Unified Modeling Language

Introduction

Radovan Cervenka

The **Unified Modeling Language** (**UML**) is a language for *specifying, constructing, visualizing* and *documenting* the artifacts of a software-intensive system.

- Can be used also for other domains, such as, business modeling, process engineering, and others non-software systems.

- Stems out from modeling languages: Booch, OMT, OOSE, and others.

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

- Now: UML is an OMG standard with several contributors.

History

Fragmentation

- 1991: Booch '91, OMT-1, OOSE, ...

- 1993: Booch '93, OMT-2

Unification

- 1995: Unified Method (OOPSLA '95)

- 1996: UML 0.9 and UML 0.91

Standardization and industrialization

- 1997: UML 1.0 and UML 1.1

- 1998-2003: several minor revisions in UML 1.3, 1.4, and 1.5

- 2005: ISO/IEC 19501:2005 Information technology–Open Distributed Processing–Unified Modeling Language (UML) Version 1.4.2

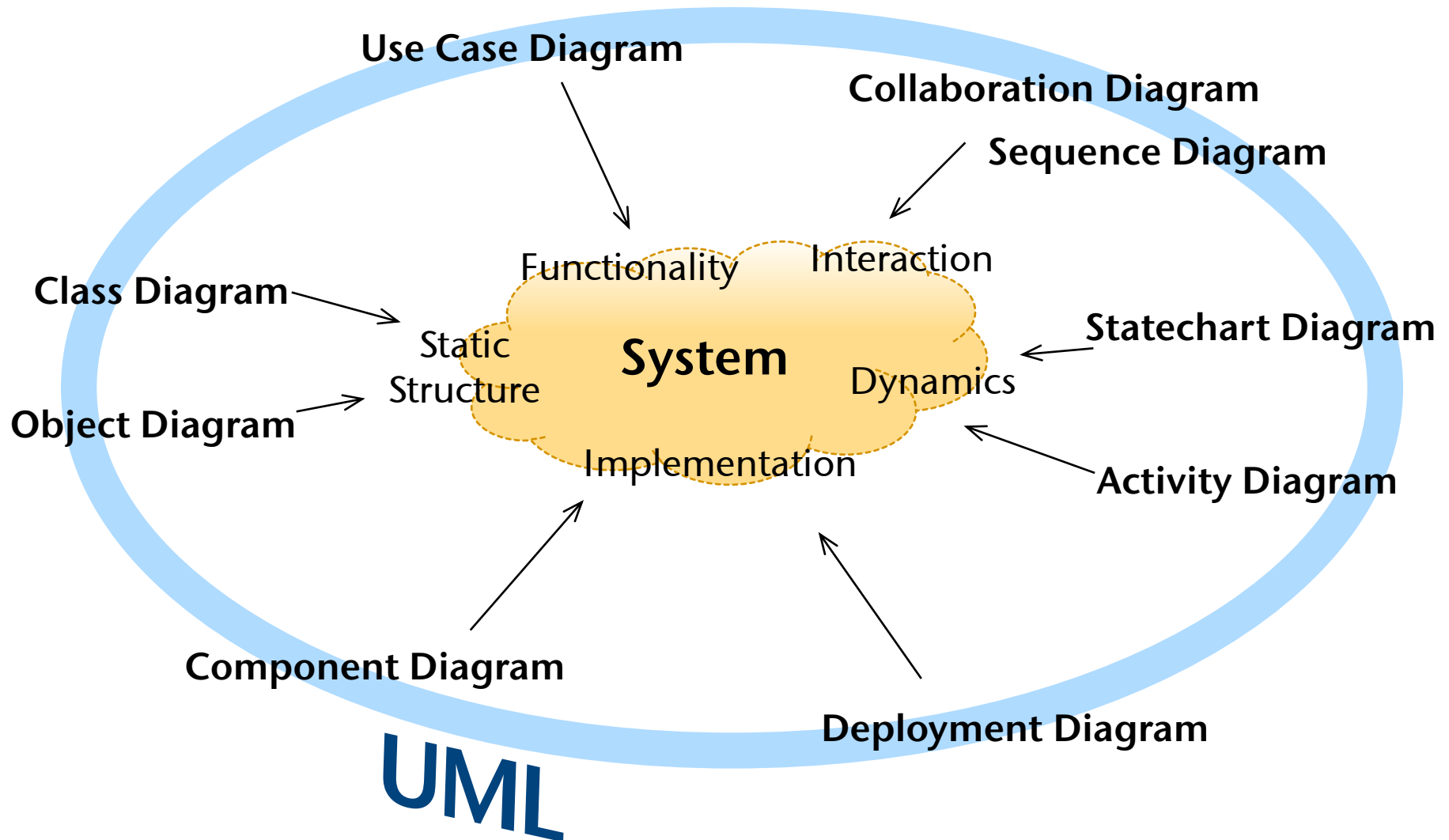- 2003-2005: UML 2.0

- ...

- 2011: UML 2.4.1

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.
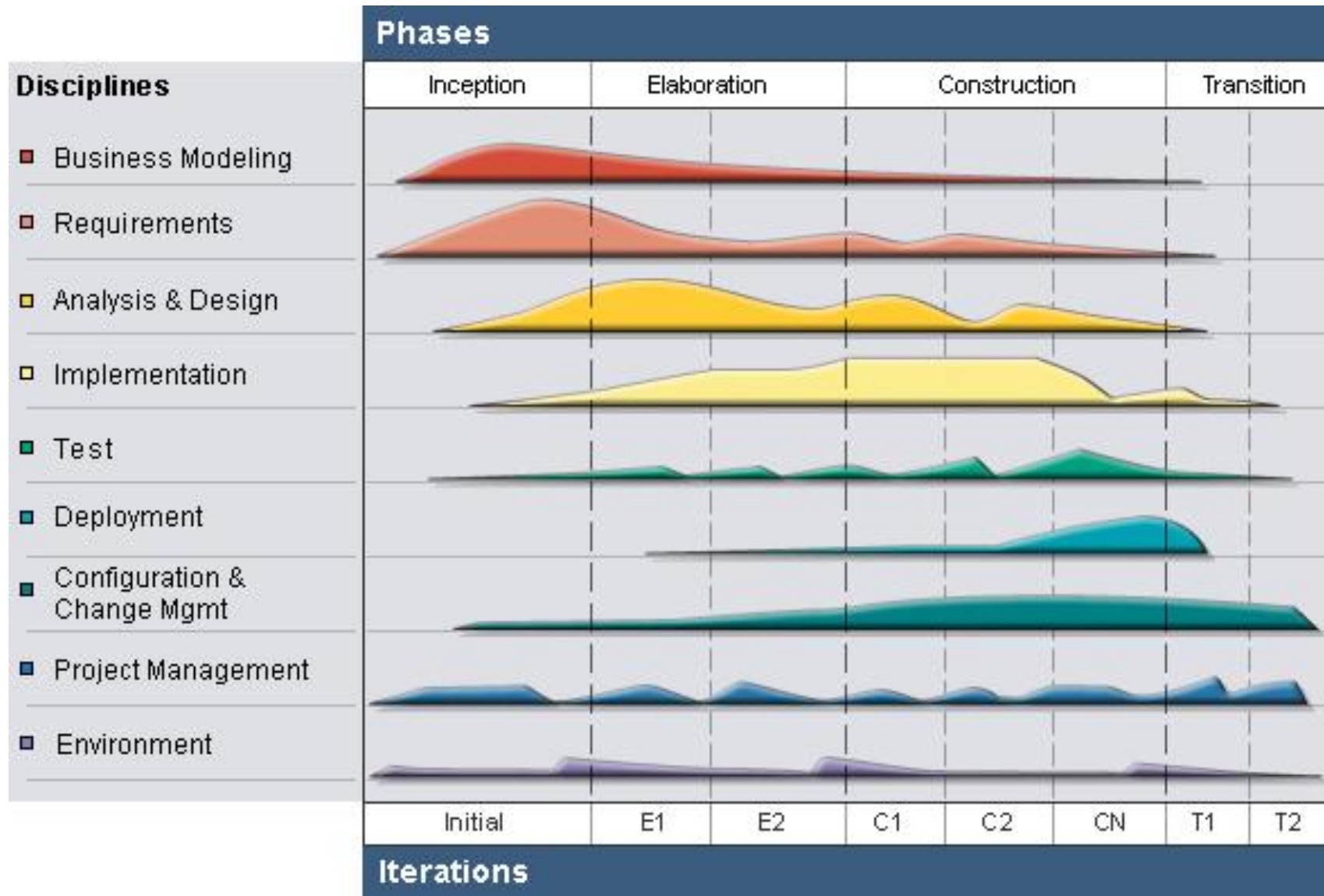
Outside the Scope of UML

- Programming Languages
    - UML is a modeling language, not programming.
    - Its aim is not to capture all necessary constructs of programming languages.

- Tools
    - UML defines a semantic metamodel, not an tool interface, storage, or run-time model.
        - The UML specification do include some tips to tool vendors on implementation choices, but do not address everything needed.

- Process
    - 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

- ***Unified Modeling Language: Infrastructure***
  - Defines the abstract foundational language constructs required for UML specification.
  - UML meta-metamodel (MOF layer M3).

- ***Unified Modeling Language: Superstructure***
  - Defines the UML language, i.e., abstract syntax (metamodel), notation, and semantics.
  - UML metamodel (MOF layer M2).

- Related OMG specifications:
  - Object Constraint Language (OCL)
  - XML Metadata Interchange (XMI)
  - Diagram Interchange (DI)
  - Human-Usable Textual Notation (HUTN)
  - Standardized UML profiles
  - Meta Object Facility (MOF)

Software Development Process

Disciplines

- Business Modeling
- Requirements
- Analysis & Design
- Implementation
- Test
- Deployment
- Configuration & Change Mgmt
- Project Management
- Environment

Phases

| Inception | Elaboration | Construction | Transition |

All UML diagrams

Use Case and Class diagrams

All UML diagrams

Used all UML diagrams

Used mainly Use Case and behavioral diagrams

Used Deployment diagrams

Used Use Case diagrams

Used mainly Component and Deployment diagrams

| Initial | E1 | E2 | C1 | C2 | CN | T1 | T2 |

Iterations

UML.Introduction | R. Cervenka