
Object-Oriented Analysis and Design using JAVA

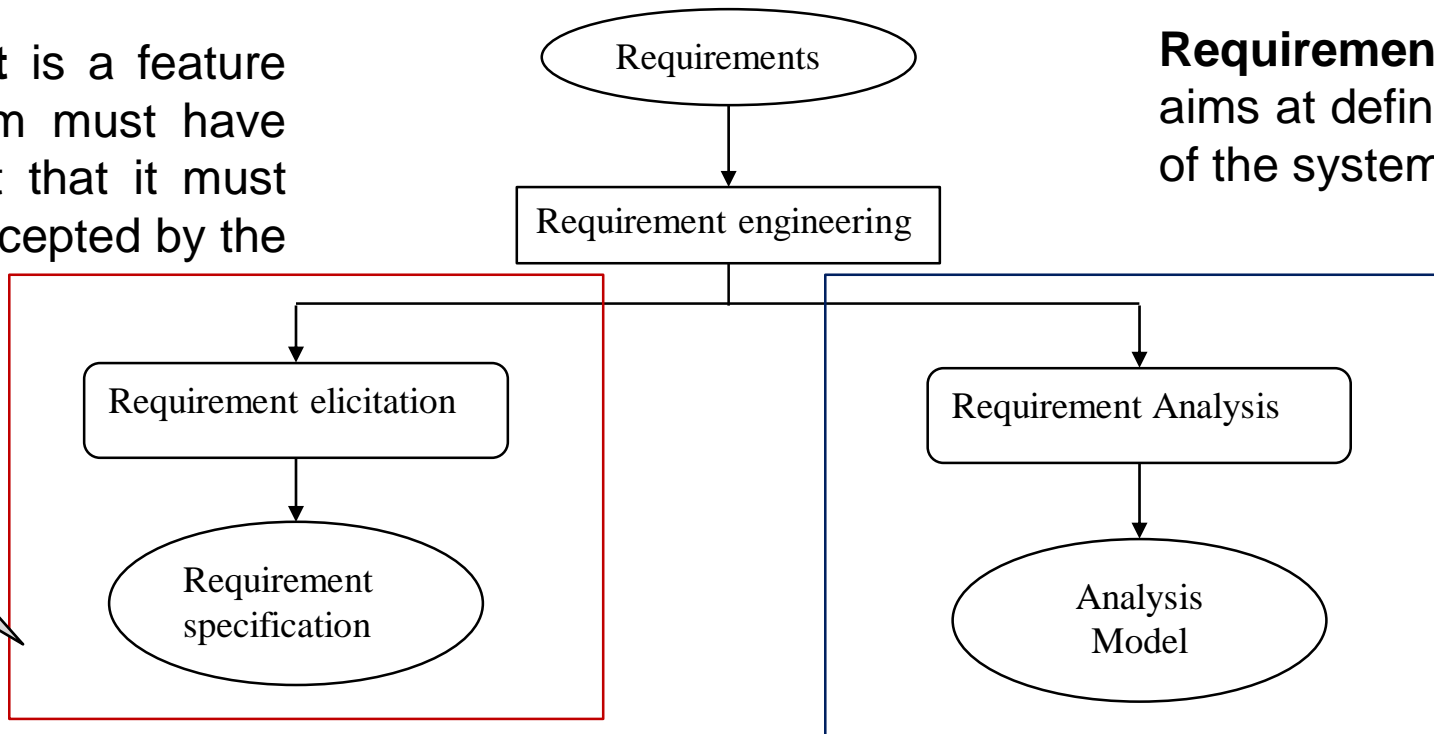
B.Tech (CSE/IT) 5th SEM
2021-2022

Lecture-9: Requirement Analysis

Introduction

A requirement is a feature that the system must have or a constraint that it must satisfy to be accepted by the client.

Covered in previous lecture

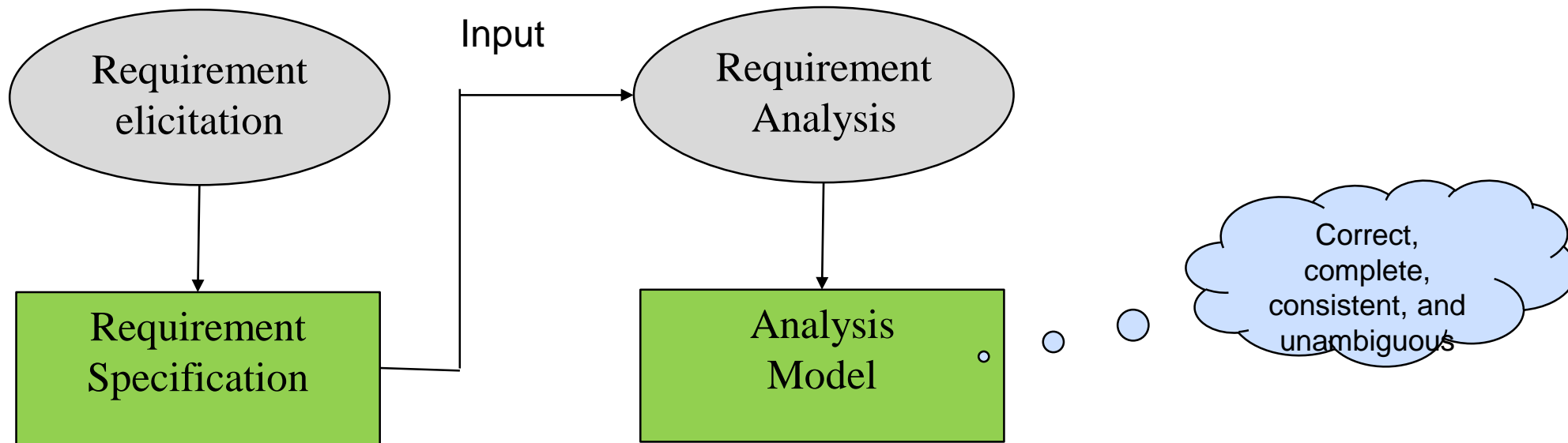


Specification of the system that the client understands

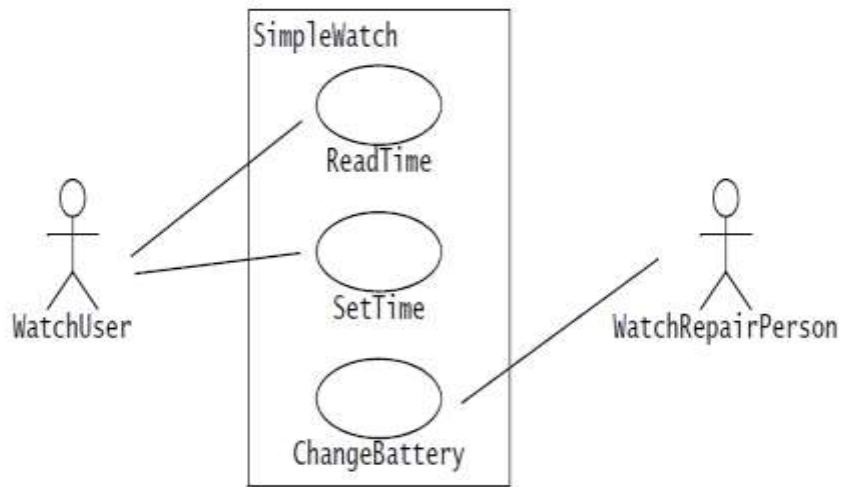
An analysis model that the developers can unambiguously interpret

Requirements engineering aims at defining the requirements of the system under construction

Will be covered in this lecture



Developers formalize the requirements specification produced during requirements elicitation and examine in more detail boundary conditions and exceptional cases. Developers validate, correct and clarify the requirements specification if any errors or ambiguities are found. The client and the user are usually involved in this activity when the requirements specification must be changed and when additional information must be gathered.



Requirement
specification

Requirement
analysis model

In object-oriented analysis, developers build a model describing the application domain.

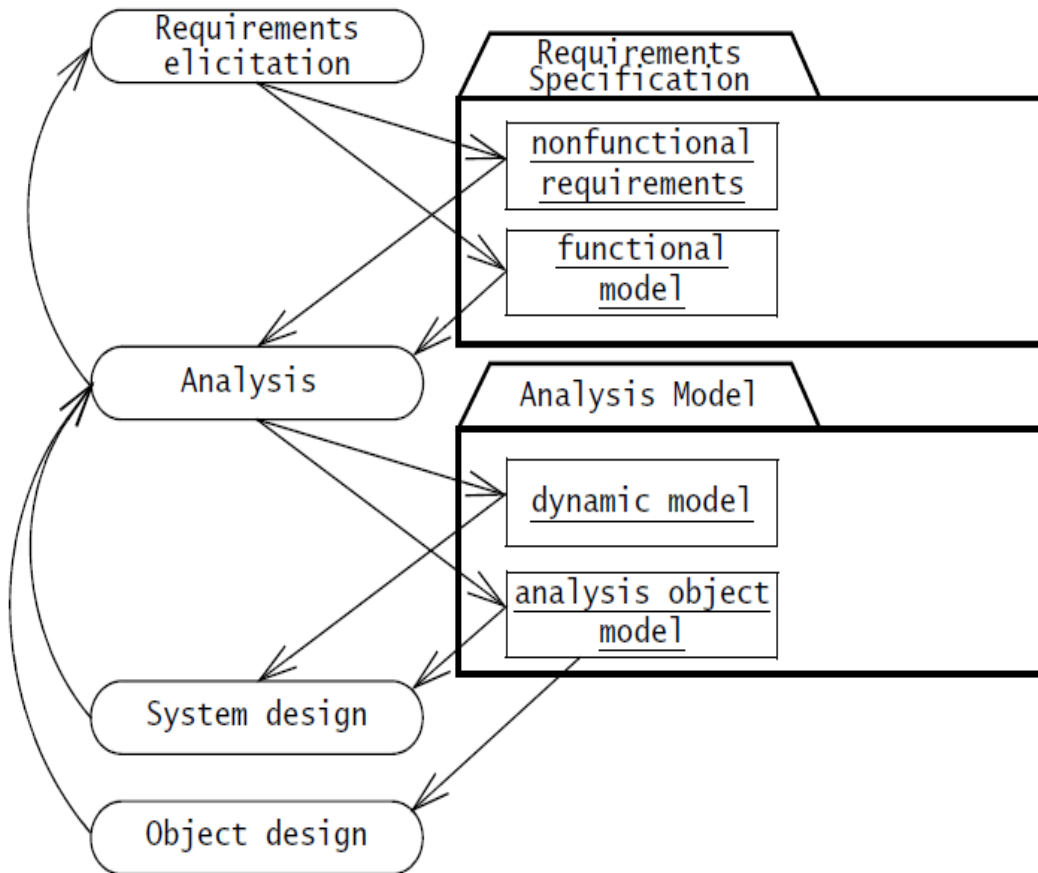
For example, the analysis model of a watch describes how the watch represents time:

Does the watch know about leap years? Does it know about the day of the week?

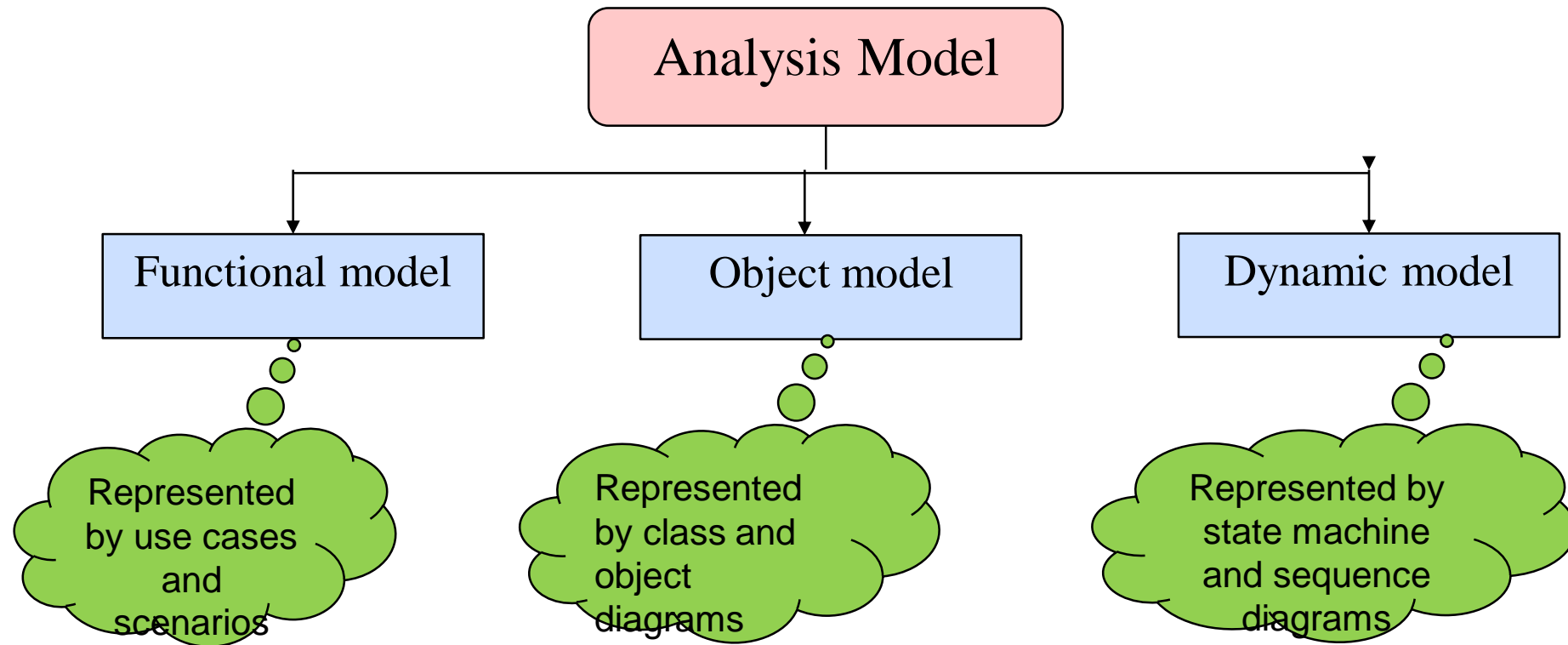
Does it know about the phases of the moon?

The analysis model is then extended to describe how the actors and the system interact to manipulate the application domain model:

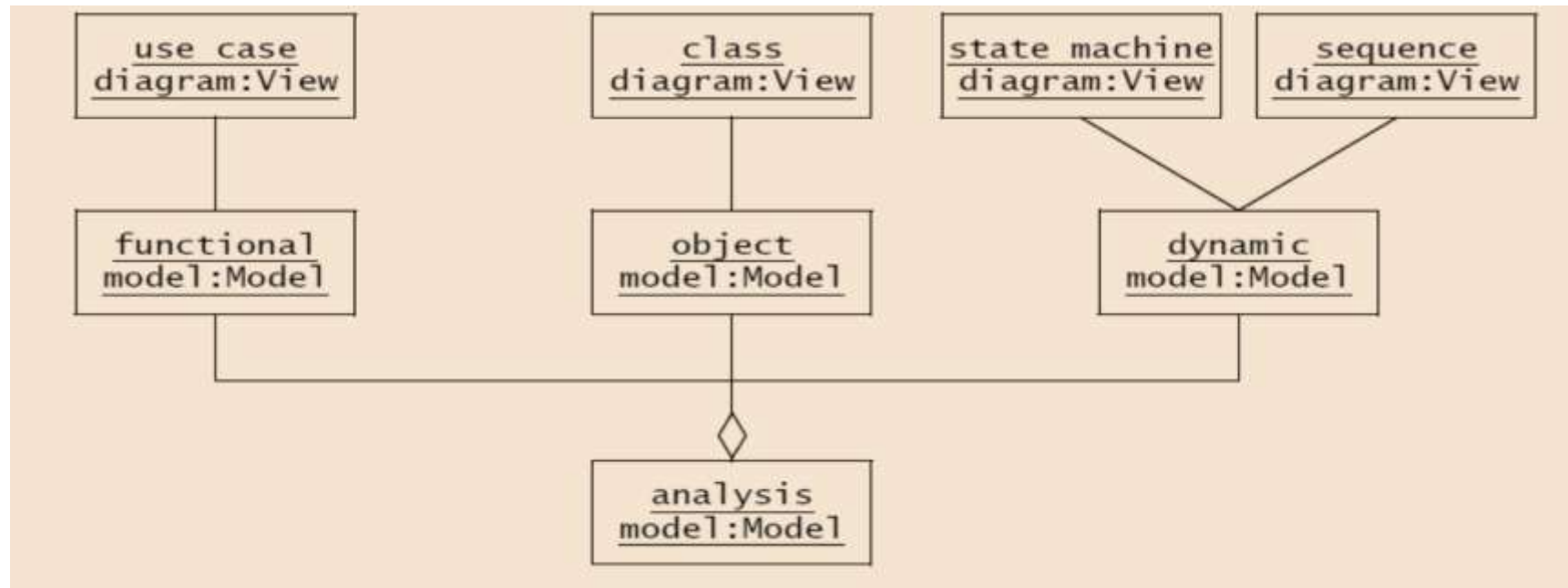
How does the watch owner reset the time? How does the watch owner reset the day of the week?



Analysis focuses on producing a model of the system, called the analysis model, which is correct, complete, consistent, and verifiable. Analysis is different from requirements elicitation in that developers focus on structuring and formalizing the requirements elicited from users. This formalization leads to new insights and the discovery of errors in the requirements. As the analysis model may not be understandable to the users and the client, developers need to update the requirements specification to reflect insights gained during analysis, then review the changes with the client and the users. In the end, the requirements, however large, should be understandable by the client and the users.



How to refine the functional model created in requirement specification and derive the object and the dynamic model. This leads to a more precise and complete specification as details are added to the analysis model.



The analysis model is composed of the functional model, the object model, and the dynamic model. In UML, the functional model is represented with use case diagrams, the object model with class diagrams, and the dynamic model with state machine and sequence diagrams

Analysis Activities: From Use Cases to Objects

We describe the activities that transform the use cases and scenarios produced during requirements elicitation into an analysis model. Analysis activities include:

- Identifying Entity Objects
- Identifying Boundary Objects
- Identifying Control Objects
- Mapping Use Cases to Objects with Sequence Diagrams
- Modeling Interactions among Objects with CRC Cards
- Identifying Associations
- Identifying Aggregates
- Identifying Attributes
- Modeling State-Dependent Behavior of Individual Objects
- Modeling Inheritance Relationships
- Reviewing the Analysis Model

Key references

Bernd Bruegge & Allen H. Dutoit - Object-Oriented Software Engineering: Using UML, Patterns, and Java

Thank You