Chapter 5 – System Modeling

Lecture 1

Topics covered

- Wk 1: ----
- Wk 2: Introduction to SE & Ethics
- Wk 3: Software Process Models and dealing with change
- Wk 4: Agile Methods (1)
- Wk 5: Agile Methods (2)
- Wk 6: Requirements Engineering
- Wk 7: System Modeling & Revision
- Wk 8: Mid-term Exam

Topics covered

- Context models
- Interaction models
- Structural models
System modeling

- System modeling is the process of developing abstract models of a system, with each model presenting a different view or perspective of that system.
- System modeling has now come to mean representing a system using some kind of graphical notation, which is now almost always based on notations in the Unified Modeling Language (UML).
- System modelling helps the analyst to understand the functionality of the system and models are used to communicate with customers.

Existing and planned system models

- Models of the existing system are used during requirements engineering. They help clarify what the existing system does and can be used as a basis for discussing its strengths and weaknesses. These then lead to requirements for the new system.
- Models of the new system are used during requirements engineering to help explain the proposed requirements to other system stakeholders. Engineers use these models to discuss design proposals and to document the system for implementation.
- In a model-driven engineering process, it is possible to generate a complete or partial system implementation from the system model.

System perspectives

- An external perspective, where you model the context or environment of the system.
- An interaction perspective, where you model the interactions between a system and its environment, or between the components of a system.
- A structural perspective, where you model the organization of a system or the structure of the data that is processed by the system.
- A behavioral perspective, where you model the dynamic behavior of the system and how it responds to events.
UML diagram types

- Activity diagrams, which show the activities involved in a process or in data processing.
- Use case diagrams, which show the interactions between a system and its environment.
- Sequence diagrams, which show interactions between actors and the system and between system components.
- Class diagrams, which show the object classes in the system and the associations between these classes.
- State diagrams, which show how the system reacts to internal and external events.

Use of graphical models

- As a means of facilitating discussion about an existing or proposed system
  - Incomplete and incorrect models are OK as their role is to support discussion.
- As a way of documenting an existing system
  - Models should be an accurate representation of the system but need not be complete.
- As a detailed system description that can be used to generate a system implementation
  - Models have to be both correct and complete.

Context models

- Context models are used to illustrate the operational context of a system - they show what lies outside the system boundaries.
- Social and organisational concerns may affect the decision on where to position system boundaries.
- Architectural models show the system and its relationship with other systems.
System boundaries

- System boundaries are established to define what is inside and what is outside the system.
  - They show other systems that are used or depend on the system being developed.
- The position of the system boundary has a profound effect on the system requirements.
- Defining a system boundary is a political judgment
  - There may be pressures to develop system boundaries that increase / decrease the influence or workload of different parts of an organization.

The context of the MHC-PMS

Process perspective

- Context models simply show the other systems in the environment, not how the system being developed is used in that environment.
- Process models reveal how the system being developed is used in broader business processes.
- UML activity diagrams may be used to define business process models.
Interaction models

- Modeling user interaction is important as it helps to identify user requirements.
- Modeling system-to-system interaction highlights the communication problems that may arise.
- Modeling component interaction helps us understand if a proposed system structure is likely to deliver the required system performance and dependability.
- Use case diagrams and sequence diagrams may be used for interaction modeling.

Use case modeling

- Use cases were developed originally to support requirements elicitation and now incorporated into the UML.
- Each use case represents a discrete task that involves external interaction with a system.
- Actors in a use case may be people or other systems.
- Represented diagrammatically to provide an overview of the use case and in a more detailed textual form.
Transfer-data use case

A use case in the MHC-PMS

Medical receptionist ➔ Transfer data ➔ Patient record system

Tabular description of the 'Transfer data' use-case

| Describe: MHC-PMS: Transfer data
| Actors: Medical receptionist, patient record system (PRS)
| Description: A receptionist may transfer data from the MHC-PMS to a general patient record database that is maintained by a health authority. The information transferred may either be updated personal information (address, phone number, etc.) or a summary of the patient’s diagnosis and history.
| Note: Patient’s personal information, treatment summary
| Stimulus: User command issued by medical receptionist
| Response: Confirmation that PRS has been updated
| Comments: The receptionist must have appropriate security permissions to access the patient information and the PRS.

Use cases in the MHC-PMS involving the role ‘Medical Receptionist’
Sequence diagrams

- Sequence diagrams are part of the UML and are used to model the interactions between the actors and the objects within a system.
- A sequence diagram shows the sequence of interactions that take place during a particular use case or use case instance.
- The objects and actors involved are listed along the top of the diagram, with a dotted line drawn vertically from these.
- Interactions between objects are indicated by annotated arrows.

Sequence diagram for View patient information

Sequence diagram for Transfer Data
Structural models

- Structural models of software display the organization of a system in terms of the components that make up that system and their relationships.
- Structural models may be static models, which show the structure of the system design, or dynamic models, which show the organization of the system when it is executing.
- You create structural models of a system when you are discussing and designing the system architecture.

Class diagrams

- Class diagrams are used when developing an object-oriented system model to show the classes in a system and the associations between these classes.
- An object class can be thought of as a general definition of one kind of system object.
- An association is a link between classes that indicates that there is some relationship between these classes.
- When you are developing models during the early stages of the software engineering process, objects represent something in the real world, such as a patient, a prescription, doctor, etc.

UML classes and association

![Patient class diagram]

- In the UML class diagram, the relationship between the Patient class and Patient record class is shown with a one-to-one association.
The Consultation class

Key points

- A model is an abstract view of a system that ignores system details. Complementary system models can be developed to show the system’s context, interactions, structure and behavior.
- Context models show how a system that is being modeled is positioned in an environment with other systems and processes.
- Use case diagrams and sequence diagrams are used to describe the interactions between users and systems in the system being designed. Use cases describe interactions between a system and external actors; sequence diagrams add more information to these by showing interactions between system objects.
- Structural models show the organization and architecture of a system. Class diagrams are used to define the static structure of classes in a system and their associations.
Revision Topics

- Exam will cover lectures:
  - Wk 2: Introduction to SE & Ethics
  - Wk 3: Software Process Models and dealing with change
  - Wk 4&5: Agile Methods
  - Wk 6: Requirements Engineering
  - Wk 7: System Modeling

- Let's recap each week….

Introduction to SE & Ethics

- What is software engineering
- It's importance
- Software process activities & issues affecting software
- Application types
- SE fundamentals & SE on the Web
- SE ethics
- No need to revise from the case studies

Software Process Models & change

- Software process models
  - What is the software process & model?
  - Plan driven vs Agile
  - Waterfall, incremental & reuse
- Process activities
  - Specification, development, validation & evolution
- Coping with change
  - Reducing the cost of change
  - Prototyping
  - Incremental delivery
Agile Methods

- Agile methods – where used and problems
- Plan-driven and agile development
- Extreme programming – using agile principles, requirements scenarios, testing in XP, pair programming
- Agile project management - scrum
- Scaling agile methods

Requirements Engineering

1. What is a requirement? How can it be used?
2. User requirements vs system requirements
3. Functional and non-functional requirements
4. The software requirements document - users, structure, ways of writing
5. Requirements discovery
   1. Stakeholders
   2. Interviewing
   3. Scenarios
   4. Use cases & ethnography

System Modeling

- A model is an abstract view of a system that ignores system details. Complementary system models can be developed to show the system’s context, interactions, structure and behaviour.
- Context models show how a system that is being modeled is positioned in an environment with other systems and processes.
- Use case diagrams and sequence diagrams are used to describe the interactions between users and systems in the system being designed. Use cases describe interactions between a system and external actors; sequence diagrams add more information to these by showing interactions between system objects.
- Structural models show the organization and architecture of a system. Class diagrams are used to define the static structure of classes in a system and their associations.