

面向对象程序的分析与设计 Object-Oriented Analysis and Design

Lecture 2

Prof. S. Xu

Contents

- ◆ UML
- ◆ UP

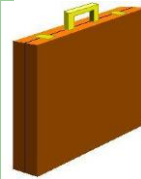
2

UML

What is modeling?

- A model is a simplification of reality.

What is this Thing?



Modeling a Briefcase

Abstracting
→

BriefCase
- Capacity
- Weight
+ open()
+ close()

4

Why Do We Model?

- We build models to better understand the system we are developing.
- Modeling achieves four aims:
 - Helps us to **visualize** a system as we want it to be.
 - Permits us to **specify** the structure or behavior of a system.
 - Gives us a template that guides us in constructing a system.
 - **Documents** the decisions we have made.
- We build models of complex systems because we cannot comprehend such a system in its entirety.

5

Historical View on Modeling Languages

- 1970's – Process-oriented methods (Structured System Analysis and Design – SSAD):
 - Use DFDs – **Data Flow Diagrams**.
- 1980's – Data-oriented methods:
 - Use ERDs – **Entity-Relationship Diagrams**.
- 1990's – Object-oriented methods (OMT, OOD, OOSE, UP):
 - Standard: UML – **Unified Modeling Language**.

6

Structured System Analysis and Design

Example: In the Library information system:

- System is broken by processes into:
 - Return/Loan,
 - AddResources,
 - reportFines,
 - ...

7

The Object-Oriented (OO) Approach

Example:

In the Library information system:

- System is broken by the object model into:
 - Classes: Book, Copy, LibraryManager, ...
 - Association: borrow/return, ...
 - Processes: Return/Loan, AddResources, reportFines, ...
- by assigning **responsibilities** to objects.

8

The Unified Modeling Language (UML)

started at 1995

9

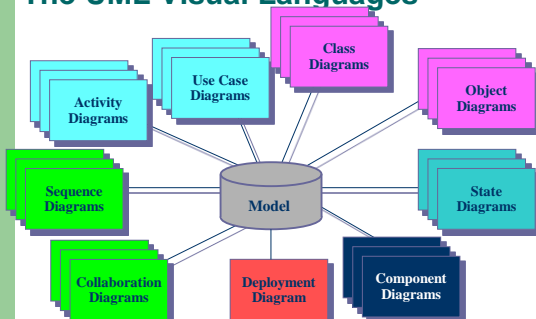
What Is the UML?

- The Unified Modeling Language (UML)
 - Specifying
 - Visualizing
 - Constructing
 - Documenting

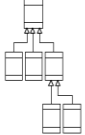
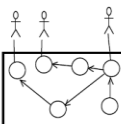
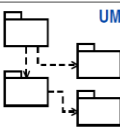
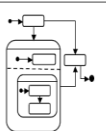
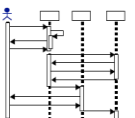
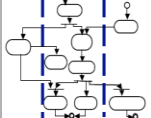


10

The UML Visual Languages



11

 UML Class Diagrams information structure relationships between data items modular structure for the system	 Use Cases user's view Lists functions visual overview of the main requirements
 UML Package Diagrams Overall architecture Dependencies between components	 (UML) Statecharts responses to events dynamic behavior event ordering, reachability, deadlock, etc
 UML Sequence Diagrams individual scenario interactions between users and system Sequence of messages	 Activity diagrams business processes; concurrency and synchronization; dependencies between tasks;

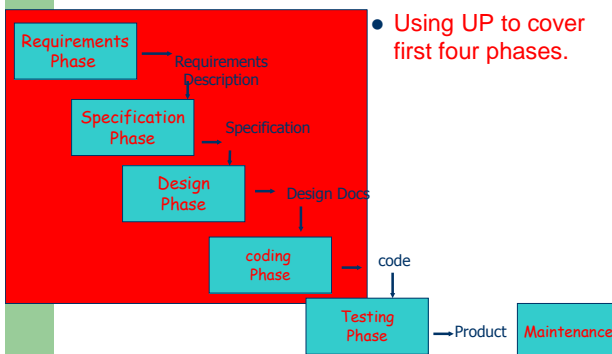
Future Technology

13

Unified Process (UP)

14

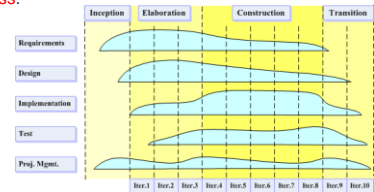
Waterfall Model



- Using UP to cover first four phases.

UP

- UP (Unified Process)
 - An agile (light, flexible) approach to the well-known Unified Process (UP) is used as the **sample iterative development process**.



- Involved with OOAD (OO Analysis and Design).

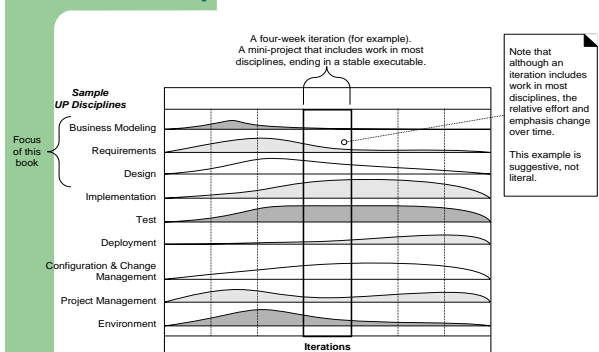
16

What's UP (Unified Process)?

- UP is an iterative development process.
- Each iteration involves choosing a small subset of the requirements, and quickly designing, implementing, and testing.
- The outcome of each is a **tested, integrated, and executable** partial system.

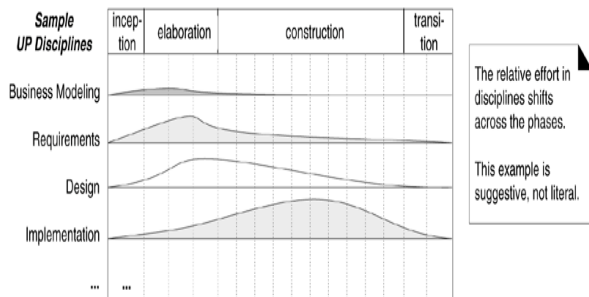
17

The UP Disciplines



18

UP Disciplines and phases



19

The UP phases

- A UP project organizes the work and iterations across four major phases:
 - Inception** -- approximate vision, business case, scope.
 - Elaboration** -- refined vision, **iterative implementation of the core architecture**, resolution of high risks, identification of most requirements and scope, more realistic estimates.
 - Construction** -- iterative **implementation** of the remaining lower risk and easier elements, and preparation for deployment.
 - Transition** -- beta tests, deployment.

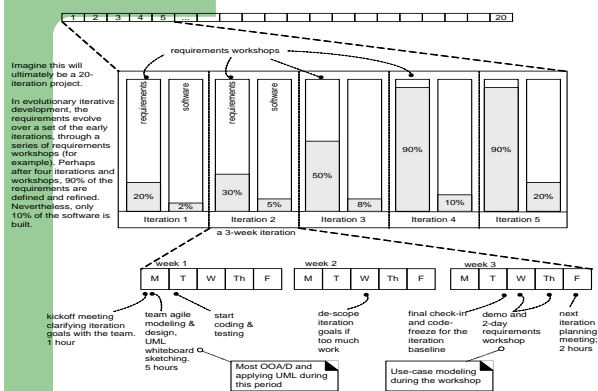
20

Customize UP, The UP Development Case

Table 2.1. Sample Development Case. s - start; r - refine

Discipline	Practice	Artifact	Incep.	Elab.	Const.	Trans.
		Iteration →	I1	E1..En	C1..Cn	T1..T2
Business Modeling	agile modeling	Domain Model		s		
Requirements	req. workshop vision box exercise dot voting	Use-Case Model	s	r		
		Vision	s	r		
		Supplementary Specification	s	r		
Design	agile modeling test-driven dev.	Glossary	s	r		
		Design Model		s	r	
		SW Architecture Document		s		
Implementation	test-driven dev. pair programming continuous integration coding standards	Data Model		s	r	
		...				
Project Management	agile PM daily Scrum meeting	...				
...						

How to do Iterative and Evolutionary Analysis and Design?



22

Inception is Not the Requirements Phase

- Inception is the initial **short step** to establish a common vision and basic scope for the project.
- It will include analysis of perhaps 10% of
 - the use cases (? , chapter 6)
 - analysis of the critical non-functional requirement,
 - creation of a business case, and
 - preparation of the development environment

so that programming can start in the following elaboration phase.

23

Inception is Not the Requirements Phase

- KEEP IN MIND:
 - the purpose of the inception phase is not to define all the requirements, or generate a believable estimate or project plan.
 - Inception, is not the time do all requirements or create believable estimates or plans. That happens during elaboration.

24

Artifacts in Inception

Artifact [†]	Comment
Vision and Business Case	Describes the high-level goals and constraints, the business case, and provides an executive summary.
Use-Case Model	Describes the functional requirements. During inception, the names of most use cases will be identified, and perhaps 10% of the use cases will be analyzed in detail.
Supplementary Specification	Describes other requirements, mostly non-functional. During inception, it is useful to have some idea of the key non-functional requirements that have will have a major impact on the architecture.
Glossary	Key domain terminology, and data dictionary.
Risk List & Risk Management Plan	Describes the risks (business, technical, resource, schedule) and ideas for their mitigation or response.
Prototypes and proof-of-concepts	To clarify the vision, and validate technical ideas.
Iteration Plan	Describes what to do in the first elaboration iteration.
Phase Plan & Software Development Plan	Low-precision guess for elaboration phase duration and effort. Tools, people, education, and other resources.
Development Case	A description of the customized UP steps and artifacts for this project. In the UP, one always customizes it for the project.

[†] These artifacts are only partially completed in this phase. They will be iteratively refined in subsequent iterations. Name capitalization implies an officially named UP artifact.

Is UML involved in inception?

- There is more focus in inception on understanding the basic scope and 10% of the requirements, expressed mostly in text forms.

26

Types and Categories of Requirements in UP

- In the UP, requirements are categorized according to the FURPS+ model
- **Functional** - features, capabilities, security.
- **Usability** - human factors, help, documentation.
- **Reliability** - frequency of failure, recoverability, predictability.
- **Performance** - response times, throughput, accuracy, availability, resource usage.
- **Supportability** - adaptability, maintainability, internationalization, configurability.

27

Types and Categories of Requirements in UP

- The “+” sign means
 - **Implementation** - resource limitations, languages and tools, hardware, ...
 - **Interface** - constraints imposed by interfacing with external systems.

28

Types and Categories of Requirements in UP

- A more general categorization of requirements:
 - Functional requirements
 - Non-functional requirements

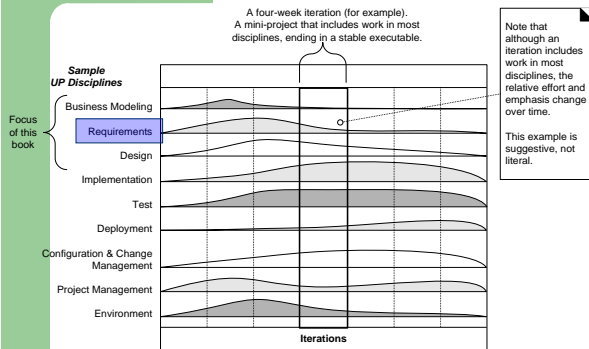
29

Types and Categories of Requirements in UP

- Functional or non-functional
 - An account debit transaction must ensure adequate funds exist prior to debiting.
 - An account debit transaction must be easy to perform for the user.

30

Requirement Artifacts in UP



31

Requirement Artifacts in UP

- The UP offers several requirements artifacts. As with all UP artifacts, they are optional. Key ones include:

- **Use-Case Model**: set of typical scenarios of using a system. CAPTURE functional (behavioral) requirements.
- **Supplementary Specification**: Basically, everything not in the use cases. CAPTURE all **non-functional requirements**, such as performance or licensing.

32

Requirement Artifacts in UP

- The UP offers several requirements artifacts. As with all UP artifacts, they are optional. Key ones include:
 - **Glossary**: Glossary defines noteworthy terms. The data dictionary
 - **Vision**: Summarizes high-level requirements that are elaborated in the Use-Case Model and Supplementary Specification, and summarizes the business case for the project.

33

Requirement Artifacts in UP

- **Business Rules**: Business rules (also called Domain Rules) typically describe requirements or policies that transcend one software project - they are required in the domain or business, and many applications may need to conform to them. An excellent example is government tax laws.

34

Review

- UML is a language for specifying, visualizing and documenting the software systems. [True/False]
- Benefit of iterative model is that high risk issues are addressed earlier. [True/False]
- Both use case model and design model will start in elaboration phase. [True/False]
- Ease of use is a non-functional requirement [True/False]
- Use case modeling in the inception phase will usually describe around 80 percent of the user requirements. [True/False]
- We continue writing use cases in the construction phase [True/False]

35

Review

- Why is Software Design important?
- Fill in the blanks:
"Do the right thing" summarizes _____
"Do the thing right" summarizes _____

36

Review

- In Object-oriented Technology the word "UML" means
 - A: Unified Module Language
 - B: Unified Modeling Language
 - C: Universal Module Leveling
 - D: Universal Module Language
- Which of the following statements is true?
 - A: Structured analysis was developed from Object-oriented Technology.
 - B: UML is a collaboration of the main stream Object-Oriented Methodologies.
 - C: Object-Oriented Technology is the culmination of High level Programming languages such as Java and C++.
 - D: Only object-oriented programming languages can be used to implement object-oriented designs.

37

Review

- What is true about Object-oriented Analysis?
 - A: It is a method of modeling requirements of the system
 - B: It is a technique for analyzing the design of the system
 - C: Code can be written directly from the Analysis
 - D: It comes after object-oriented design in the product life cycle

38

Introduction to Higher Education

Country	USA	CANADA	BELGIUM
Control	State control	Provincial control	Governmental control (two ministries)
Number of institutions	>3000 Private and public	150 All public	40 All public
Undergraduate Entry requirements	High school record SAT	High school record	High school record
Graduate entry requirements	GRE GMAT	NON GMAT	Non
Entry requirements for foreigners	TOEFL	TOFEL	Institutional Level

39

Structure of Higher Education

USA	CANADA	BELGIUM
Diploma, associate degree, Bachelor,	Diploma, Bachelor,	First stage Candidature
Master, Post-Master	Master	Second stage Licence, Maitrise
Doctorate	Doctorate	Third stage DES, DEA, Doctorat
Post-doctorate	Post-doctorate	Fourth stage Agregation de l'Enseignement superieur

40

Length of graduate study (Years)

Country	USA	CANADA	BELGIUM
Master's degree	Master of Arts: 1 Master of Science: 2	M.Sc. 1.5-2	1 year
Ph.D. degree	4-8 years (10 courses, 2-3 exams, papers?)	3-4 years (3 courses, 1 exam, 2-3 papers)	Research oriented 4-5 years

41

Financial support for graduate study

Country	USA	CANADA	BELGIUM
Tuition	Tuition is covered by the TA, RA	Tuition should be paid by students	Almost no tuition for Ph.D. study
Amount	TA (\$15,000 per two semester) RA	TA (\$4000 per semester) RA	RA

42

World's top 101 research universities: geo-distribution

