FPT UNIVERSITY

SOFTWARE DESIGN (SWD392)

CH01 – INTRODUCTION

Software Modeling

Modelling is the designing of SW applications before coding

- Models are built and analysed prior to the implementation of the system, and are used to direct the subsequent implementation.
- A graphical modelling language such as UML helps in developing, understanding, and communicating the different views.

OO Methods & the UML

- OO concepts
  - Address fundamental issues of software modifiability, adaptation, and evolution
  - Methods based on concepts of information hiding, classes, and inheritance
- UML ~ Unified Modeling Language (UML)
  - Developed to provide a standardized graphical language and notation for describing object-oriented models.
  - UC modeling: functional requirements
  - Static modeling: structural view of the system
  - Dynamic modeling: behavioral view of the system

Software Architectural Design

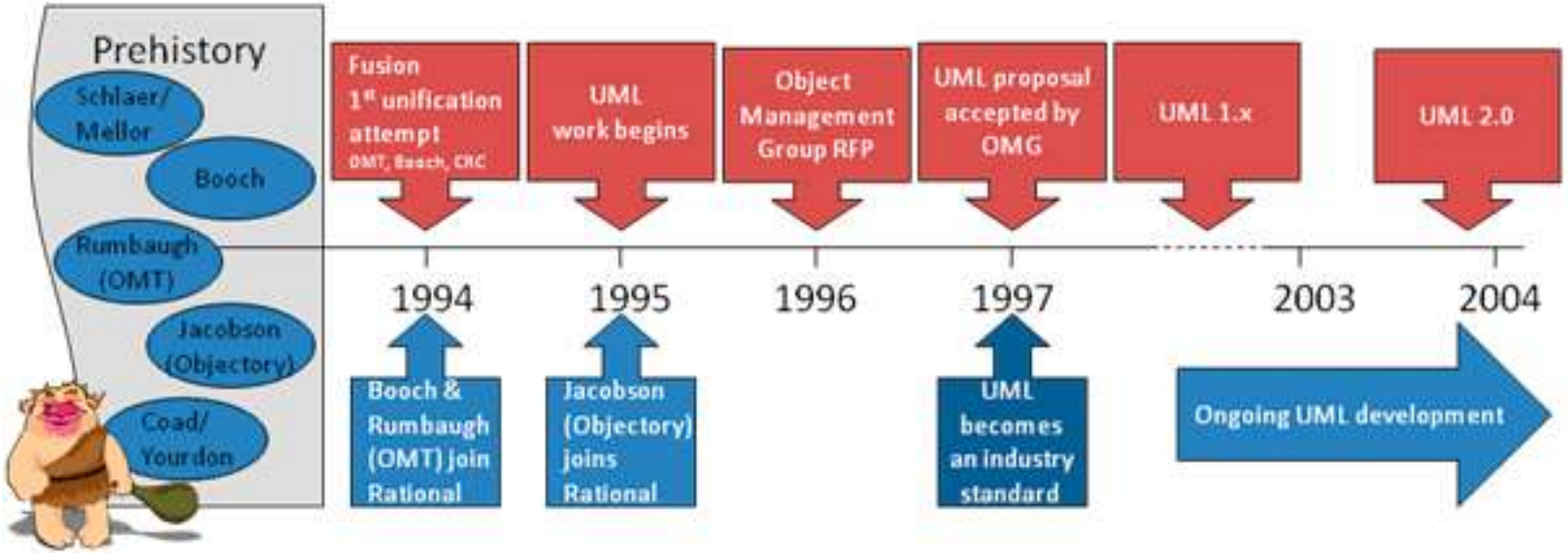
- Sometimes referred to as a high-level design
  - Describe decomposition of the software system into subsystems.
  - Address software quality attributes
- Starting point for the detailed design and implementation (decomposition of subsystems into modules or components).

Method & Notation

- Software design notation: graphically, textually
- A **software design concept** is a fundamental idea that can be applied to designing a system - ie information hiding.
- A **software design strategy** is an overall plan and direction for developing a design – i.e OO decomposition
- **Software structuring criteria** are heuristics or guidelines used to help a designer in structuring a software system into its components. i.e: object structuring criteria provide guidelines for decomposing the system into objects.
- A **software design method** is a systematic approach that describes the sequence of steps to follow in order to create a design, given the software requirements of the application.

COMET & UML

- The Collaborative Object Modeling and Design Method, or **COMET**, uses the UML notation to describe the design
- UML: a standard modeling language and notation for describing object-oriented designs



- Latest version 2.5.1, published Dec/2017

Multiple Views of SW Architecture

- Use case view: use case diagrams
- Static view: class diagrams
- Dynamic interaction view: objects & interaction messages between them
- Dynamic state machine view: state machine / state chart diagram
- Structural component view: structured class diagrams
- Dynamic concurrent view: concurrent communication diagrams
- Deployment view: deployment diagrams


Q
A