System Modeling in the textbook

- Context models
- Interaction models
- Structural models
- Behavioral models
- Model-driven engineering

System Modeling

- System modeling is
  - the process of developing abstract models of a system,
  - each presenting a different view or perspective of that system.

- System models are Abstract
  - Not an alternate representation

System Perspectives

Different perspectives presented by models:

- **external**: context or environment of system
- **interaction**: between system and environment, or between components
- **structural**: organization of the system, or structure of data
- **behavioral**: dynamic behavior, how the system responds to events
System Modeling

- Notation to represent the models:
  - Graphical/diagrams (UML=Unified Modeling Language)
  - Formal/mathematical (ch 12)
- Models of the new system are used in:
  - Requirements development
  - Design process
  - Model-driven engineering
- Precision, completeness: not always necessary

UML Diagrams

We’ll discuss these UML Diagrams

- **Activity diagrams**: the activities in a process.
- **Use case diagrams**: interactions between a system and its environment.
- **Sequence diagrams**: interactions between actors and the system and components.
- **Class diagrams**: classes in the system and the associations between these classes.
- **State diagrams**: how the system reacts to events.

5.1 Context Models

- Primarily an external perspective
  - shows how system is situated or involved in its context
- **Static View**
  - shows what other systems it will interact with
- **Dynamic View**
  - shows how it is involved in business processes

Simple Context Model

- **Static view**
- **Used to define system boundaries**
  - what is done by new system, manually, or by another system
  - stakeholders must decide early
- **Simple context model**:
  - Boxes show each of the systems involved
  - Lines show interaction between systems
  - Overly simplified architectural model (ch 6)
  - Technically NOT a UML diagram
**Fig 5.1: The context of the MHC-PMS**

- **System** Management reporting system
- **System** Patient record system
- **System** Admissions system
- **System** MHC-PMS
- **System** Prescriptions system
- **System** Appointments system

**Note:** <<system>> is an example of a “stereotype” in UML. A mechanism to categorize an element in some way.

**Process Model**

- **Dynamic view**
- Shows how system is used in business processes
- **UML Activity diagram**
  - Shows activity and flow of control

  - **Filled circle:** start
  - **Filled concentric circle:** finish
  - **Rounded rectangles:** activities
  - **Rectangles:** other objects (the different systems in fig 5.2)
  - **Diamonds:** branch (and merge)
  - **Guards:** condition under which flow is taken out of branch
  - **Solid bar:** activity coordination/concurrency control (fork, join)

**Fig 5.2: Process model of involuntary detention**

**Example of a UML Activity diagram**

**Note:** This diagram is missing one branch and 2 merge diamonds.

**5.2 Interaction Models**

- Represents interactions
  - between system and environment or users
  - between components

- Uses:
  - user and system: developing requirements
  - system components: help to understand flow of control in an object oriented system

- Use Cases: user - system interactions

- Sequence Diagrams: components and external actors in more detail
5.2.1 Use Case Modeling

- Main purpose: requirements elicitation + analysis
- Overview of one discrete user/system interaction
  - Focusing on one goal of the actor
- Diagram components:
  - **stick figure**: actor (user or system)
  - **ellipse**: named interaction (verb-noun)
  - **line**: indicates involvement in interaction
- Diagram is supplemented with further details
  - textual description
  - structured description (form/template/table)
  - sequence diagram(s)

**Fig 5.3: Transfer data use case**

*Example of a UML Use case diagram*

<table>
<thead>
<tr>
<th>Actor</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Medical receptionist</td>
<td>Transfer data</td>
</tr>
<tr>
<td>Patient record system</td>
<td></td>
</tr>
</tbody>
</table>

Note: arrows are not part of UML, but shows direction of data flow

Note: primary actor on left, supporting actor on right

**Fig 5.4: Tabular description of Transfer data use case**

<table>
<thead>
<tr>
<th>MHC-PMS: Transfer data</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Actors</strong></td>
</tr>
<tr>
<td><strong>Description</strong></td>
</tr>
<tr>
<td><strong>Data</strong></td>
</tr>
<tr>
<td><strong>Stimulus</strong></td>
</tr>
<tr>
<td><strong>Response</strong></td>
</tr>
<tr>
<td><strong>Comments</strong></td>
</tr>
</tbody>
</table>

**Fig 5.5: Use cases involving Medical Receptionist**

*A composite use case diagram: all interactions involving a given actor*
5.2.2 Sequence Diagram

- Models the interactions between actors and objects within a system in some detail
- Shows the sequence of interactions in a given use case
- Diagram notes:

Read sequence from top to bottom

- **objects and actors**: listed across top with dotted lines going down
- **boxes on dotted line**: lifetime of object (in this interaction)
- **dotted arrows between lines from objects**: interactions
- **annotations on arrows**: calls to objects with parameters, return values
- **box named alt with conditions in brackets**: for branching/alternatives

### Sequence Diagram Uses

- **Requirements Development**:
  - Leave out detail, so as not to constrain developers
  - For example:
    - Minimal sequence diagram: only two components: user and system
    - Use to show sequence of interactions between user and system

- **Implementation**:
  - Details are required:
  - Interfaces to method calls between objects
  - Source of parameters in method calls