USC Viterbi
School of Engineering

Model-Based System Design of a Space Mission Communication System Using SysML

Document Details

Author: Anish Kulkarni

Course: SAE 547 - Model-based Systems Engineering and Architecture (Elective)

Instructor: Prof. Mark McKelvin

Degree: Master of Science in Product Development Engineering

University: University of Southern California

Date: December 8th, 2024

Executive Summary

This project applies Model-Based Systems Engineering (MBSE) principles using SysML in IBM Rhapsody to design a space mission communication system. The objective is to achieve traceability between requirements and system behavior, enabling efficient system modeling and validation.

The project includes:

- Use case models, block definition diagrams, and activity diagrams for system architecture.
- Finite state machine simulations to develop a digital twin of the system.
- Process optimization techniques using BPMN ontology to streamline system operations.

This work **demonstrates the power of MBSE in aligning engineering workflows with product development goals** and showcases how structured modeling techniques can enhance system efficiency in real-world applications.

Table of Contents

Table of Figures

Project Requirements (Deliverables)

As part of the **SAE 547 Fall 2024** course, this project follows the given specifications to model a **Space Mission Communication System** using **SysML in IBM Rhapsody**. Below are the professor's requirements and how they are fulfilled in this project.

1. Use Case Model

a. Given the requirement, **REQ-1**: The **Space Mission System** shall transport a message within 10 seconds between users, create a use case that refines **REQ-1**.
b. Create a view (table, matrix, or SysML Requirements Diagram) that shows the appropriate relationship between the use case and requirement using SysML.
c. Create a **SysML Use Case Diagram** to show your use case, actors, and system of interest. To simplify, assume you have two actors, **Sender** and **Receiver**, and the system of interest is the **Space Mission System**.

2. Scenario Model

A general description of the scenario is as follows:

- **Sender** sends a request to the **Space Mission System** to enable communication
- **Space Mission System** verifies the authenticity of the request
- If the request is verified, the **Space Mission System** sends an acknowledgement to the **Sender** that communication is enabled.
- If the request fails to verify, the **Space Mission System** sends an acknowledgement to the **Sender** that communication is not enabled; therefore, the **Sender** cannot transport any messages to the **Receiver**.
- After communication is enabled by the **Space Mission System**, **Sender** can send messages until **Sender** sends a request to close the connection. After each message **Sender** sends to the **Receiver**, the **Receiver** sends an acknowledgement message to acknowledges receipt of the message.
- When **Sender** sends a request to end communication to the **Space Mission System**, the **Space Mission System** closes the communication connection so that no further messages can be sent.

a. Given the scenario above, create a use case scenario for the use case in 1a using a SysML Activity Diagram with swim lanes to show which actor or system of interest that is performing the scenario action.
b. Create a SysML dependency relationship between the scenario and use case to show that the scenario depends on the use case.
c. Create a view (table, matrix, or a SysML diagram) to show the relationship between the use case and scenario.
d. Derive a set of requirements from REQ-1. Create a SysML requirement element for each derived requirement and show the requirements in a requirements hierarchy on a SysML Requirements Diagram
e. The requirements should map to the scenario actions using the appropriate SysML relationship that asserts the scenario actions satisfies the derived requirements.

3. Logical Component Model

a. Given the **Space Mission System** is realized by a **Satellite** and a **Control System**, create a view (table, matrix, or SysML Block Definition Diagram) that shows the realization relationship between components of the **Space Mission System**.
b. The **Sender** (from the use case model) is realized by a **Human Operator1** and **Computer Terminal1**, create a view (table, matrix, or SysML Block Definition Diagram) that shows the realization relationship between components of the **Sender**.
c. The **Receiver** (from the use case model) is realized by a **Human Operator2** and **Computer Terminal2**, create a view (table, matrix, or SysML Block Definition Diagram) that shows the relationship between components of the **Receiver**.
d. Create a state-based behavior (using a SysML Finite State Diagram) for the **Satellite**, **Control System**, **Human Operator1**, **Computer Terminal1**, and Computer Ternminal2.
Tip: to create a behavior for a SysML Block (to be used for each component), create a "default behavior". The default behavior is the behavior that the owning Block will perform.
e. Create SysML Full Ports for your components. Each port should be typed by a SysML Block that represents the type of information that is exchanged between components and the components' behavior.
f. Create a SysML **Internal Block Diagram** to show the topology of your logical architecture (connection between components). Note: the key to identifying the right connections is to ensure that the connections realize the interactions between actors and the system of interest from the use case scenario model.
g. Derive a set of requirements from the requirements created in 2d. Create a SysML requirement element for each derived requirement and show the requirements in a requirements hierarchy on a SysML Requirements Diagram.
h. The requirements should map to the components using the appropriate SysML relationship that asserts the components satisfy the derived requirements.

Use Case Modeling

Requirement:

- The system shall transport a message within **10 seconds** between users.
- The **Use Case Diagram** must show **Sender, Receiver, and the Space Mission System** as the primary actors.
- The relationship between **requirements and the use case** must be captured in a SysML **Requirements Diagram**.

Implementation in this project:

✓ **SysML Use Case Diagram** created to define system interactions (ref. Figure 1).
✓ **SysML Use Case Diagram** maps use cases to REQ-1 (ref. Figure 1).
✓ **SysML Use Case Diagram** shows requirement traceability (ref. Figure 1).

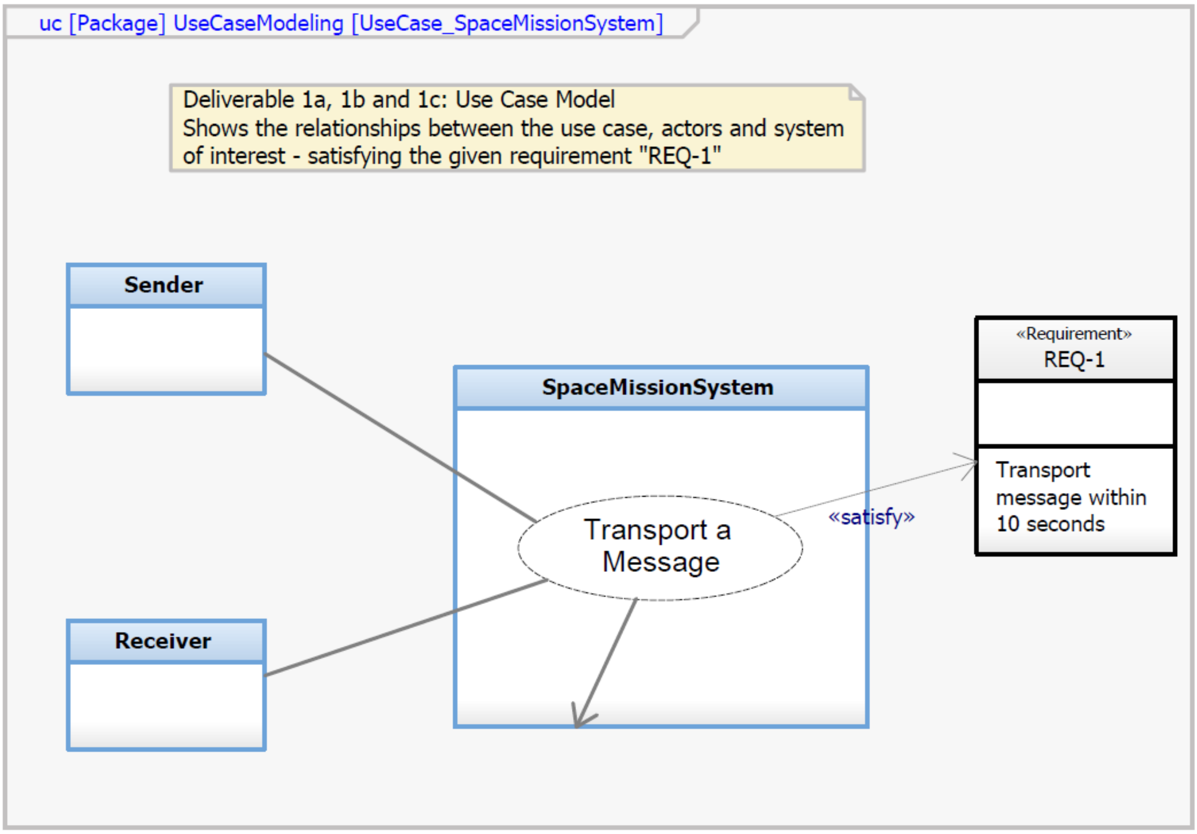

Figure 1: Use Case Diagram

Scenario Model

Requirement:

- The **activity diagram** must model the full message transmission process, including authentication and acknowledgment.
- The **use case and scenario relationship** must be mapped.
- **Derived requirements** from REQ-1 should be created.

Implementation in this project:

✓ **Dependency Diagram** links scenario model to use case model (ref. Figure 2).
✓ **Activity Diagram** with swim lanes demonstrates sender-receiver interactions (ref. Figure 3).
✓ **Derived SysML Requirements Diagram** captures constraints like message verification timing (ref. Figure 4).
✓ **Derived SysML Requirements Diagram** demonstrates the relationship between each derived requirement with the respective action (ref. Figure 5).

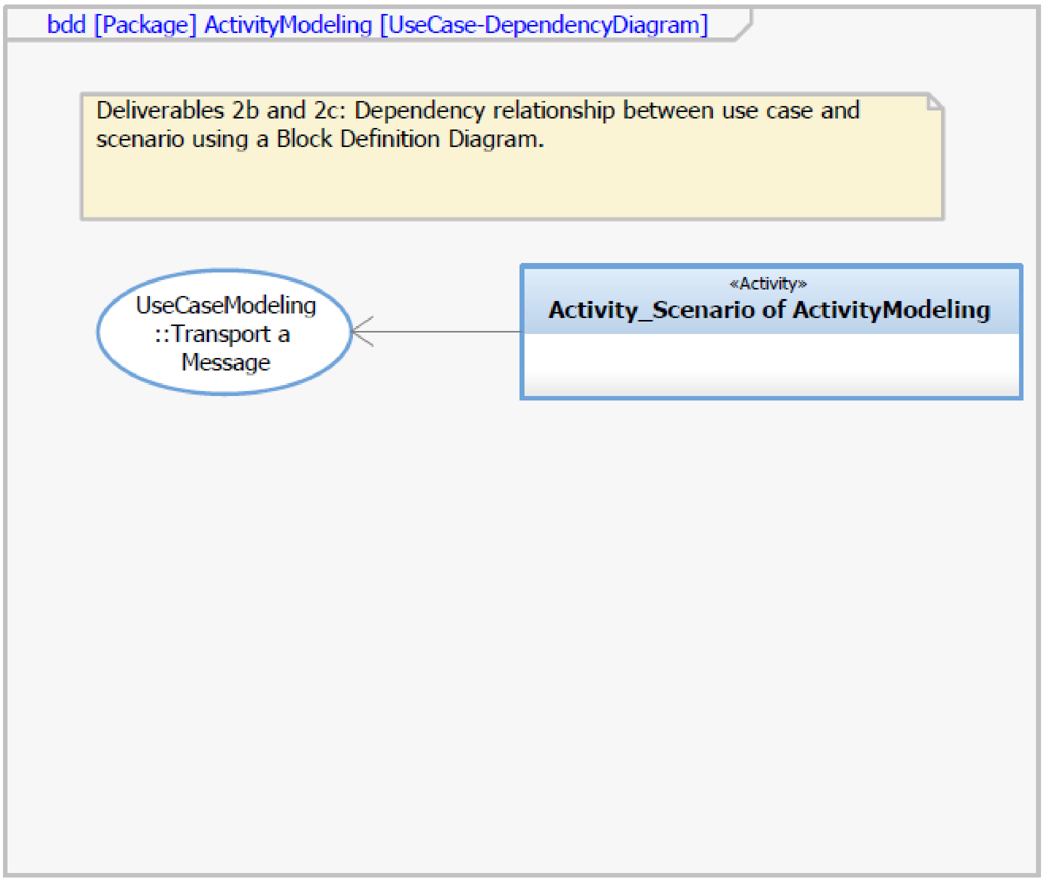

Figure 2: Dependency Diagram

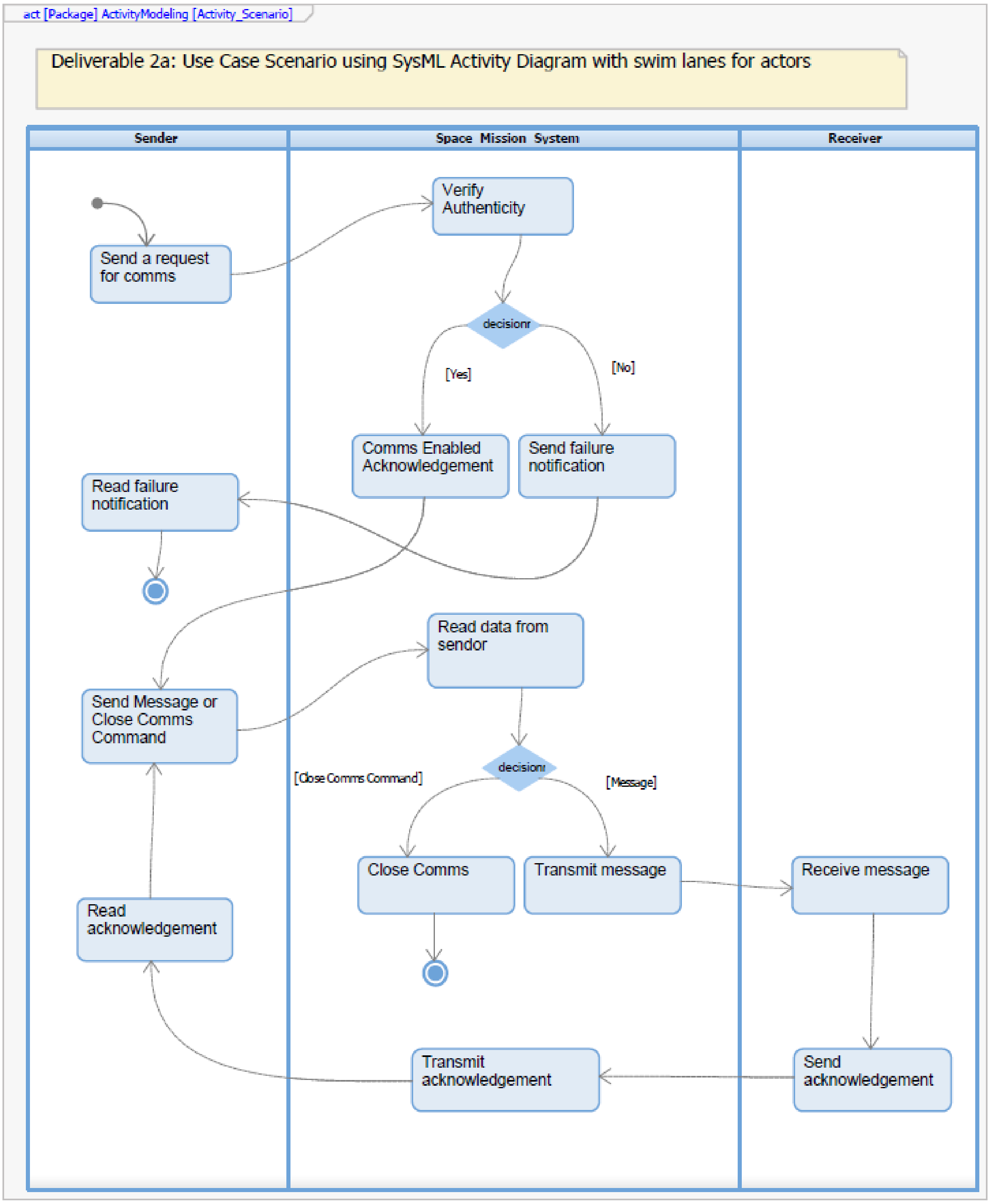

Figure 3: Activity Diagram

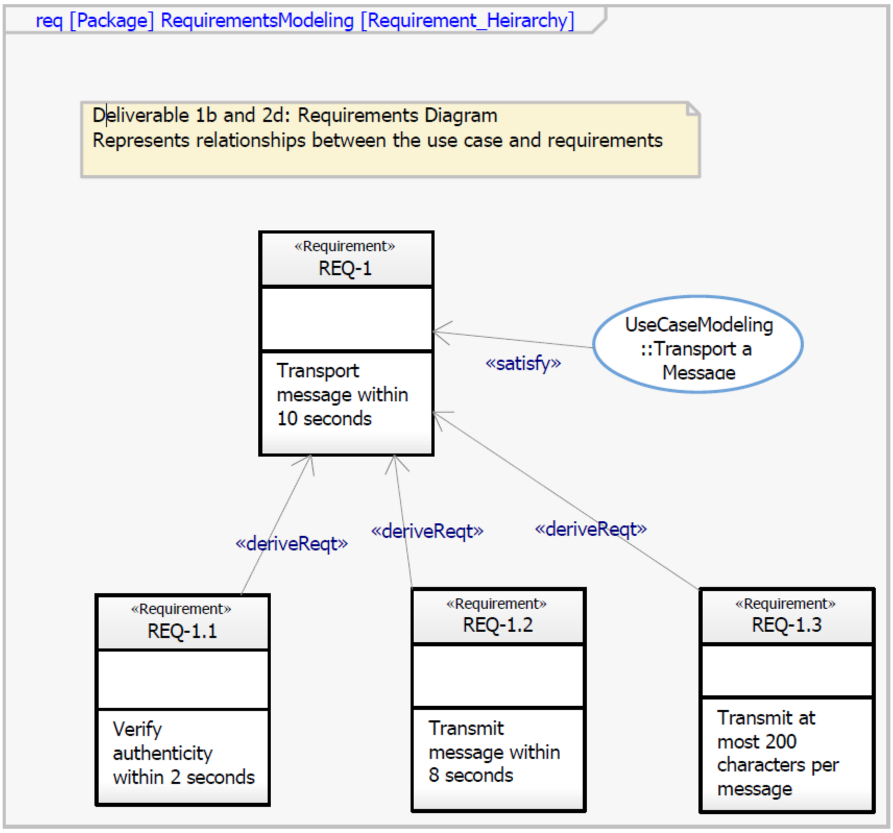

Figure 4: Requirements Diagram

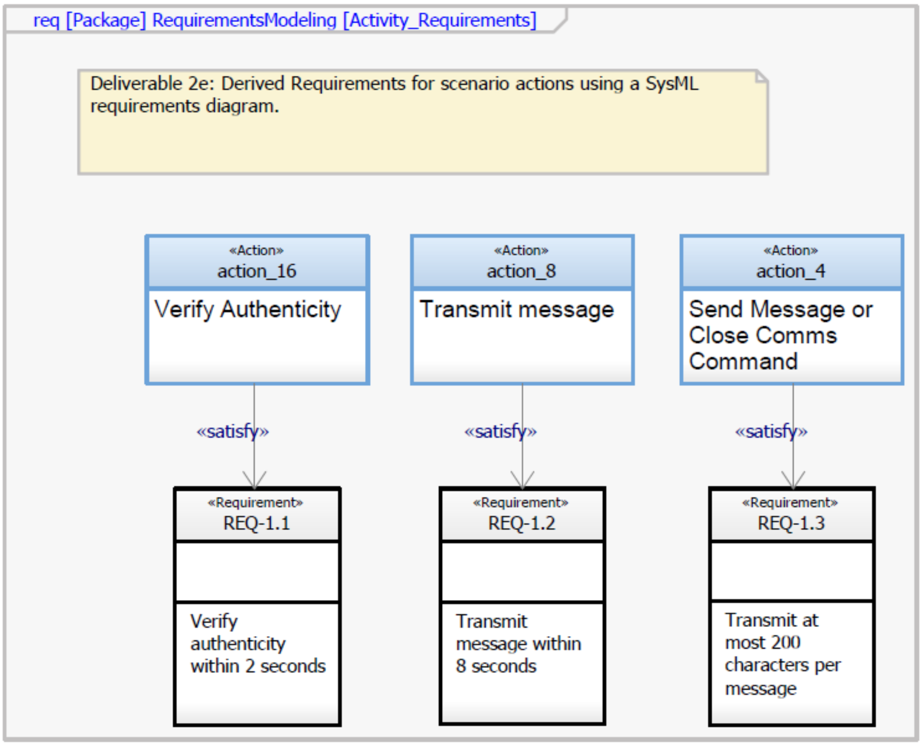

Figure 5: Activity Requirements

Logical Component Model

Requirement:

- The **Space Mission System** must be realized by **Satellite and Control System**.
- **Sender and Receiver** must be realized by **Human Operators and Computer Terminals**.
- **Finite State Diagrams** must show behavior transitions.

Implementation in this project:

- ✓ **Block Definition Diagrams (BDDs)** illustrate the realization of system components (ref. Figures 6, 7 and 8).
- ✓ **Finite State Diagrams** for Satellite, Control System, Human Operators, and Terminals (ref. Figures 9, 10, 11, 12, 13).
- ✓ **SysML Internal Block Diagram (IBD)** demonstrates component connections (ref. Figure 14).
- ✓ **SysML Requirements Diagram** demonstrates the relationships between all system components with their requirements in a single diagram (ref. Figure 15).

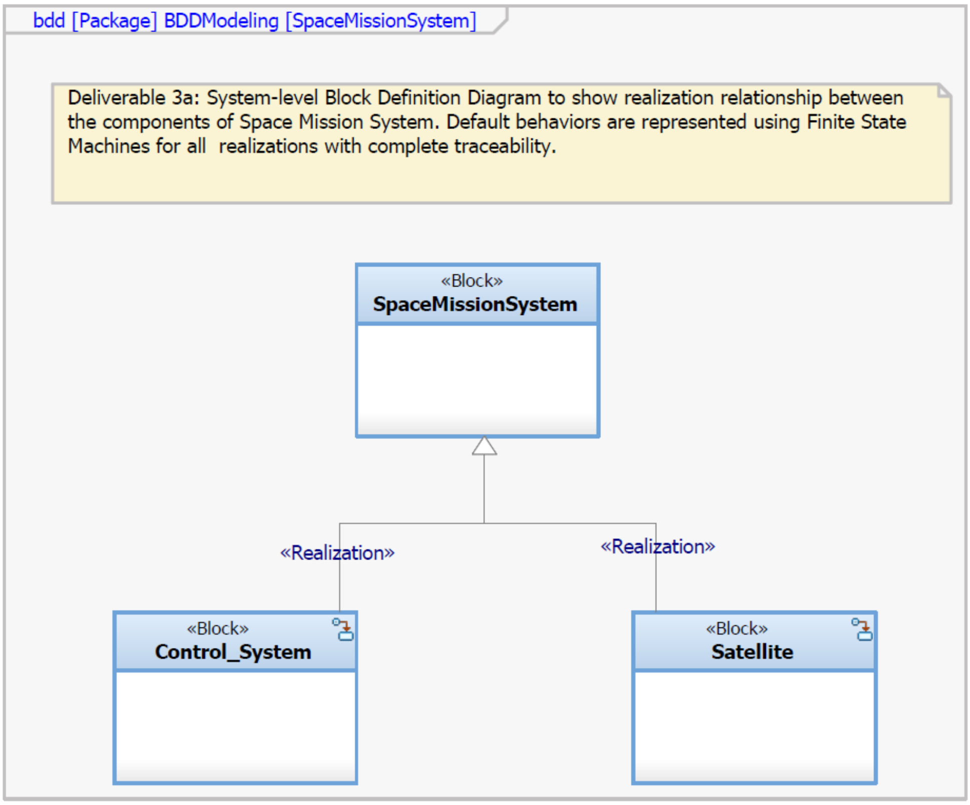

Figure 6: Block Diagram - Space Mission System

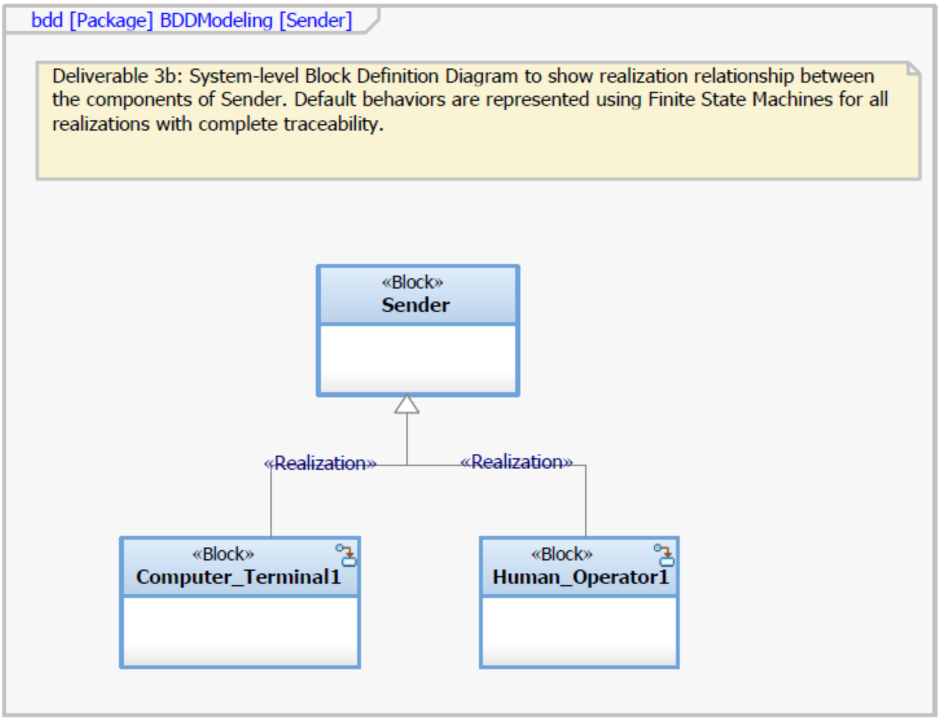

Figure 7: Block Diagram – Sender

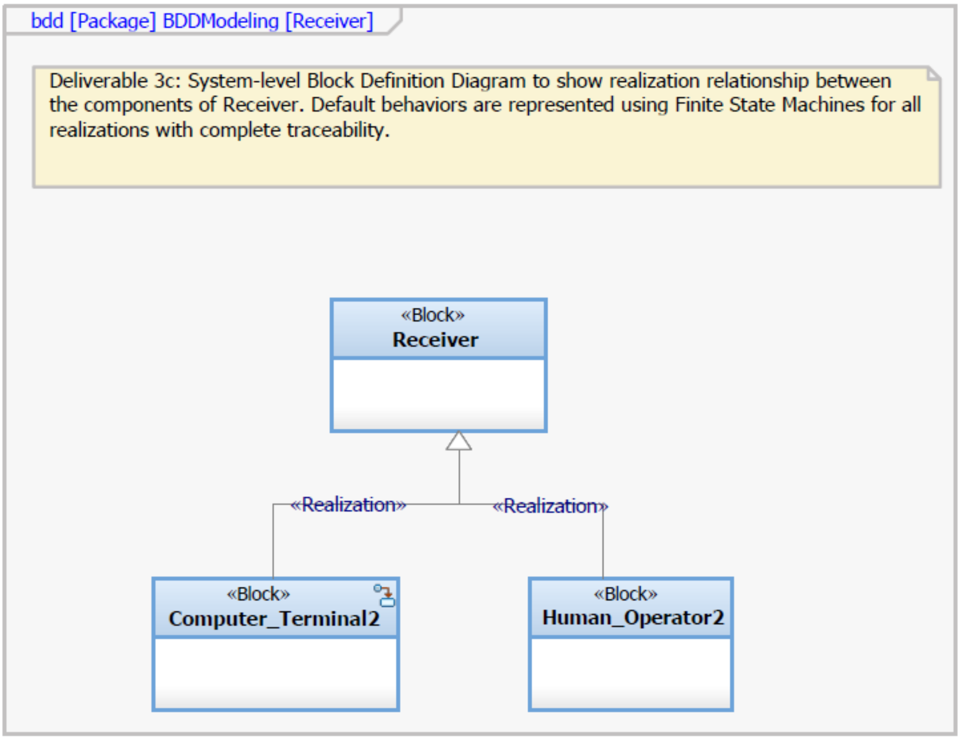

Figure 8: Block Diagram – Receiver

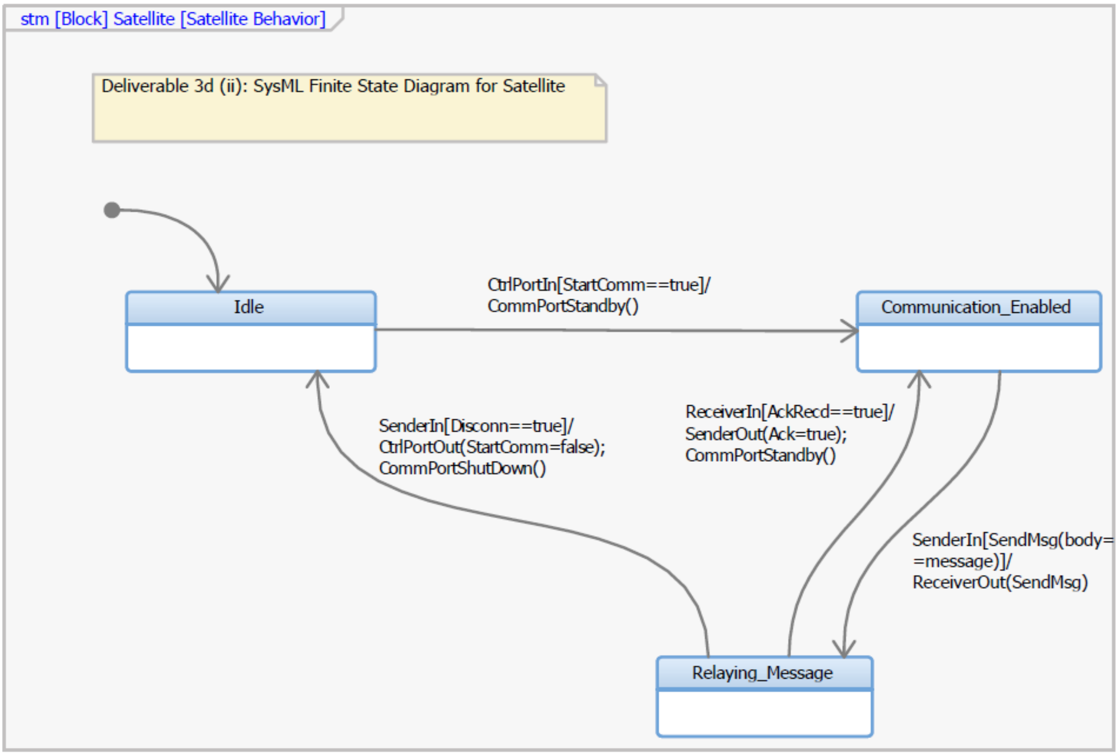

Figure 9: Finite State Machine – Satellite

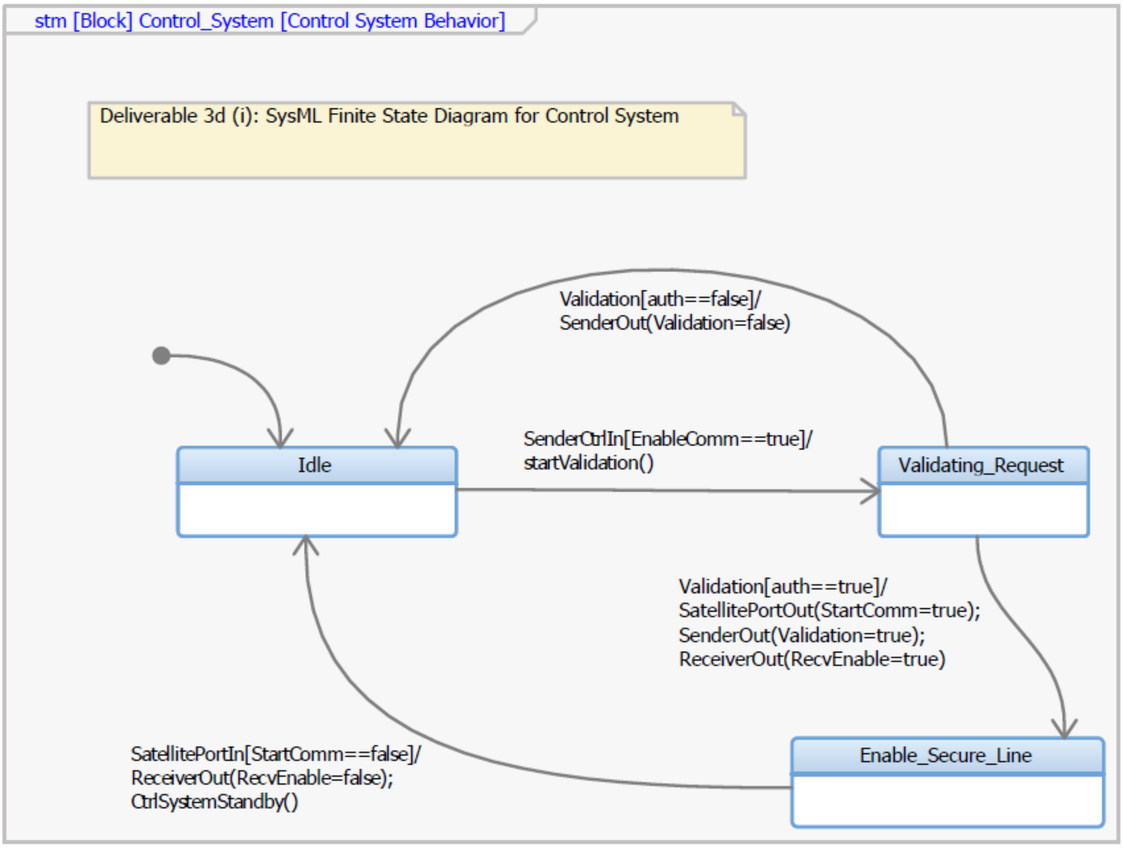

Figure 10: Finite State Machine - Control System

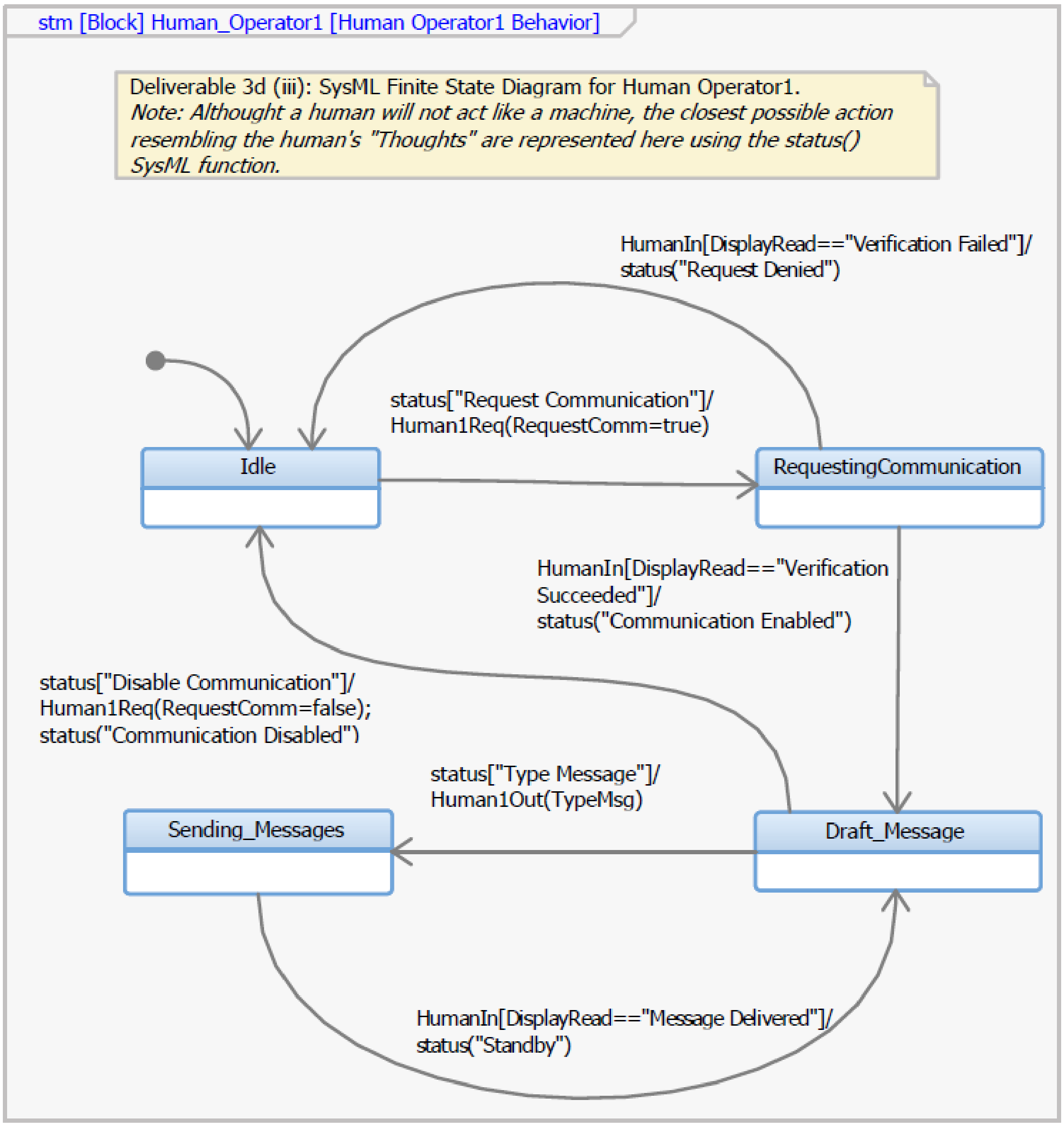

Figure 11: Finite State Machine - Human Operator1

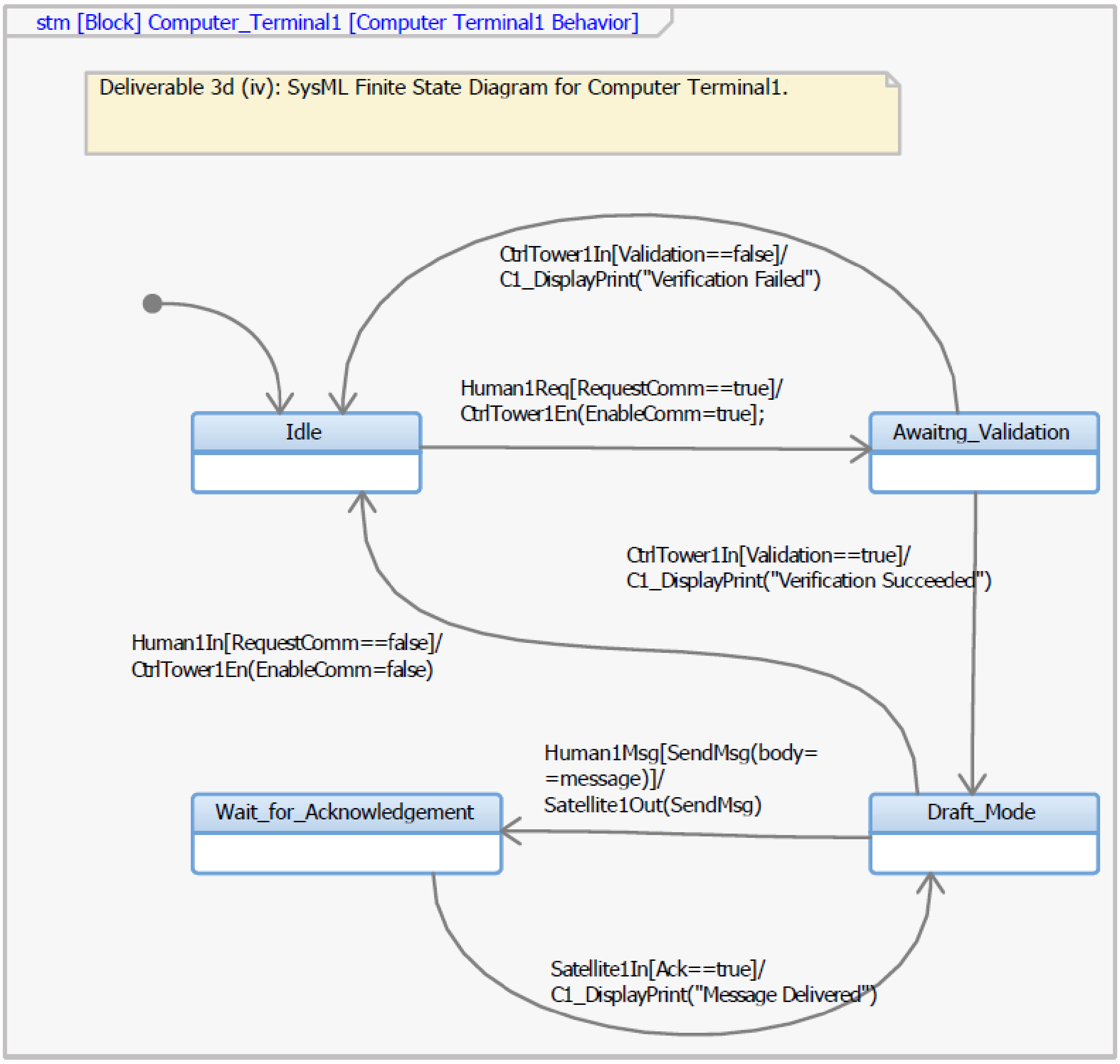

Figure 12: Finite State Machine - Computer Terminal1

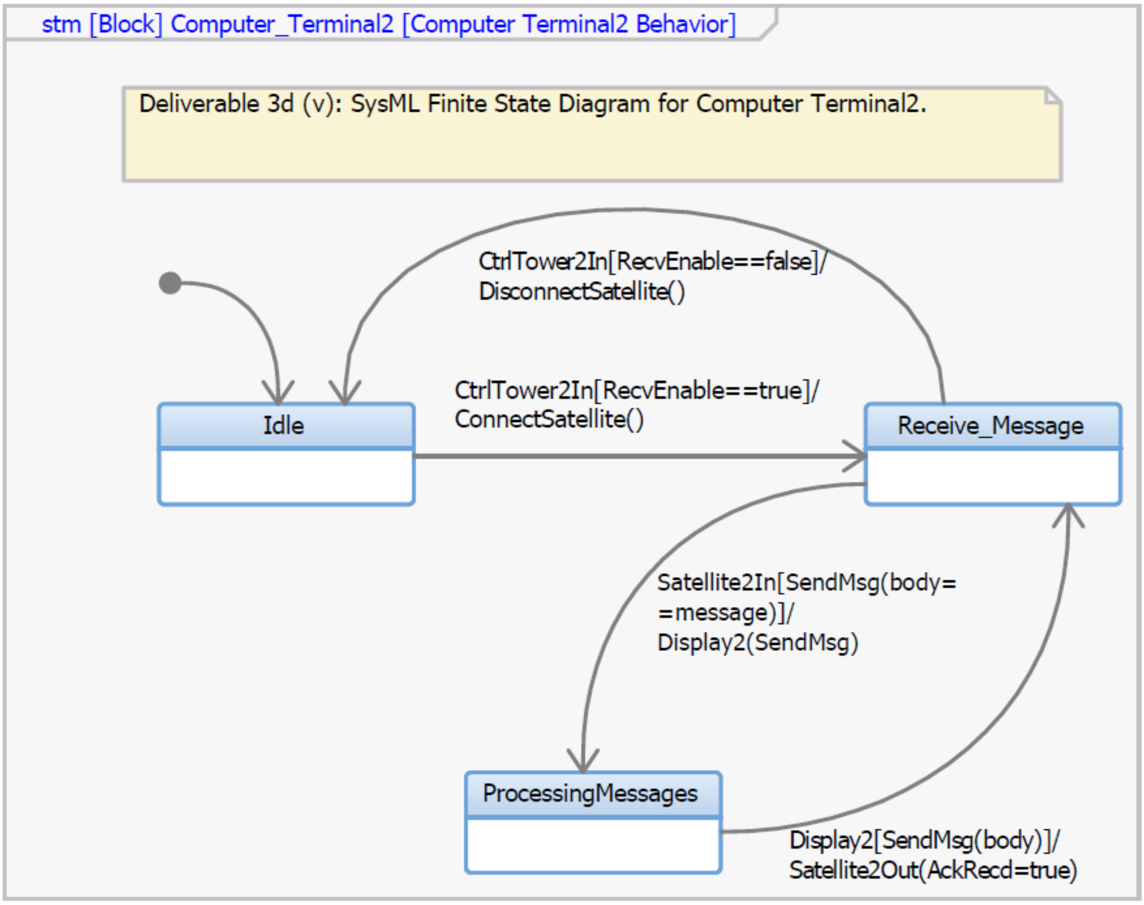

Figure 13: Finite State Machine - Computer Terminal2

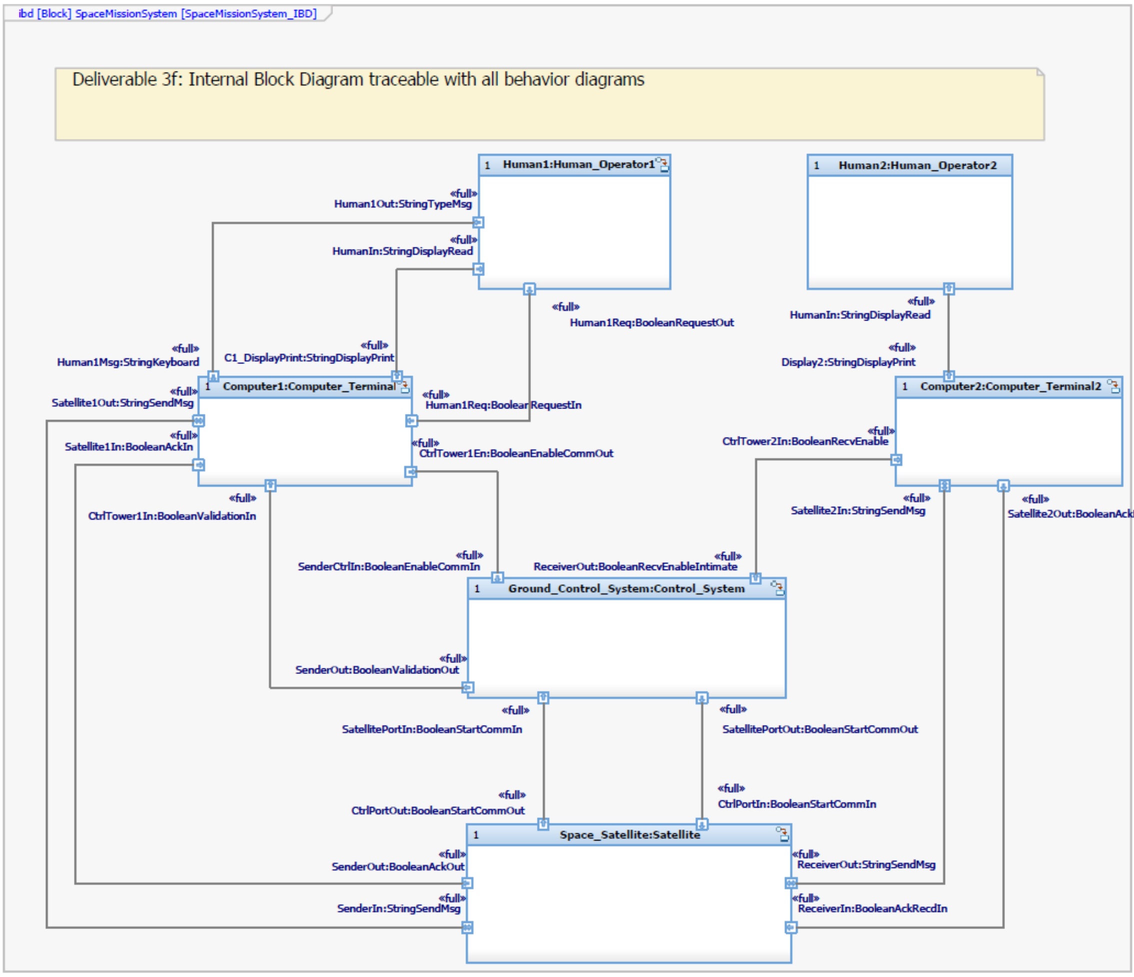

Figure 14: Internal Block Diagram

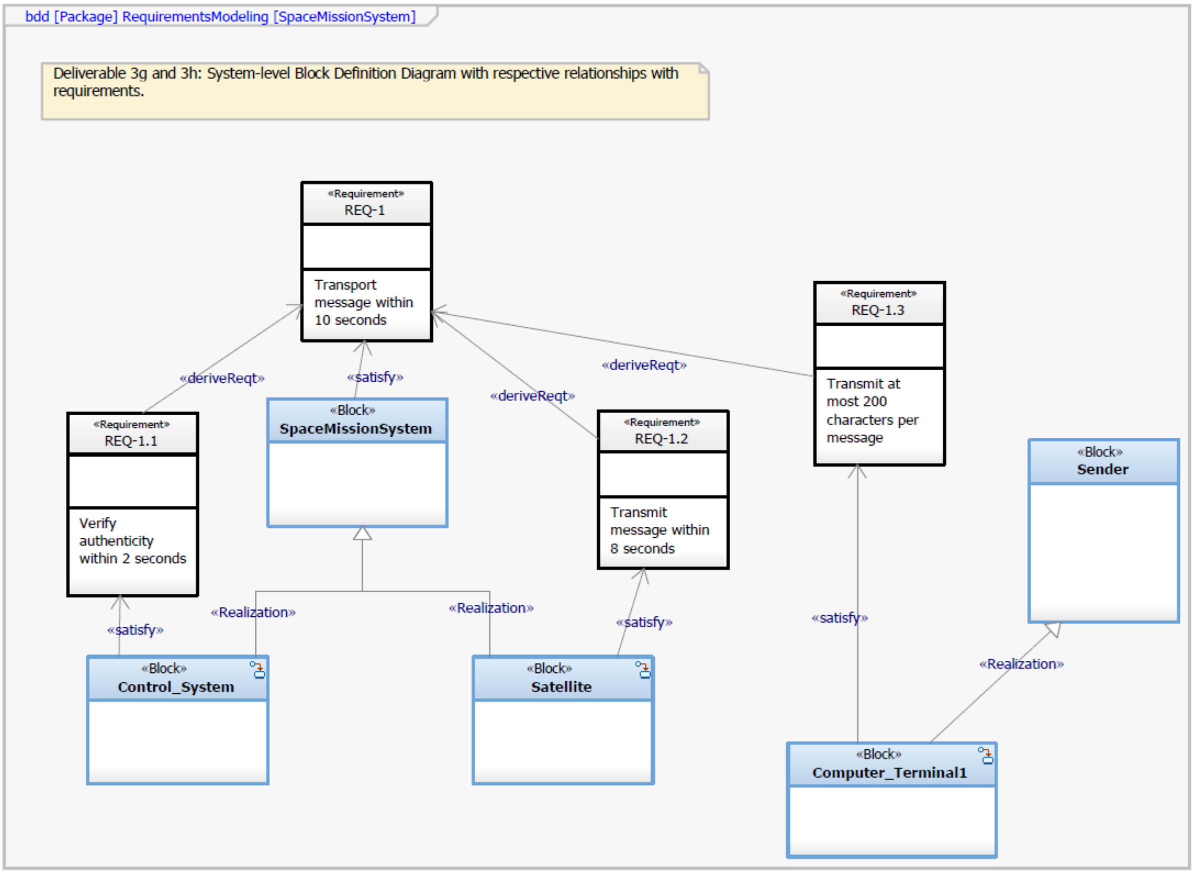

Figure 15: Requirements Diagram with Block Definition Diagram

Conclusion and Industry Relevance

The **increasing complexity of modern engineering systems** demands structured approaches like **Model-Based Systems Engineering (MBSE)**. This project reinforces the **role of MBSE in industries** such as **aerospace, automotive, and high-tech product development**.

By leveraging **SysML in IBM Rhapsody**, engineers can:

- **Enhance requirement traceability** to ensure robust system designs.
- **Optimize workflows** through structured modeling.
- **Develop reliable digital twins** for system validation and real-time analysis.

MBSE is an **essential skill for modern product development**, helping engineers bridge the gap between conceptual design and practical implementation.

Acknowledgements

I am incredibly **grateful to my family and teachers** for guiding me in selecting electives that align with my career aspirations. Their mentorship and encouragement have played a crucial role in my learning journey.

This course has been instrumental in broadening my understanding of **systems engineering** and its applications in **product development**, helping me build skills that are **highly valued in the industry**.