Departamento de Informática

Engenharia de Software
(14341, 16230, 15386)

L07: Dynamic Modelling #2

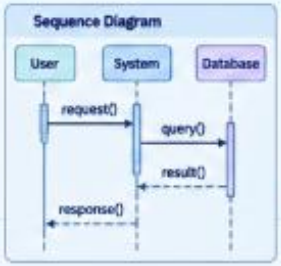
L07

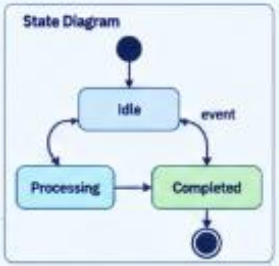
Practical Lab #7

Sequence and State Diagrams

Modelling Interactions and Behaviour Over Time

Model Interactions. Understand Behaviour.





Today's Competency Readiness Indicators

By the end of this lab, you will be better able to:

Identify interactions and messages

Determine the main actors, objects and messages in a use case and represent their interactions in sequence diagrams.

Model the temporal behaviour

Represent the order and timing of events, activations and message flows, including alternative and parallel interactions.

Design state machines

Model the lifecycle of an object using state diagrams, defining states, events, transitions and guards.

Analyse and validate behaviour

Use sequence and state diagrams to analyse system behaviour, identify missing scenarios and verify consistency with requirements and use cases.

Ensure coherence between dynamic models

Maintain alignment between sequence diagrams, state diagrams and other UML models (e.g., use cases and class diagrams).

Communicate and justify design decisions

Explain the dynamic behaviour to stakeholders and justify modelling choices based on requirements, scenarios and expected system behaviour.

Related SFIA Competencies (main areas)

- **MODL** Systems Modelling
- **DESN** Systems Design
- **ANAL** Business Analysis
- **PROG** Programming / Software Development

PEOPLE | IDEAS | IMPACT

SMART CAMPUS
Engineering the Digital University

L07 – DYNAMIC MODELLING II: SEQUENCE & STATE DIAGRAMS

Goal: Model interactions between objects and how their states evolve over time.

PART 1 Project Brief

10 min

CLIENT REQUEST #3

"Students should receive an immediate confirmation after booking a study room and be notified of any changes."

To deliver this, the system must coordinate several components and manage the reservation status over time.

Your task:
Model the interactions and the state changes for the selected use case.

Think about:
- Which objects interact?
- What messages are exchanged?
- What are the key states?
- What events trigger transitions?

TODAY'S DELIVERABLES
- Sequence Diagram
- State Diagram (for main class)



PART 2 Core Project

60 min

Continue with the same use case from L06 (e.g., "Reserve Study Room").

1 Identify Lifelines (Objects)

List the key objects that participate in the scenario.

EXAMPLE (RESERVE STUDY ROOM)
- Student (Actor)
- Mobile App / Web Portal
- Reservation Service
- Room
- Notification Service
- Lecturer (Approver)

2 Create the Sequence Diagram

Model the main flow, including alt/opt fragments for alternatives and exceptions.

DELIVERABLE
Sequence Diagram
(+ brief explanation of the scenario and alternatives)

3 Create the State Diagram

Model the lifecycle of the main class (e.g., Reservation).

DELIVERABLE
State Diagram
(+ description of states and events)

PART 3 Engineering Challenge

20 min

INTERACTION & STATE CHALLENGE

A new requirement is introduced:

"If a reservation is not confirmed within 15 minutes, it should be automatically cancelled."

Analyse and answer:
1. How does this affect the sequence of messages?
2. What new messages or conditions are required?
3. How does this change the state diagram?
4. Which states or transitions are new or modified?

GUIDELINES
- Identify timer, event and guard conditions.
- Update both diagrams accordingly.
- Justify your design decisions.

DELIVERABLE
Updated Sequence Diagram +
Updated State Diagram
(+ justification of changes)

PART 4 AI Challenge

15 min

Use an LLM (e.g., ChatGPT) to generate both diagrams for the same use case.

SUGGESTED PROMPT

"Create a Sequence Diagram and a State Diagram for the use case 'Reserve Study Room' in the Smart Campus system. Include:
- Objects involved in the interaction
- Main flow and alternatives
- Lifecycle of the Reservation class
- Timer that cancels the reservation if not confirmed within 15 minutes."



Compare the generated diagrams with yours.

CHECKLIST
- [ ] Are the objects/lifelines appropriate?
- [ ] Are messages clear and ordered?
- [ ] Are alternatives and exceptions covered?
- [ ] Are states complete and consistent?
- [ ] Are transitions triggered by events?

PART 5 Reflection & Discussion

15 min

Discuss with your team and be ready to share.

- Which part of the sequence diagram was the most difficult to model? Why?
- Do the interactions you modelled reflect the activity workflow from L06? How?
- Which states are critical for the business rules? Why?
- What did you learn today about interactions and state evolution?

ENGINEERING INSIGHT

Sequence diagrams show how objects communicate to achieve behaviour.
State diagrams show how an object changes over time in response to events.

PROJECT MILESTONES

1 Requirements Engineering
- [x] Stakeholders
- [x] Features
- [x] Personas
- [x] Scenarios
- [x] User Stories

2 Requirements Analysis
- [x] Use Case Specifications
- [x] Use Case Diagram
- [x] Class Diagram (Domain Model)

3 Dynamic Modelling
- [x] Activity Diagram
- [x] Sequence Diagram
- [x] State Diagram

4 Software Design
- [ ] Software Architecture
- [ ] Design Decisions

Legend
- [x] Completed milestone
- [ ] Pending milestone

TIPS FOR SUCCESS

- Start from the main success scenario.
- Use alt/opt/loop fragments when needed.
- Keep messages at the right level of detail.
- Ensure states are meaningful and minimal.
- Validate with the use case specification.

SEQUENCE DIAGRAM ELEMENTS (UML)

Lifeline (Object)	opt Optional Fragment
Lifeline	loop Loop Fragment
Synchronous Message	par Parallel Fragment
Return Message	par Parallel Fragment
alt Alternative Fragment	Message to Self

STATE DIAGRAM ELEMENTS (UML)

- Initial State
- State
- Final State
- Transition
- [event] Event (Guard / Action)
- Timeout Transition

TOOLS YOU CAN USE

- diagrams.net (draw.io)
- Lucidchart
- Visual Paradigm Online
- PlantUML
- miro

TODAY'S SPRINT GOAL

Model the interactions and state evolution for the selected use case using Sequence and State Diagrams to better understand system behaviour.

QUESTIONS?

Let's clarify, explore and learn together!



No question is too small!
Your questions help everyone learn and grow.

Ask freely, challenge ideas, and let's build understanding **together!**