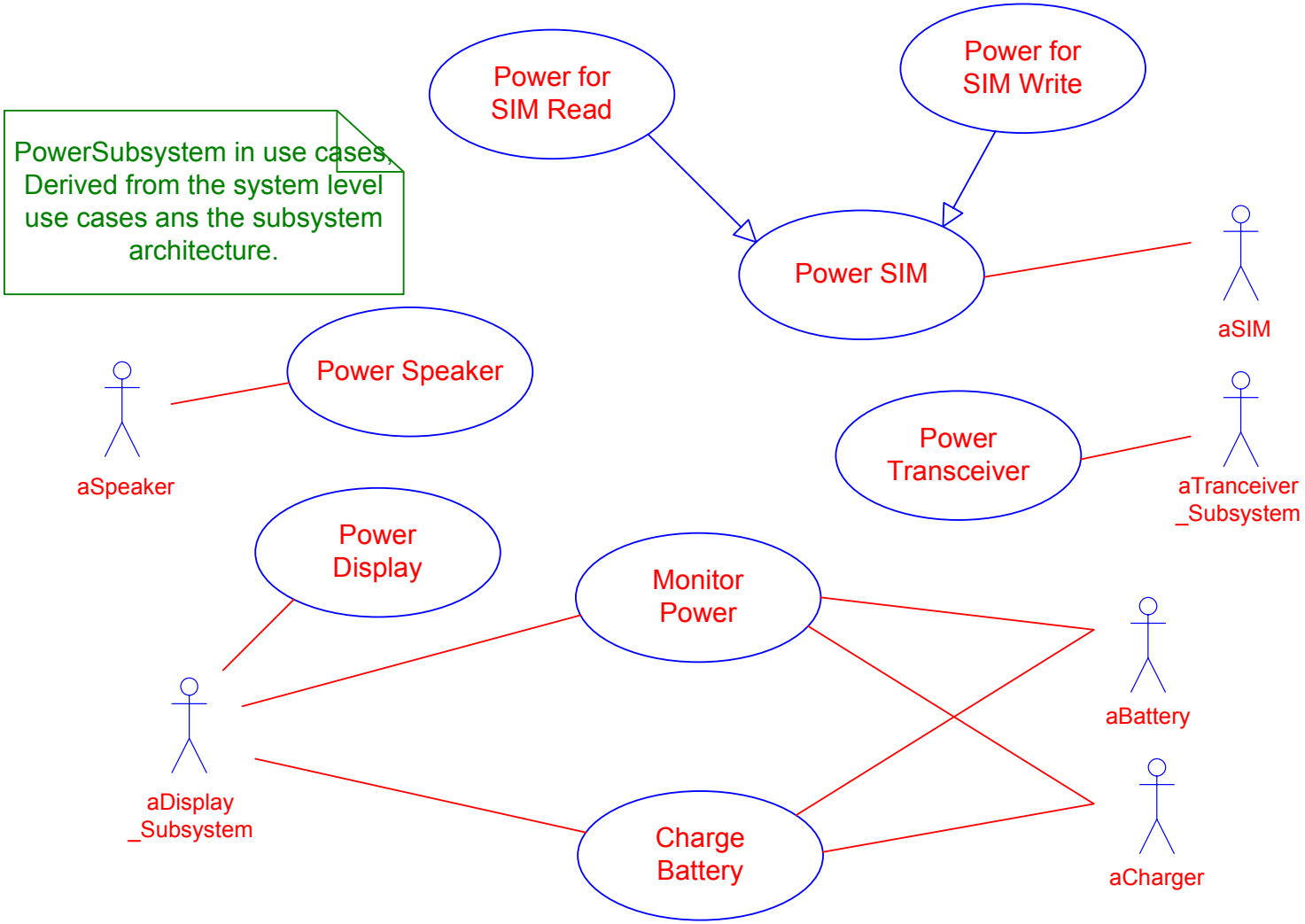
Systems engineer's concerns

Discipline experts

Subsystem level Use Cases



Subsystem Use Cases



Discipline decomposition

- **Goal**: Decompose the internal structure of a subsystem into the various disciplines, define the interfaces among those design aspects, and map the requirements to those disciplines.
- **Primary Artifacts**: Subsystem Deployment Design
- **UML Diagrams and Model Elements Utilized**:
  - Deployment Diagram
  - Component Diagram

Deployment architecture