Architecture of Distributed Systems

A very short intro to UML

R. H. Mak
Goal of the slideset

- Architectural descriptions (ADs) use models to express various aspects of an architecture. Often these models are formulated in some kind of Architectural Description Language (ADL), of which UML is a very prominent one.

- This slide set is a very brief and incomplete introduction to UML, intended for students who have no prior exposure to the language or need a recap
  - it provides the basic concepts and vocabulary
  - it illustrates the appearance and usage of several kinds of UML-diagrams
  - it provides references to more elaborate tutorials
  - it will help you in making the homework assignments
Unified Modeling Language

• **A graphical language**
  • for visualizing, specifying, constructing and documenting artifacts of a software-intensive system
  • originated from unification of methods by Booch, Rumbaugh (OMT) and Jacobson (OOSE)
  • standardized by the OMG
  • with a well-defined syntax and semantics
    • including the possibility for user-defined extensions
  • with support for viewpoint libraries
    • e.g. Kruchten’s 4+1 views
  • with tool support for Model-Driven Architecture (MDA)
UML: terms and concepts

• **System**
  • A set of elements organized to accomplish a purpose and described by a set of models possible from different viewpoints
  • Can be decomposed into a set of subsystems

• **Model**
  • A simplification of reality, an abstraction of a system, created in order to better understand the system

• **View**
  • A projection of a model, which is seen from one perspective or vantage point and omits entities that are not relevant to this perspective
UML vocabulary

- **Things**
  - Structural, behavioral, grouping, annotational
- **Relationships**
  - Dependencies, associations, generalizations, realizations
- **Structural diagrams**
  - Class diagram, object diagram, component diagram, deployment diagram
- **Behavioral diagrams**
  - Use case diagram, interaction (sequence or collaboration) diagram, state-chart diagram, activity diagram
Structural things (1)

- Represent building blocks for modeling the static structure of the architecture
- Dynamic models contain separate building blocks
Structural things (2)

Collaboration

Chain of responsibility

extra compartments may be used to show contents

Use Case

Place order

Active Class

EventManager

q : EventQueue

suspend()

flush()

Component

spelling.java

Node

Server

extra compartments may be used to show contents
Relationships

- **Dependency**
  - name
  - source → target

- **Generalization**
  - discriminator
  - child → parent

- **Association**
  - name
  - multiplicity
  - navigation
  - end
  - role name
  - interface specifier
  - unfilled diamond for aggregation
  - filled diamond for composition

- **Symbols**
  - Association
  - Inheritance
  - Realization / Implementation
  - Dependency
  - Aggregation
  - Composition
Behavioral things (1)

- Entire model (sequence diagram): contains both building blocks and relationships
Behavioral things (2)
Diagrams (Model kinds)

• **Structural diagrams**
  • Class diagram
  • Object diagram
  • Component diagram
  • Deployment diagram

• **Behavioral diagrams**
  • Use case diagram
  • Interaction diagram (sequence & collaboration)
  • State machine diagram
  • Activity diagram

This is not all:
see book by Booch, Rumbaugh, Jacobson,
or tutorials on the WWW
UML version: Kruchten 4+1 view

- **Design View**
  - Classes, Interfaces, Collaborations

- **Implementation View**
  - Use Cases
  - Components

- **Use Case View**
  - K: scenarios

- **Process View**
  - Active Classes

- **Deployment View**
  - K: also physical

K: logical
RW: functional
K & RW: development
Examples

The next 3 slides contain some examples taken from the web to illustrate the look and usage of various kinds of UML-diagrams.

Acknowledgement:

The diagrams are respectively taken from

I.  https://www.tutorialspoint.com/uml/
II. https://www.uml-diagrams.org/
III. https://www.uml-diagrams.org/
Some examples I (look and usage)

- Interaction with environment: users, but also other systems
- Externally visible functionality/behavior
- Mostly relevant for logical/functional view, scenarios
- Relevant for a lot of stakeholders
  - customers, end users, acquirers, educators, operators, ...

- Captures e.g. domain and/or software entities and concepts and their relationships
  - at various levels of abstraction
  - just classes or also attributes, methods and responsibilities
  - for understanding
  - as a starting point for implementation
- Mostly relevant for logical and development view
Examples II (look and usage)

- Deployment diagram (on the left) and component diagram (on the right)
- Used to exhibit the main components of a system, how they are organized in sub-systems and how they are distributed over (deployed on) various hardware devices of the system.
- Relevant to developers that are interested in the structure of the system, to system engineers and administrators that install the system and keep it running.
- Knowledge of interfaces is helpful for testing
Examples III (look and usage)

- Activity diagram (right)
- Sequence diagram (middle)
- State machine diagram (left)
- Used to describe (internal) behavior of the system, i.e. interactions between components possibly ordered in time, elaborating scenarios indicated in the use case view and possibly annotated with timing or other performance related requirements

- Primarily part of a process view and relevant for stakeholders that have to deal with systems testing and integration. Also end-users and developers that need to know detailed order of interactions with system components and between system components benefit from these diagrams
SysML

- UML primarily targets software (intensive) systems
- SysML is a variant of UML that is better suited for systems engineering:
  - Reuses 7 diagrams from UML
  - Adds two diagrams
    - requirements diagrams for requirements engineering
    - parameter diagrams for performance analysis, quantitative analysis in general
  - Facilitates automated verification and validation
  - Used in the automotive domain, embedded systems, ...

Both UML and SysML are considered Architectural Description Languages
Literature

- OMG UML (SysML) standard and related documents
  - http://www.uml.org/
  - http://www.omgsysml.org/

- Martin Fowler, *UML Distilled*, 3rd edition
  - https://martinfowler.com/books/uml.html


- Various sources on the WWW, e.g.
  - https://www.tutorialspoint.com/uml/
  - https://www.uml-diagrams.org/
  - http://sysmlforum.com/sysml-resources/tutorials/