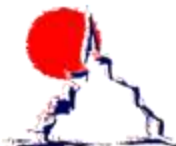


# Brief introduction to SysML

David MENTRÉ

2013-04-16

Work licensed under Creative Common Attribution-ShareAlike 3.0 Unported License



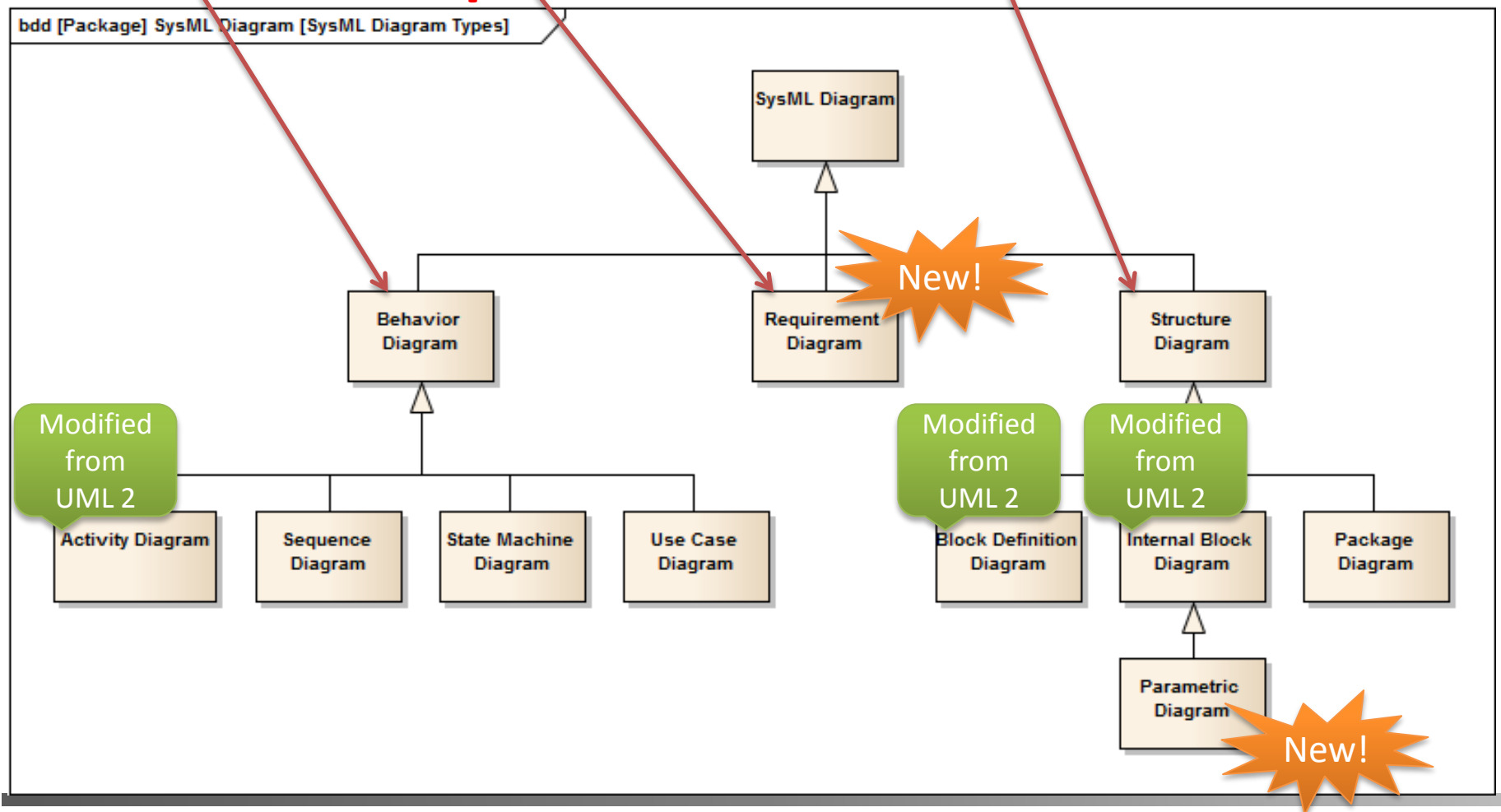
# What is SysML?

---

- **S**ystem **M**odeling **L**anguage
  - **Derived** from (aka profile of) **UML** 2.0
  - Like UML but **system-wide**, not only software
- OMG (Object Management Group) **standard**
  - v1.0: September 2007
  - **v1.3**: June 2012
- Supported by many **tools**
  - Rhapsody, Enterprise Architect, Magic Draw, Artisan Studio, Modelio, Papyrus, Topcased, ...

# SysML diagrams

- 9 kinds of diagrams to model **Behavior, Requirements and Structure**



# Diagram kinds

---

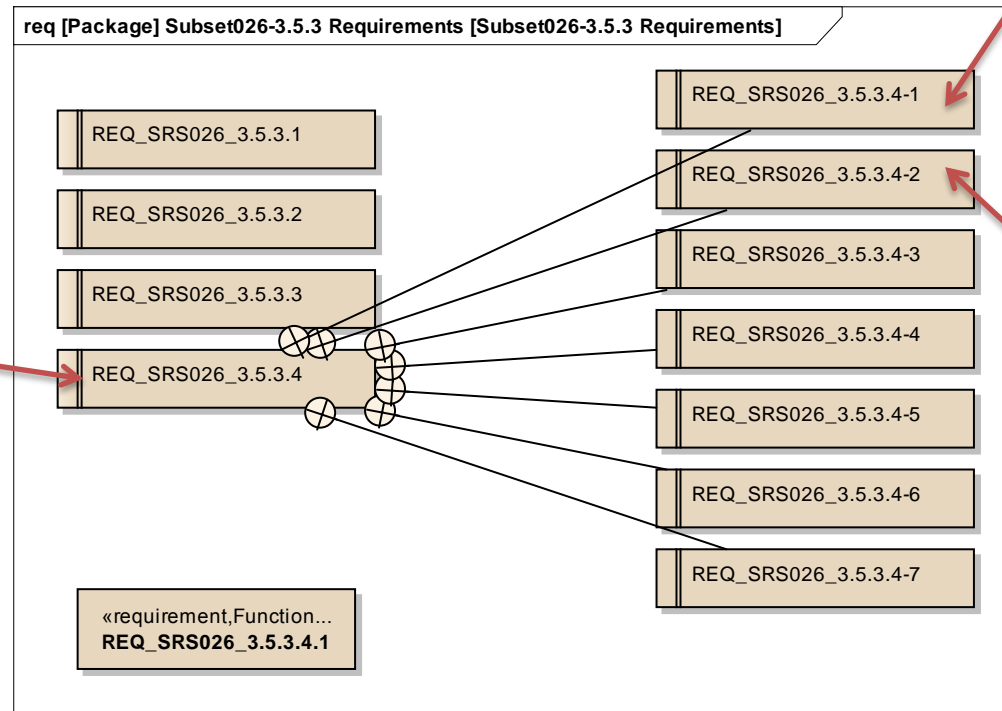
- **Structure** diagrams
  - **Static composition** of the system
    - blocks, sub-blocks, ...
  - **Static relationships** between blocks
    - Information or physical flows
- **Behavior** diagrams
  - **Dynamic** behavior of the system
    - How the system evolves along time
  - **Evolution** and control of information or physical **flows**

# Requirement diagram

- Express **requirements** and their **relationships**
  - Containment, Trace, Copy, Derive, Verify, Refine, Satisfy
  - Between two req. or req. and model elements

- Excerpt

The on-board shall establish a communication session



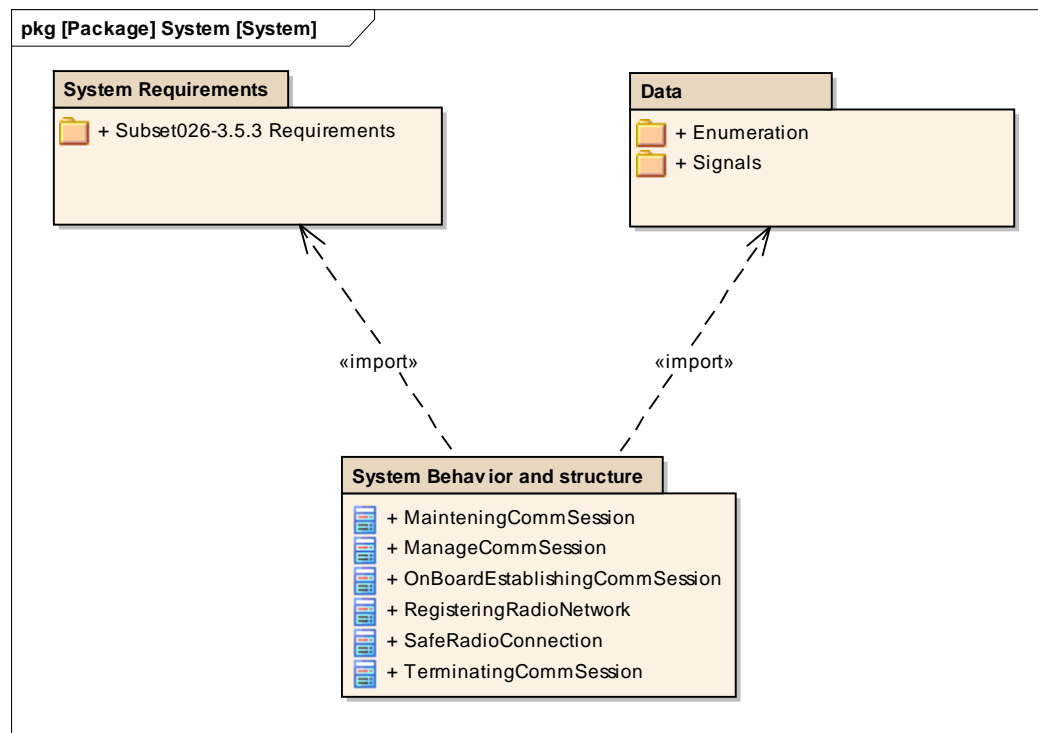
a) At Start of Mission (only if level 2 or 3).

b) If ordered from trackside.

# STRUCTURE DIAGRAMS

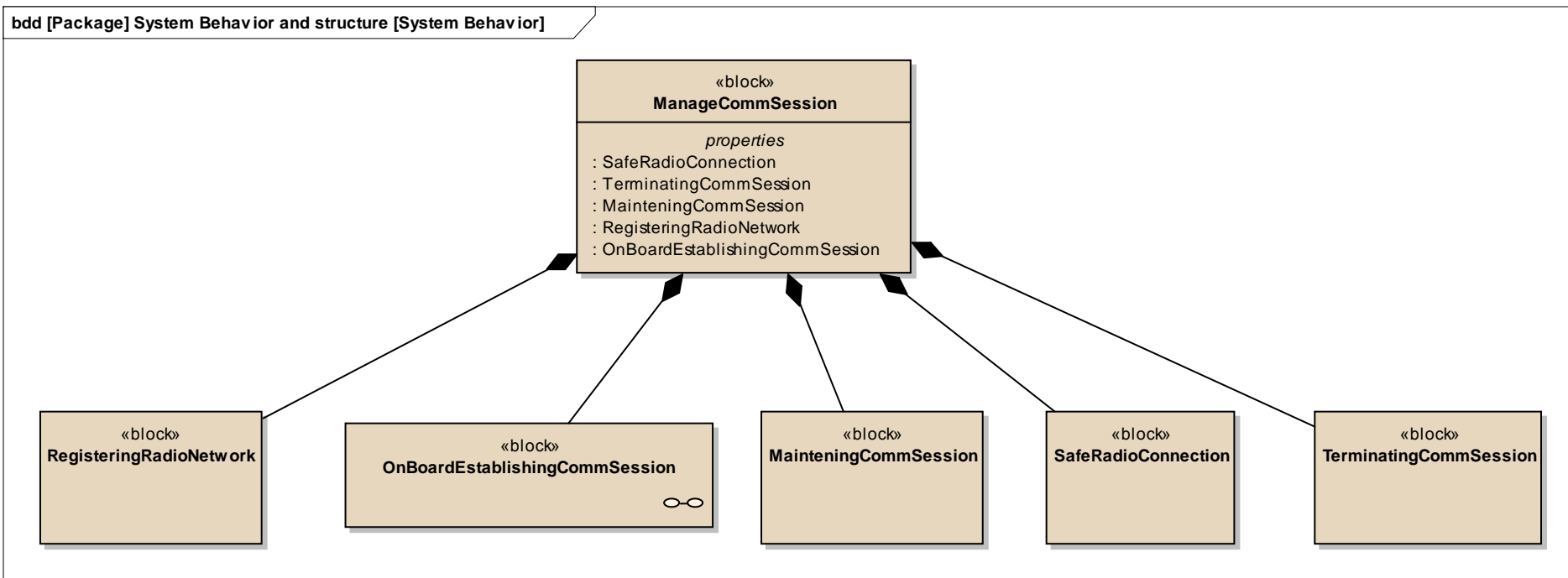
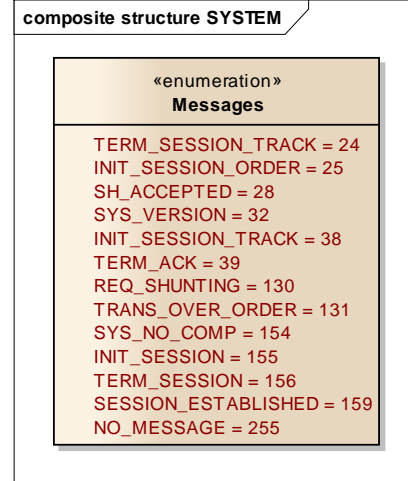
# Package diagram

- **Group** elements for **readability** and **scalability**
- Allow to define objects **reused** in other parts of the model



# Block Definition diagram

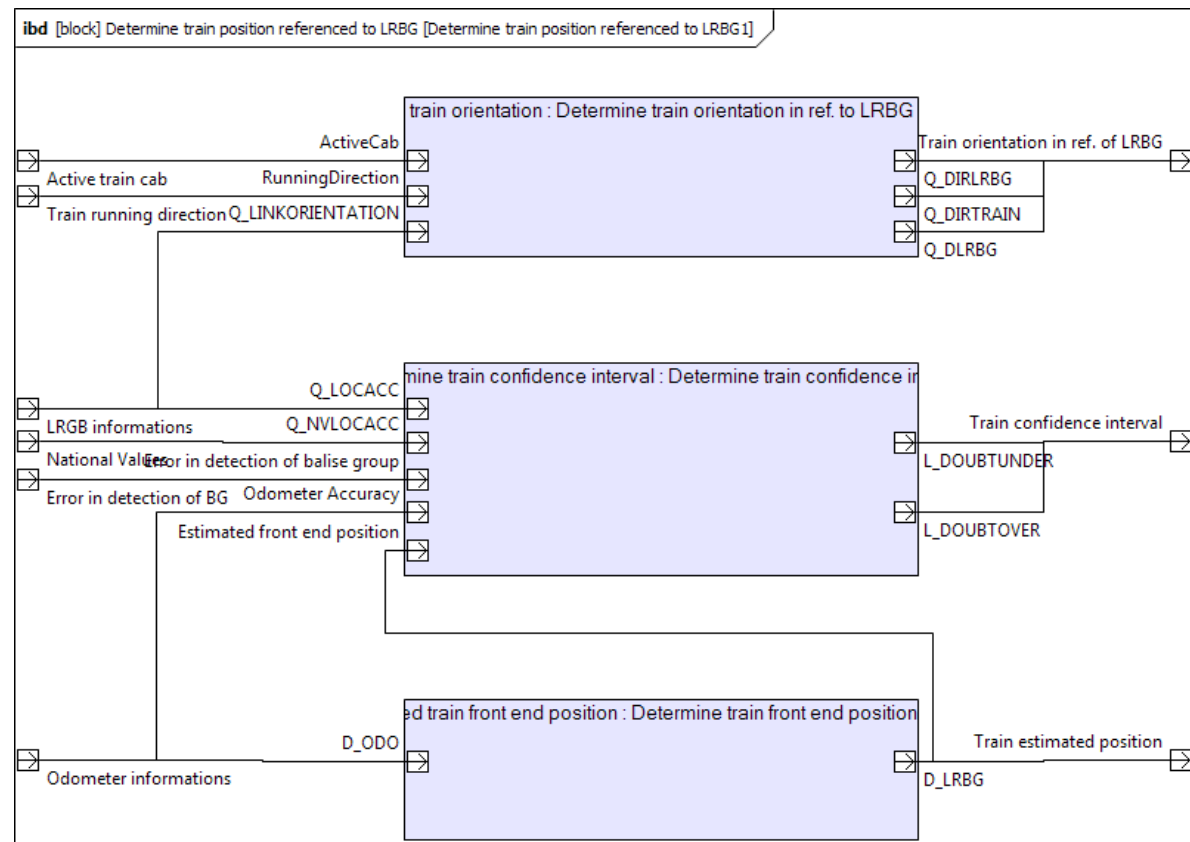
- Static block **content**
- **Relationships** between blocks
- Definition of block **parts**





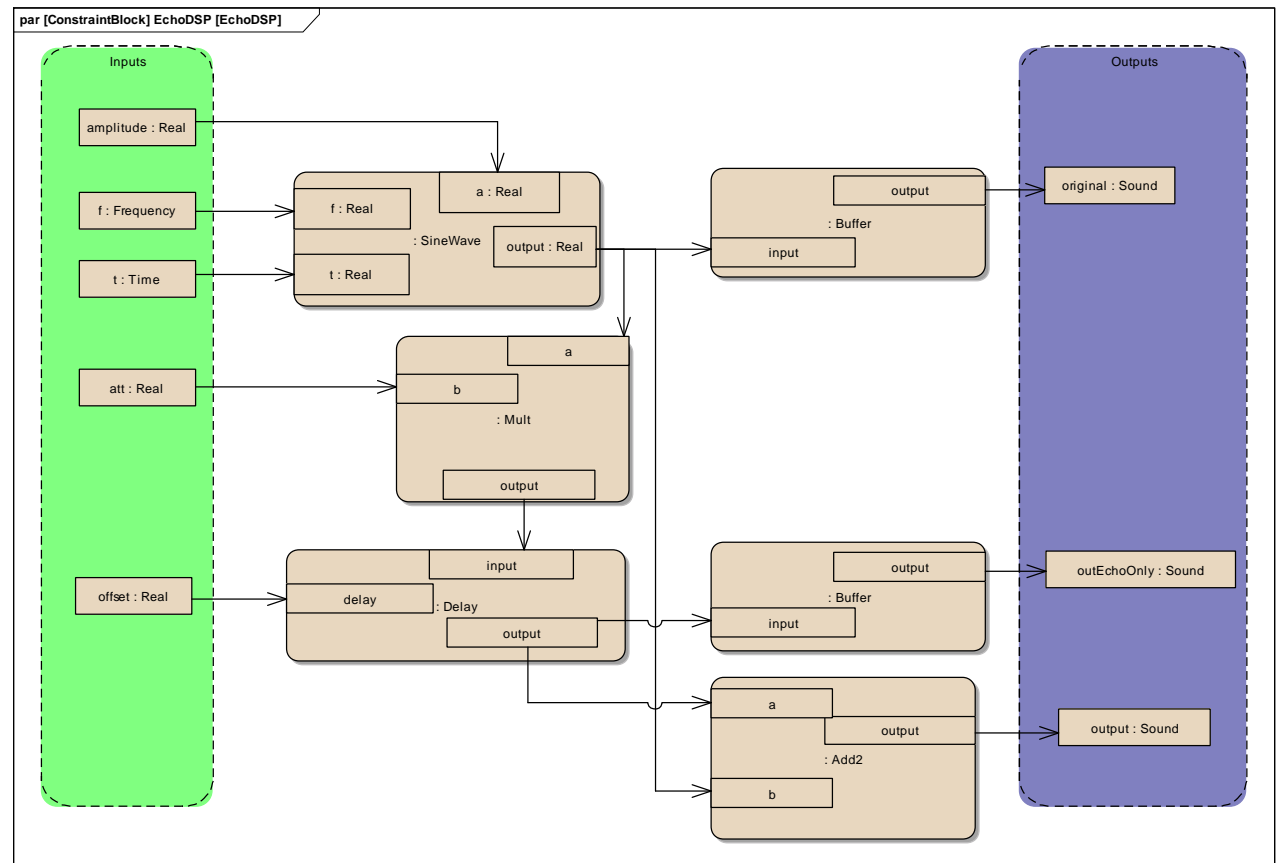
# Internal Block diagram

- **Flow** of information between parts and their own blocks
- **Input/output ports, types, ...**



# Parametric diagram

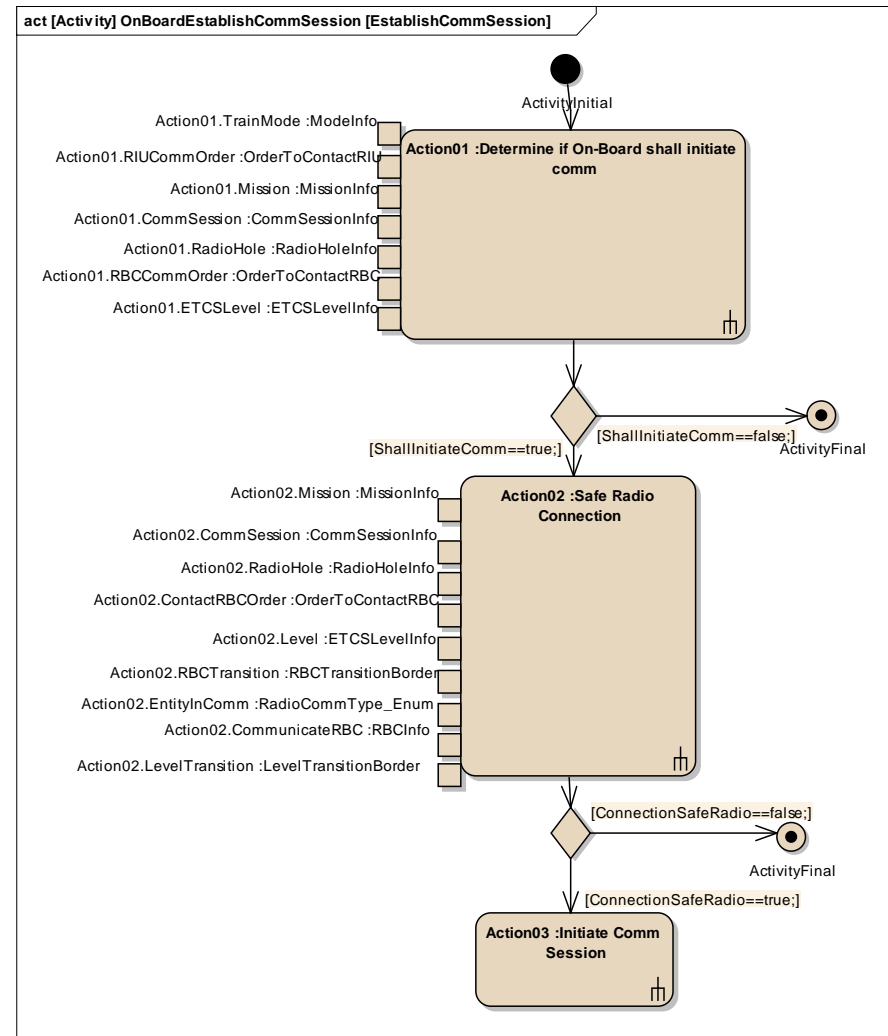
- Mathematical **equations** and constraints



# BEHAVIOR DIAGRAMS

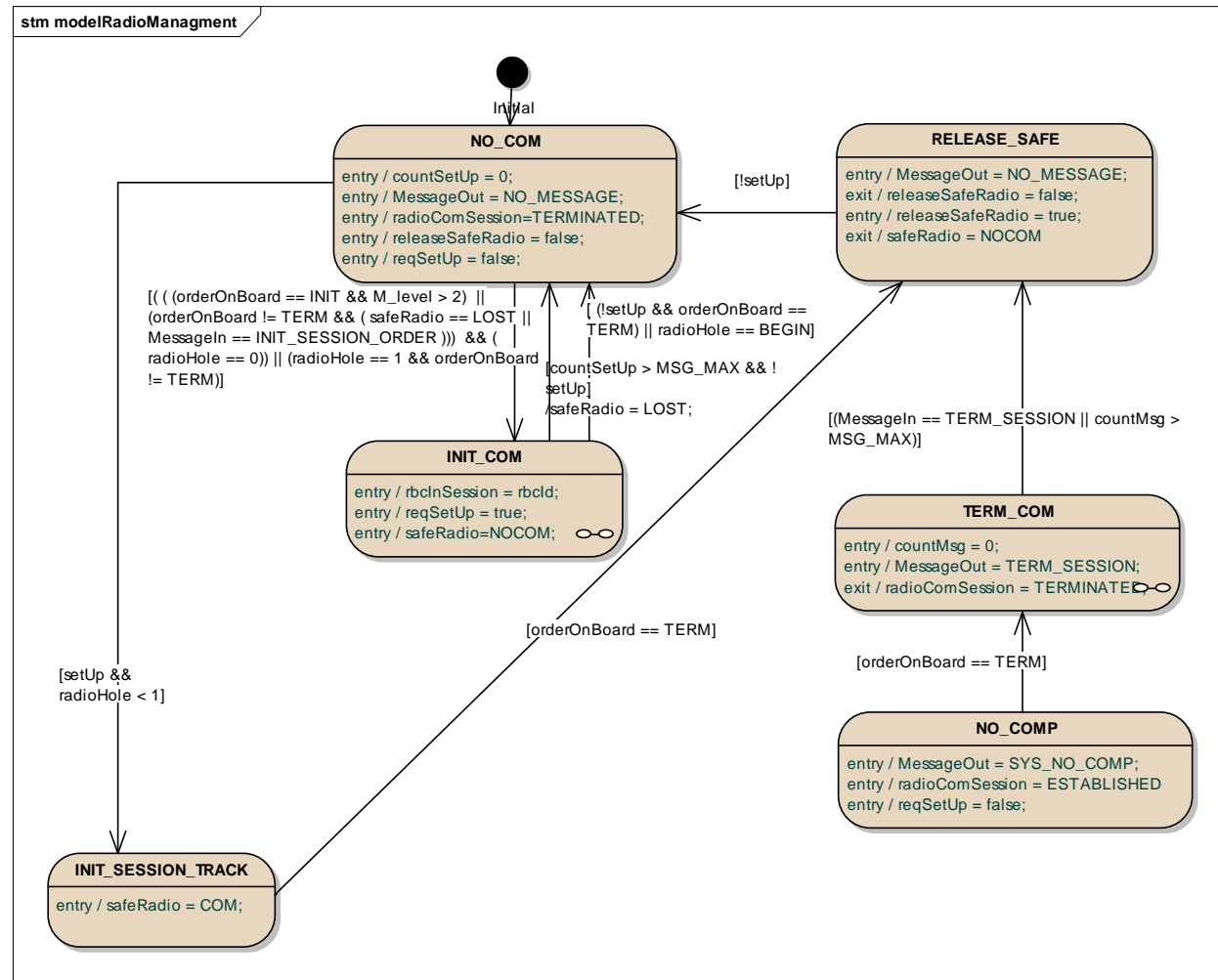
# Activity diagram

- **Flow** of information **controlled** with a sequence of actions
- Describe
  - Block **behavior**
  - How block **outputs** are computed
- Very similar to **Petri Nets**



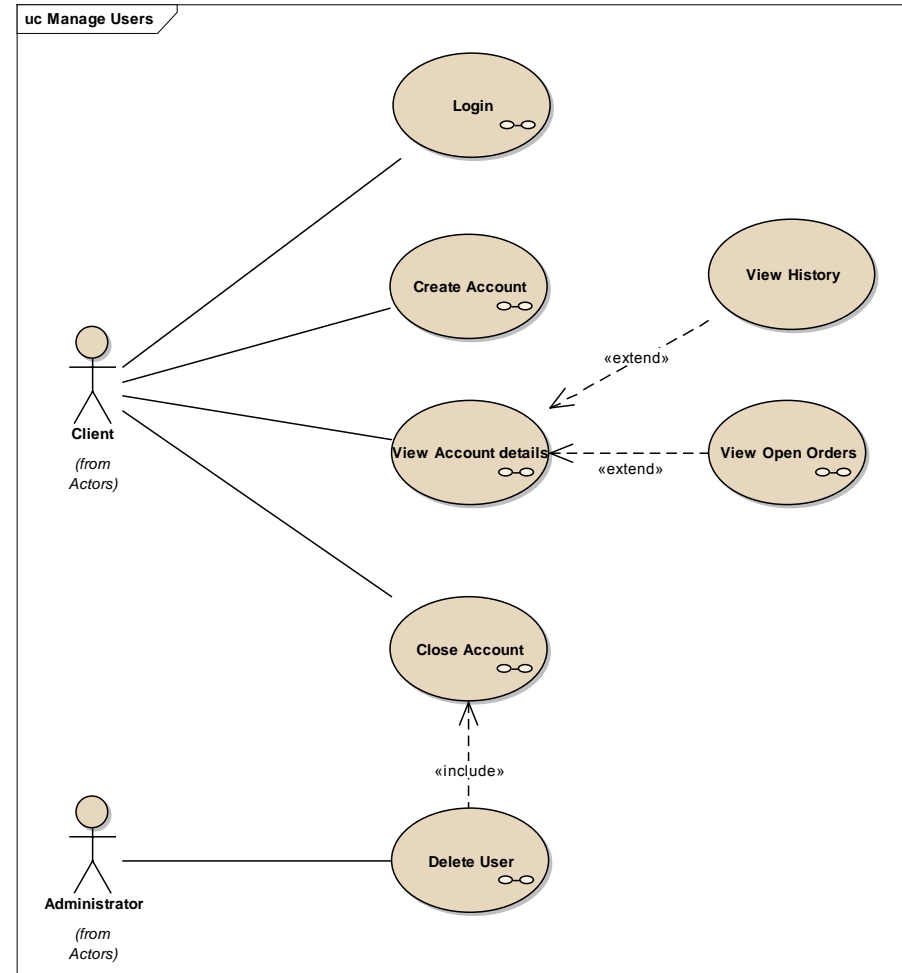
# State-machine

- **States** and **transition** between states
- **Hierarchy** of state machines
- Excerpt
  - from Cecile's model



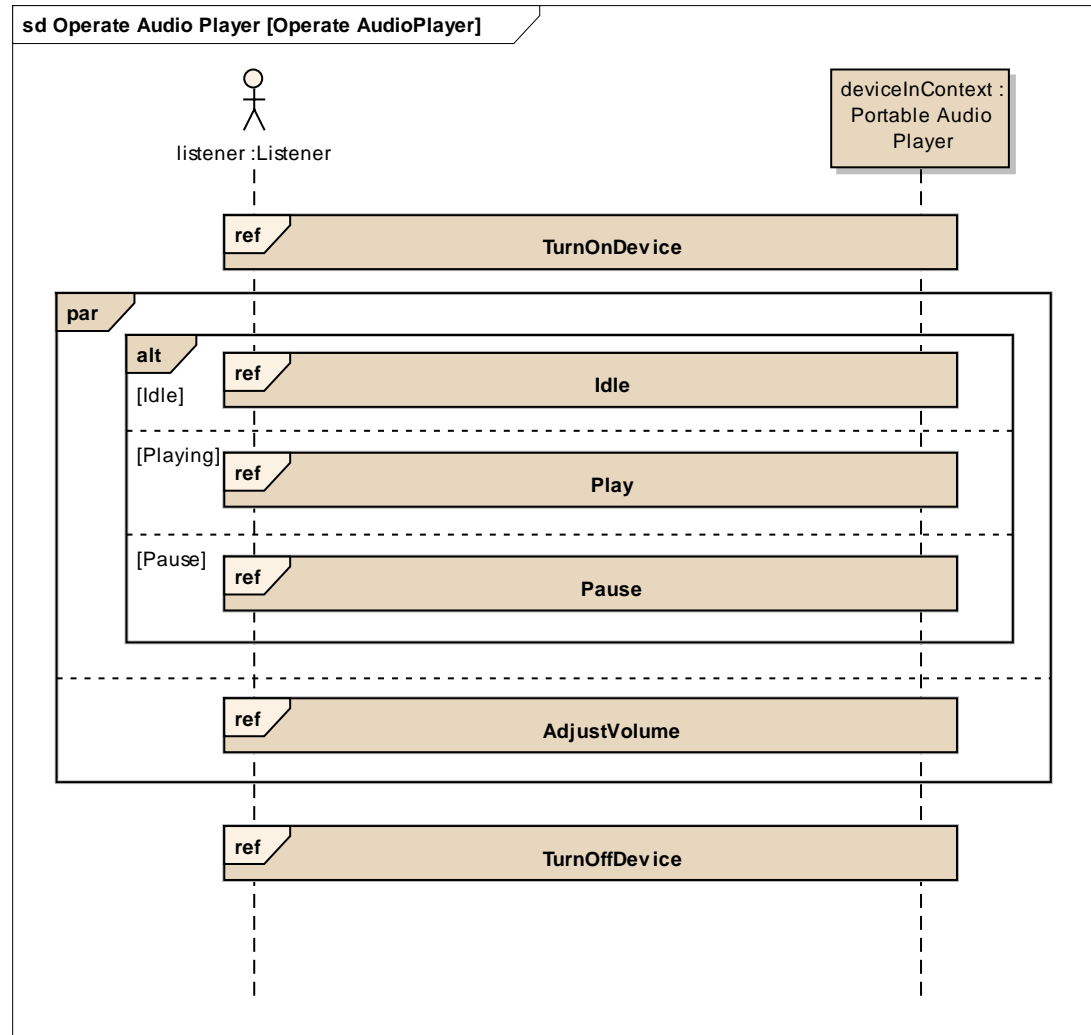
# Use-case diagram

- Example of **use** of described system
- Describe which **actors** and system **parts** are interacting



# Sequence diagram

- **Interactions** between different parts of the system
- Flow of **exchanges** along the time



 **TO CONCLUDE**



# Why use SysML... or not?

---

- To model a **system**
  - Including hardware, software, physical components, ...
  - Group all design artifacts into a **single model**
    - A single reference
- Well **known notation**
  - Really? Which semantics behind elements and connectors?
- **Graphical** notation
  - “A picture is worth a thousand words”... except when one has **thousand** pictures?
  - Is a picture with **all** information (types, names, ...) still readable?
- Questions?

