

# UML 2.1 Fundamentals

abelski

# UML Introduction

# Object Oriented Analysis & Design

- Object Oriented Analysis and Design (OOAD) is a software engineering approach for modeling a system as a group of interacting objects.

Each object represents an entity, part of the system we model. Each object is characterized by a class (we declared and from which it was instantiated), its state (its data elements) and its behavior. The objects we create show the static structure, the dynamic behavior and the run time deployment of the system we model.

# Object Oriented Analysis & Design

- OOA focuses on what the system does and OOD focuses on how does the system do it.

The Object Oriented Analysis (OOA) applies various techniques to analyze the functional requirements for a system. The Object Oriented Design (OOD) elaborates the analysis models in order to produce the implementation specification.

# The Need for UML

- Prior to UML there were various types of graphical diagrams that were in use during the OOA and the OOD stages.
- The lack of standardization created the need to have one common graphics language to be used during the OOA and the OOD stages.

# What is UML?

- UML stands for Unified Modeling Language
- The UML is developed by OMG (Object Management Group).
- Since its introduction in 1997, UML has evolved and become the common standard for designing software systems.
- UML is a graphics language used for modeling software systems before coding them.

Modeling a software has the same role small physical models and architect's diagrams play in building a big house.

# What Can We Do With UML?

- Using UML models we can work at a higher level of abstraction.
- The UML abstraction level allows us to use it for modeling any type of application, hardware, operation system and/or programming language.
- Various tools allow generating code out of UML models and/or reverse engineering executable code into UML models. Other tools are capable of generating test suits and verification suits out of UML models.

# UML History

- **Before UML 1.X**

James Rumbaugh's “Object Oriented Technique (OOT)” and Grady Booch's “Booch Method”, the two most popular object oriented approaches together with Ivar Jacobson's “Object Oriented Software Engineering (OOSE)” method were collectively called “The Three Amigos”. The three amigos lead the “UML Partners” organization that was established in 1996 in order to complete the UML specification process.

- **UML 1.X Specifications**

In January 1997, UML 1.0 specification was proposed to OMG as a response for their RFP. Following OMG's remarks, the UML 1.0 specification were integrated with other standardization efforts and in August 1997 the UML 1.1 specification were submitted to the OMG. In November 1997 they were approved.

# UML History

- UML 2.X Specifications

During the years, small changes are introduced into the UML specification and the 1.2, 1.3, 1.4 & 1.5 specifications are released. In 2003 OMG adopts UML 2.0, that includes major changes and set a new ERA in UML.

# UML 1.0 & UML 2.0

- UML 2.0 adds new types of diagrams. In total, UML 2.0 has thirteen types of diagrams.

These diagrams are divided into three categories: structure diagrams, behavior diagrams & interaction diagrams.

- UML 2.0 allows nesting classifiers within each other. In other words, you can nest a diagram of one type within another and interact between the two.

One example is placing a class diagram within a component in order to present the classes that component manages.

# UML 1.0 & UML 2.0

- In UML 2.0 the behavioral models derive from a fundamental definition of behavior.

The behavioral diagrams include the Use Case Diagram, the State Diagram & the Activity Diagram. This improved modeling simplifies the UML diagrams.

- UML 2.0 enable us to use it for modeling purposes in different types of disciplines, including non computing ones.

UML 2.0 goes beyond UML 1.0 main purpose to model object and classes. UML 2.0 new diagrams enable us to use it in various different disciplines, such as business processes and rules.

# UML Diagrams

- UML 2.0 defines thirteen types of diagrams, divided into three categories:

## Structure Diagrams

Class Diagram, Object Diagram, Component Diagram, Composite Structure Diagram, Package Diagram, and Deployment Diagram.

## Behavior Diagrams

Use Case Diagram, Activity Diagram and State Machine Diagram.

## Interaction Diagrams

Sequence Diagram, Communication Diagram, Timing Diagram and Interaction Overview Diagram.

# Methodologies

- Before we start creating UML diagrams there is a need to choose the methodology we want to follow.
- A methodology defines the process we are going to use to gather the requirements, analyze them and design an application that meets them.
- Samples for methodologies:

Water Fall

Scrum

# UML Tools

- Different UML tools strength different methodologies.
- It is highly important to start with choosing the methodology and based on our choice to move forward and choose a UML tool that fits our needs.

# UML 2.1 Fundamentals

abelski

# UML Introduction

11/05/08

© 2008 Haim Michael. All Rights Reserved.

2

## Object Oriented Analysis & Design

- Object Oriented Analysis and Design (OOAD) is a software engineering approach for modeling a system as a group of interacting objects.

Each object represents an entity, part of the system we model. Each object is characterized by a class (we declared and from which it was instantiated), its state (its data elements) and its behavior. The objects we create show the static structure, the dynamic behavior and the run time deployment of the system we model.

## Object Oriented Analysis & Design

- OOA focuses on what the system does and OOD focuses on how does the system do it.

The Object Oriented Analysis (OOA) applies various techniques to analyze the functional requirements for a system. The Object Oriented Design (OOD) elaborates the analysis models in order to produce the implementation specification.

Object Oriented Analysis (OOA) looks at the problem trying to produce a conceptual model of the information without getting into any implementation constraints.

Object Oriented Design (OOD) transforms the conceptual model produced in the OOA stage into implementation classes and interfaces while taking into account the constraints the chosen architecture sets as well as any other non functional technological and environmental constraint.

## The Need for UML

- Prior to UML there were various types of graphical diagrams that were in use during the OOA and the OOD stages.
- The lack of standardization created the need to have one common graphics language to be used during the OOA and the OOD stages.

## What is UML?

- UML stands for Unified Modeling Language
- The UML is developed by OMG (Object Management Group).
- Since its introduction in 1997, UML has evolved and become the common standard for designing software systems.
- UML is a graphics language used for modeling software systems before coding them.

Modeling a software has the same role small physical models and architect's diagrams play in building a big house.

11/05/08

© 2008 Haim Michael. All Rights Reserved.

6

OMG is an international, open membership, not-for-profit computer industry consortium that was established in 1989. The OMG Task Forces develop enterprise integration standards for a wide range of technologies.

OMG's modeling standards, including the Unified Modeling Language™ (UML®) and Model Driven Architecture® (MDA®), enable powerful visual design, execution and maintenance of software and other processes (including IT Systems Modeling and Business Process Management).

For more info about OMG visit their official website at [www.omg.org](http://www.omg.org).

## What Can We Do With UML?

- Using UML models we can work at a higher level of abstraction.
- The UML abstraction level allows us to use it for modeling any type of application, hardware, operation system and/or programming language.
- Various tools allow generating code out of UML models and/or reverse engineering executable code into UML models. Other tools are capable of generating test suits and verification suits out of UML models.

11/05/08

© 2008 Haim Michael. All Rights Reserved.

7

UML diagrams help us by hiding or masking details, bringing out the big picture. Using UML we can focus on different aspects of the prototype.

UML Object Oriented fundamental concepts turns it into the natural choice for object oriented languages such as C++, Java & C#. UML Profiles (subset of UML tailored for specific purposes) allow us to model non Object Oriented applications as well.

Currently, OMG is working on a specification for execution UML.

## UML History

- **Before UML 1.X**

James Rumbaugh's "Object Oriented Technique (OOT)" and Grady Booch's "Booch Method", the two most popular object oriented approaches together with Ivar Jacobson's "Object Oriented Software Engineering (OOSE)" method were collectively called "The Three Amigos". The three amigos lead the "UML Partners" organization that was established in 1996 in order to complete the UML specification process.

- **UML 1.X Specifications**

In January 1997, UML 1.0 specification was proposed to OMG as a response for their RFP. Following OMG's remarks, the UML 1.0 specification were integrated with other standardization efforts and in August 1997 the UML 1.1 specification were submitted to the OMG. In November 1997 they were approved.

11/05/08

© 2008 Haim Michael. All Rights Reserved.

8

**The Object Management Group (OMG)** is a consortium founded in 1887 by eleven companies, including: IBM, HP, Sun Microsystems and, Apple. Its original targets were to set the standards for distributed object oriented systems. Today, OMG focuses on modeling and model based standards.

## UML History

- **UML 2.X Specifications**

During the years, small changes are introduced into the UML specification and the 1.2, 1.3, 1.4 & 1.5 specifications are released. In 2003 OMG adopts UML 2.0, that includes major changes and set a new ERA in UML.

## UML 1.0 & UML 2.0

- UML 2.0 adds new types of diagrams. In total, UML 2.0 has thirteen types of diagrams.

These diagrams are divided into three categories: structure diagrams, behavior diagrams & interaction diagrams.

- UML 2.0 allows nesting classifiers within each other. In other words, you can nest a diagram of one type within another and interact between the two.

One example is placing a class diagram within a component in order to present the classes that component manages.

11/05/08

© 2008 Haim Michael. All Rights Reserved.

10

UML 1.0 includes the following types of diagrams:

1. Use Case Diagram
2. Class Diagram
3. Behavior Diagram
  - 3.1 State Chart Diagram
  - 3.2 Activity Diagram
  - 3.3 Interaction Diagram
    - 3.3.1 Sequence Diagram
    - 3.3.2 Collaboration Diagram
4. Implementation Diagram
  - 4.1 Component Diagram
  - 4.2 Deployment Diagram

## UML 1.0 & UML 2.0

- In UML 2.0 the behavioral models derive from a fundamental definition of behavior.

The behavioral diagrams include the Use Case Diagram, the State Diagram & the Activity Diagram. This improved modeling simplifies the UML diagrams.

- UML 2.0 enable us to use it for modeling purposes in different types of disciplines, including non computing ones.

UML 2.0 goes beyond UML 1.0 main purpose to model object and classes. UML 2.0 new diagrams enable us to use it in various different disciplines, such as business processes and rules.

# UML Diagrams

- UML 2.0 defines thirteen types of diagrams, divided into three categories:

## Structure Diagrams

Class Diagram, Object Diagram, Component Diagram, Composite Structure Diagram, Package Diagram, and Deployment Diagram.

## Behavior Diagrams

Use Case Diagram, Activity Diagram and State Machine Diagram.

## Interaction Diagrams

Sequence Diagram, Communication Diagram, Timing Diagram and Interaction Overview Diagram.

## Methodologies

- Before we start creating UML diagrams there is a need to choose the methodology we want to follow.
- A methodology defines the process we are going to use to gather the requirements, analyze them and design an application that meets them.
- Samples for methodologies:
  - Water Fall
  - Scrum

11/05/08

© 2008 Haim Michael. All Rights Reserved.

13

Some methodologies might be better than others for a particular project for various reasons. Some methodologies are better suited for large enterprise applications while others were created to design small embedded systems. Some methods better large number of software designers while others work better when used by one person or a small group.

“An appropriate lifecycle methodology for OO developments must contain ALL of the following components: a full lifecycle process for both business and technological issues; a full set of concepts and models which are internally self-consistent; a collection of rules and guidelines; a full description of all deliverables; a workable notation; ideally supported by third party drawing tools; a set of tried and tested techniques; a set of appropriate metrics, standards and test strategies; identification of organizational roles e.g. business analyst, programmer; guidelines for project management and quality assurance; advice on library management and reuse.” (B. Henderson-Sellers)

## UML Tools

- Different UML tools strength different methodologies.
- It is highly important to start with choosing the methodology and based on our choice to move forward and choose a UML tool that fits our needs.