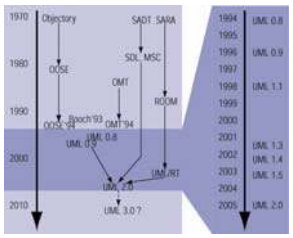
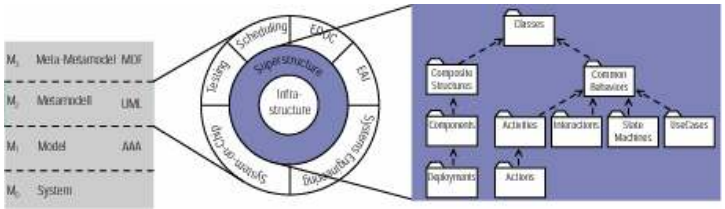
Unified Modeling Language 2.0

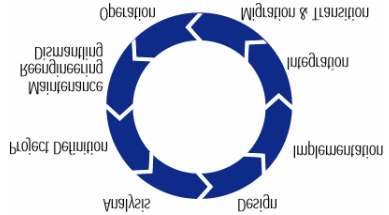
Part 1 – Introduction

Prof. Dr. Harald Störrle
University of Innsbruck
mgm technology partners

Dr. Alexander Knapp
University of Munich

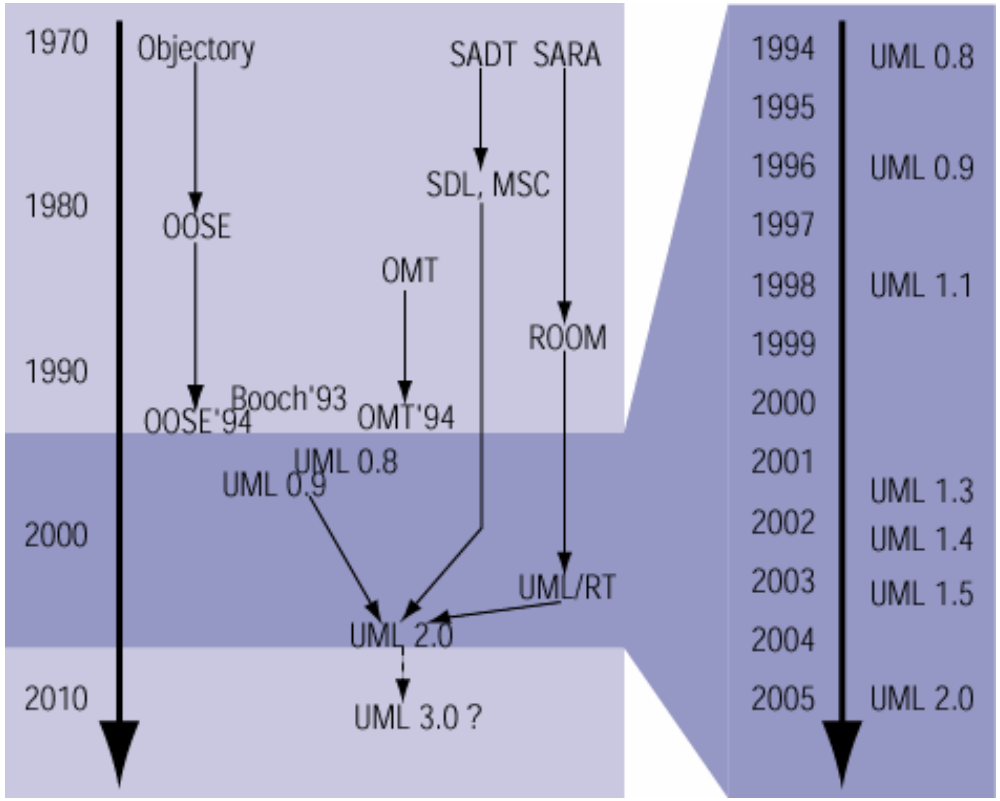






1 – Introduction
History and Predecessors

- The UML is the “lingua franca” of software engineering.
- It subsumes, integrates and consolidates most predecessors.
- Through the network effect, UML has a much broader spread and much better support (tools, books, trainings etc.) than other notations.
- The transition from UML 1.x to UML 2.0 has
  - resolved a great number of issues;
  - introduced many new concepts and notations (often feebly defined);
  - overhauled and improved the internal structure completely.
- While UML 2.0 still has many problems, it is much better than what we ever had before.

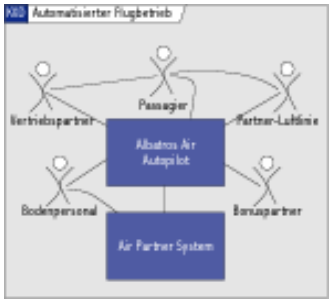


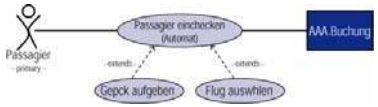
current version (“the standard”)
formal/05-07-04 of August ‘05

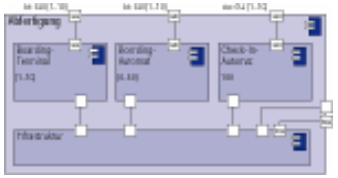
1 – Introduction
Usage Scenarios

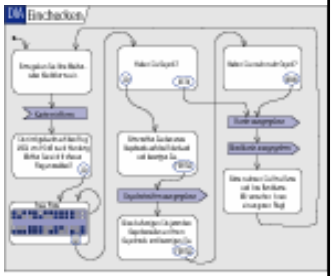
- UML has not been designed for specific, limited usages.
- There is currently no consensus on the role of the UML:
  - Some see UML only as tool for sketching class diagrams representing Java programs.
  - Some believe that UML is "*the prototype of the next generation of programming languages*".
- UML is a really a system of languages ("notations", "diagram types") each of which may be used in a number of different situations.
- UML is applicable for a multitude of purposes, during all phases of the software lifecycle, and for all sizes of systems – to varying degrees.

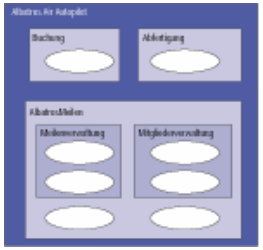
1 – Introduction
Usage Scenarios











Analysis

Design

Project Definition

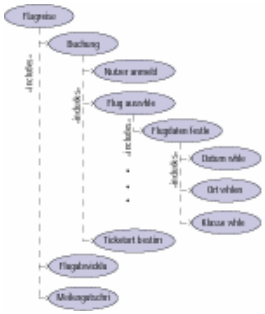
Implementation

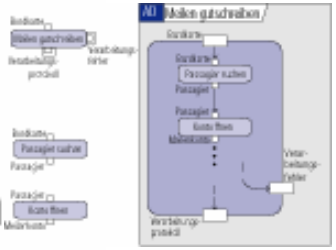
Maintenance
Reengineering
Dismantling

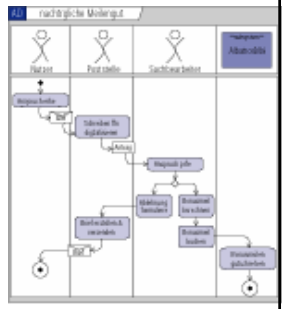
Integration

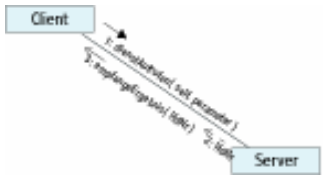
Operation

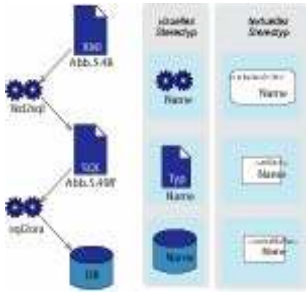
Migration & Transition

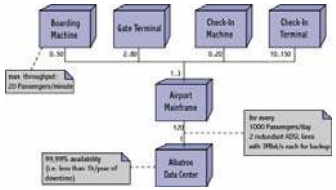


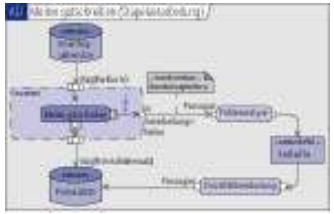


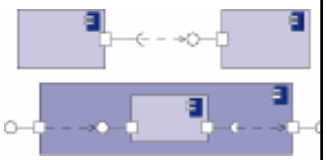


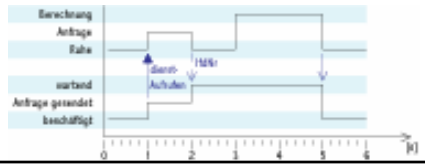












1 – Introduction
Diagram types in UML 2

UML is a coherent system of languages rather than a single language. Each language has its particular focus.

Structure	Class Diagram		static structure (generic/snapshot)
	Composite Structure Diagram		logical system structure
	Component Diagram		physical system structure
	Deployment Diagram		computing infrastructure / deployment
	Package Diagram		containment hierarchy
Behavior	Use Case Diagram		abstract functionality
	Activity Diagram		controlflow and dataflow
	Interaction Sequence Diagram	*interactions by message exchange*	message exchange over time
	Communication Diagram		structure of interacting elements
	Timing Diagram		coordinated state change over time
	Interaction Overview Diagram		flows of interactions
	State Machine Diagram		event-triggered state change