Aitecon

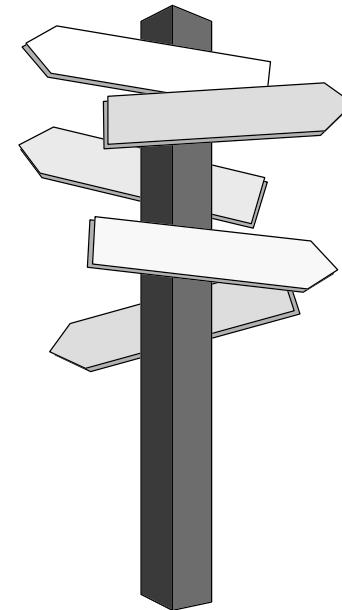# UNIFIED MODELING LANGUAGE

Introduction

Radovan Červenka, October 1998 (version 0.04)

Context

- **Introduction**

  - Generic Mechanisms

  - Use Case Modeling

  - Static Structure Modeling

  - Dynamic Behavior Modeling

  - Interaction Modeling

  - Physical Structure Modeling

  - General Extension Mechanisms
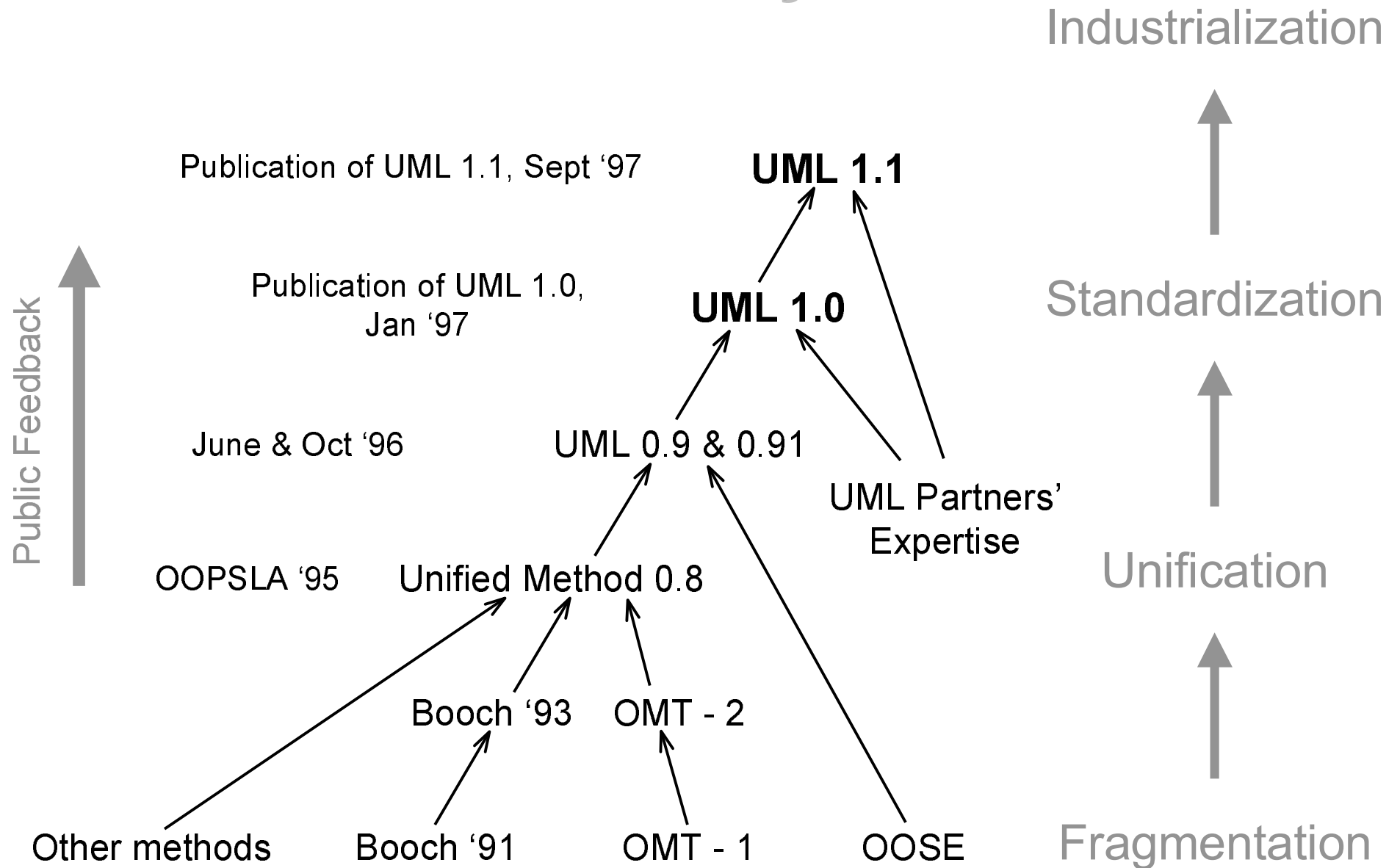
UML

→ a language for **specifying**, **constructing**, **visualizing** and **documenting** the artifacts of a software-intensive system

∗ also for *Business Modeling* and others non-software systems

Authors: Grady Booch, Jim Rumbaugh and Ivar Jacobson
(all from Rational Software Corp.)

Stems out from: *Booch*, *OMT*, *OOSE* (and others)

History

Industrialization

Publication of UML 1.1, Sept '97   **UML 1.1**

Publication of UML 1.0,
Jan '97   **UML 1.0**

Standardization

June & Oct '96   UML 0.9 & 0.91

UML Partners'
Expertise

Unification

OOPSLA '95   Unified Method 0.8

Public Feedback

Booch '93   OMT - 2

Other methods   Booch '91   OMT - 1   OOSE

Fragmentation

Goals

- ready-to-use, expressive visual modeling language

- extensibility and specialization mechanisms to extend the core concepts

- independent of particular programming languages and development process

- formal basis for understanding the modeling language

- encourage the growth of the OO tools (*CASE*) market

- support higher-level development concepts such as collaboration, frameworks, patterns, and components

- integrate best practices

Outside the Scope of UML

Programming Languages

$\rightarrow$ the UML is a modeling language, not programming

- its aim is not to capture all necessary constructs of programming languages

Tools

$\rightarrow$ the UML defines a semantic metamodel, not an tool interface, storage, or run-time model

  * the UML documents do include some tips to tool vendors on implementation choices, but do not address everything needed

Process

$\rightarrow$ the UML is intentionally process independent, and defining a standard process was not a goal of the UML

- a common language for project artifacts, developed in the context of different processes

Artifacts

UML Semantics

$\rightarrow$ description of the *UML Metamodel*, i.e. a model defining the UML language

- Abstract syntax
- Well-formedness rules
- Semantics

UML Notation Guide
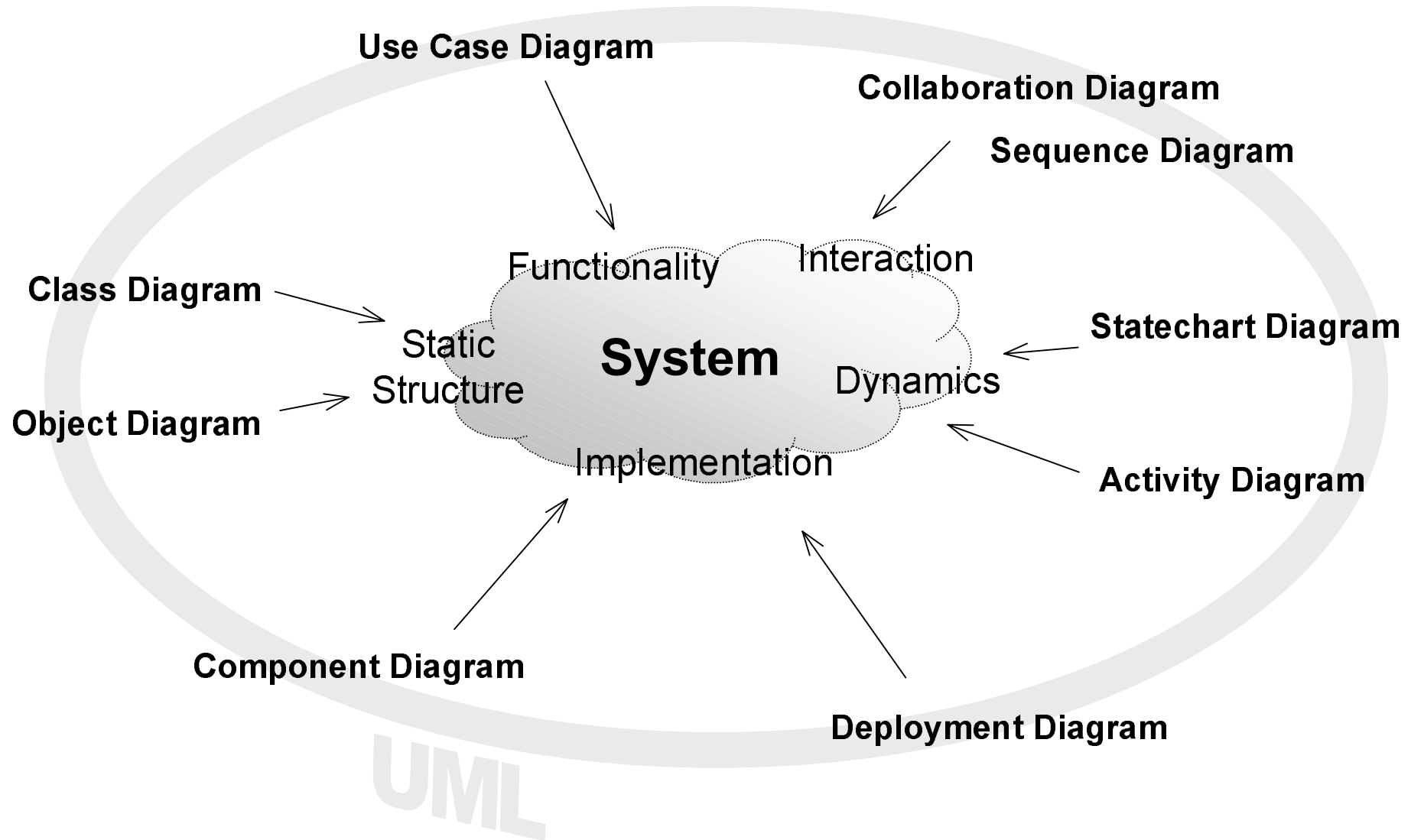
$\rightarrow$ graphical notation and textual syntax of "visual" elements of the UML language

UML Extensions

$\rightarrow$ language extensions for *Objectory Process* and *Business Engineering*

UML and Different System's Aspects

UML and SW Development Process

Phases

	Inception	Elaboration	Construction	Transition

Process Components

Requirements Capture

Analysis & Design

Implementation

Test

Supporting Components

Management

Environment

Deployment

Use Case and Static Structure Diagrams

Interaction, Static Structure, Statechart and Implementation Diagrams

Used all UML Diagrams

Used mainly Use Case Diagrams

Used Use Case Diagrams

Used mainly Implementation Diagrams

preliminary iteration(s)	iter. #1	iter. #2	iter. #n	iter. #n+1	iter. #n+2	iter. #m	iter. #m+1
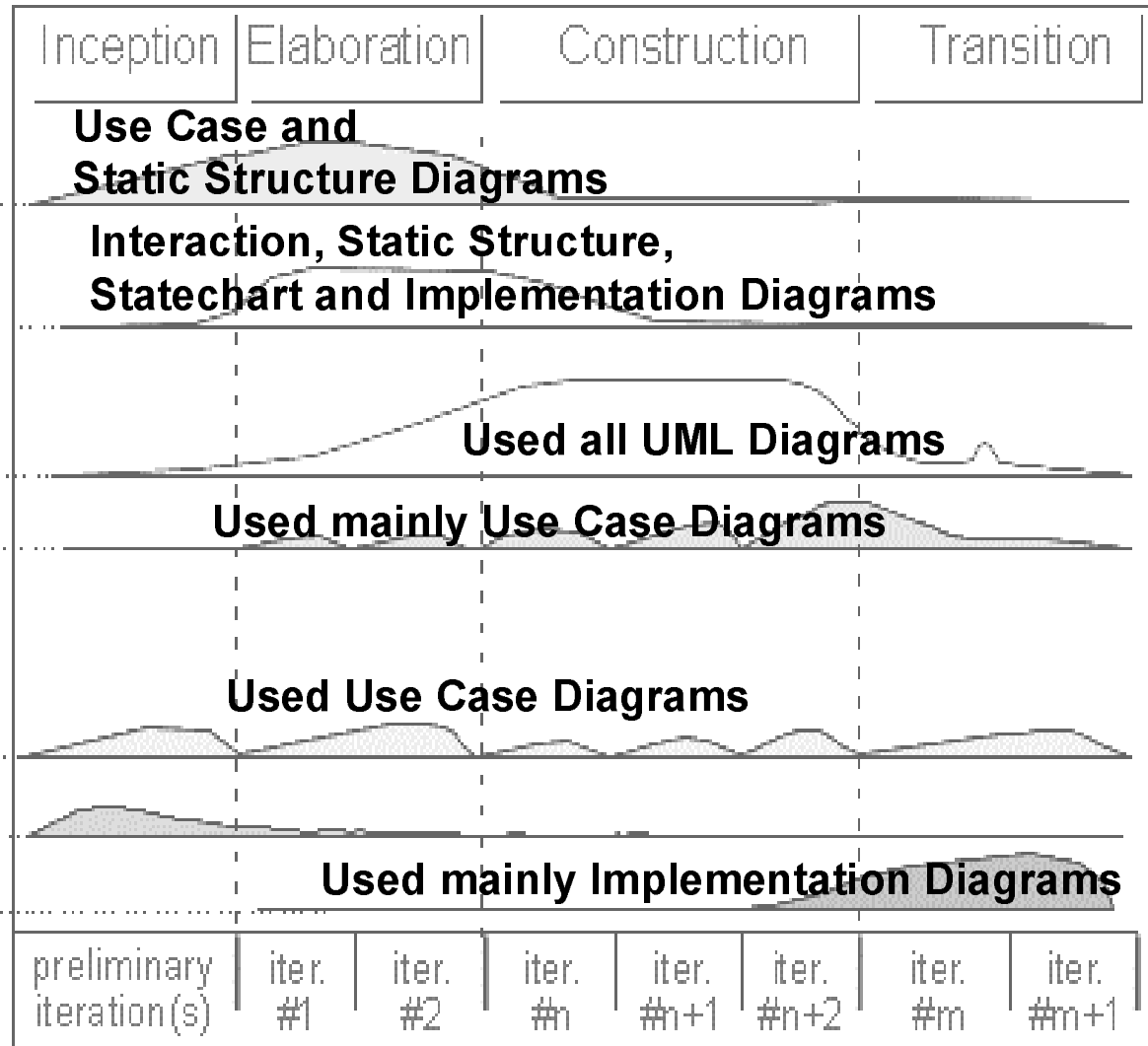
Iterations

Models Development

Requirements Capture

Analysis and Design

Use Case Model
[Functionality]

Interaction Model
[Use Case Realization]

...
[UI Description]

Static Structure Model
[Architecture]

...
[Non-functional Requirements]

Dynamic Model
[Object's Behavior]

Static Structure Model
[Domain Model]

Component Model
[Component Architecture]

Deployment Model
[HW Architecture]