



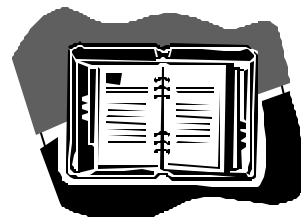
Taking Care of e-Business

The role of data modeling in component-based development

David L. Shapiro
DLS Associates
DavidS@dlsa.com

Agenda

- Setting the Stage
- Component-Based Development
- The Role of Data Modeling



Setting the Stage: Recent Trends

- Business process re-engineering
- Development trends
- The dot-com "revolution"
- What is e-Business?

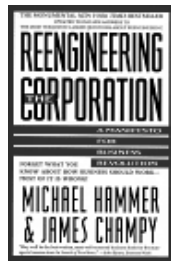
Copyright © 2002 DLS Associates (www.dlsa.com)

3

Business Process Re-engineering

- *"the critical analysis and radical redesign of existing business processes to achieve breakthrough improvements in performance measures."*

Malhotra, Yogesh. "Business Process Redesign: An Overview," IEEE Engineering Management Review



"Don't automate, obliterate!"

Hammer, M. (1990, July-August). "Reengineering Work: Don't Automate, Obliterate," *Harvard Business Review*

Copyright ©1994

Copyright © 2002 DLS Associates (www.dlsa.com)

4

Development Trends

- eXtreme Programming (XP)
 - Requirements and design obsolescence
 - Pair programming
 - Continuous integration and testing
- Object-Oriented Design (OOD)
 - Component-based development
 - Visual modeling (UML)
 - Java development

Copyright © 2002 DLS Associates (www.dlsa.com)

5

The dot-COM Revolution

- ebay, Amazon and Yahoo!
- Increasing awareness
 - e.g., SuperBowl 2000
- The soaring NASDAQ
- Increasing bandwidth



Copyright © 2002 DLS Associates (www.dlsa.com)

6

The Impact on IT

- Dot-COM's everywhere
 - Job/Salary explosion
 - Employee shortage
- e-Business at the Office
 - Blank check projects
 - "Death March" web projects
 - Unprecedented training and recruiting

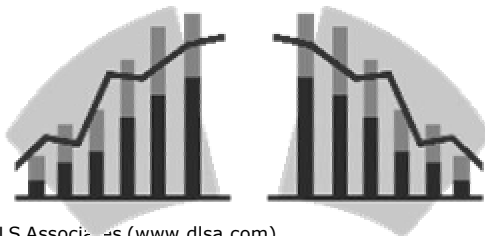


Copyright © 2002 DLS Associates (www.dlsa.com)

7

What goes up...

- Overnight from feast to famine
 - Employee surpluses
 - Hiring freezes
 - Shelved projects

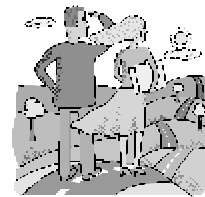


Copyright © 2002 DLS Associates (www.dlsa.com)

8

Where Do We Go From Here?

- Can't turn back now
- More accountable (no blank checks)
- More careful: proven technologies
- Standards and process



Copyright © 2002 DLS Associates (www.dlsa.com)

9

What is e-Business?

Either:

- (1) **the transaction of business** over an electronic medium such as the Internet or
- (2) **a business** that uses Internet technologies and network computing in their internal business processes (via intranets), their business relationships (via extranets), and the buying and selling of goods, services, and information (via electronic commerce).

(Source: Rational Unified Process Glossary)

Copyright © 2002 DLS Associates (www.dlsa.com)

10

What Is Success?

- On Time
- On Budget
- Meets User Expectations
- Good Quality



Copyright © 2002 DLS Associates (www.dlsa.com)

11

How Successful Are We?

- 31.1% of projects canceled before they get completed
- 52.7% of projects cost 189% of their original estimates
- 16.2% for software projects completed on-time and on-budget

Source: The Standish Group Chaos Report
(results based on survey of 8,380 applications)
<http://standishgroup.com/visitor/chaos.htm>



Copyright © 2002 DLS Associates (www.dlsa.com)

12

How Can We Do Better?

- Capitalize on successes
- Learn from mistakes
- Reuse

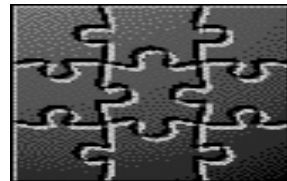


Copyright © 2002 DLS Associates (www.dlsa.com)

13

Technology Standards

- Component Standards
 - CORBA
 - COM/DCOM
 - J2EE
 - .NET

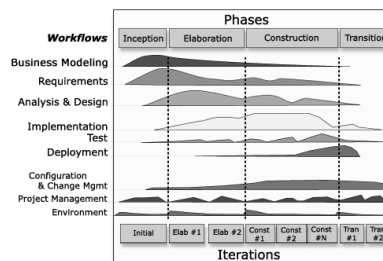


Copyright © 2002 DLS Associates (www.dlsa.com)

14

Design and Process Standards

- Design and Process Standards
 - Unified Modeling Language (UML)
 - Rational Unified Process (RUP)



Copyright © 2002 DLS Associates (www.dlsa.com)

15

Rational Unified Process

- Iterative Development
- Architecture Focused
- Use Case Driven

Copyright © 2002 DLS Associates (www.dlsa.com)

16

Iterative Development

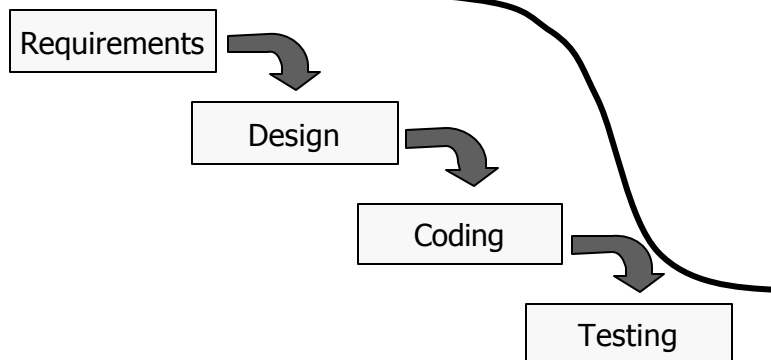


Copyright © 2002 DLS Associates (www.dlsa.com)

17

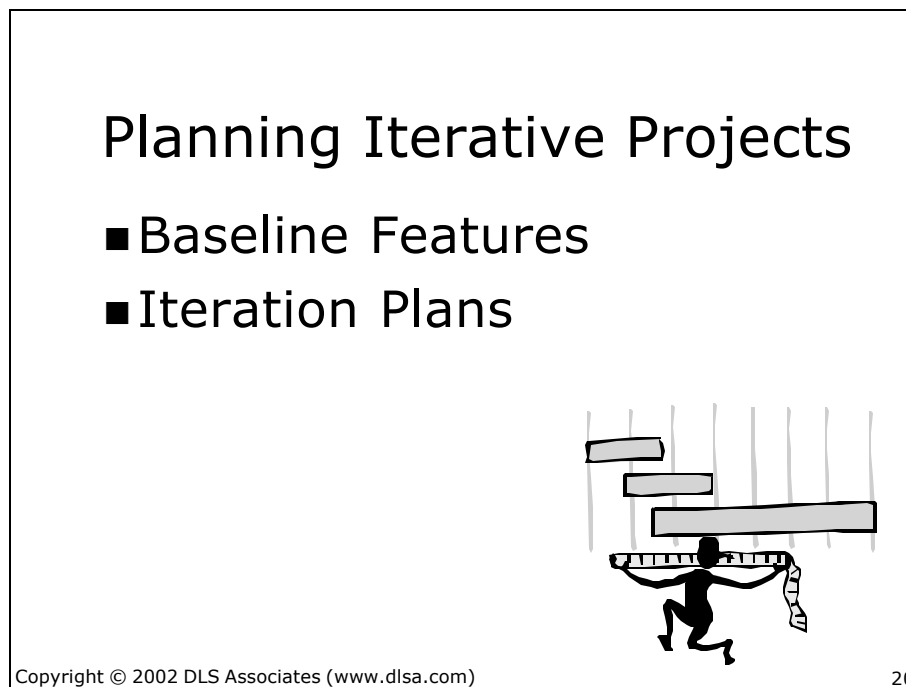
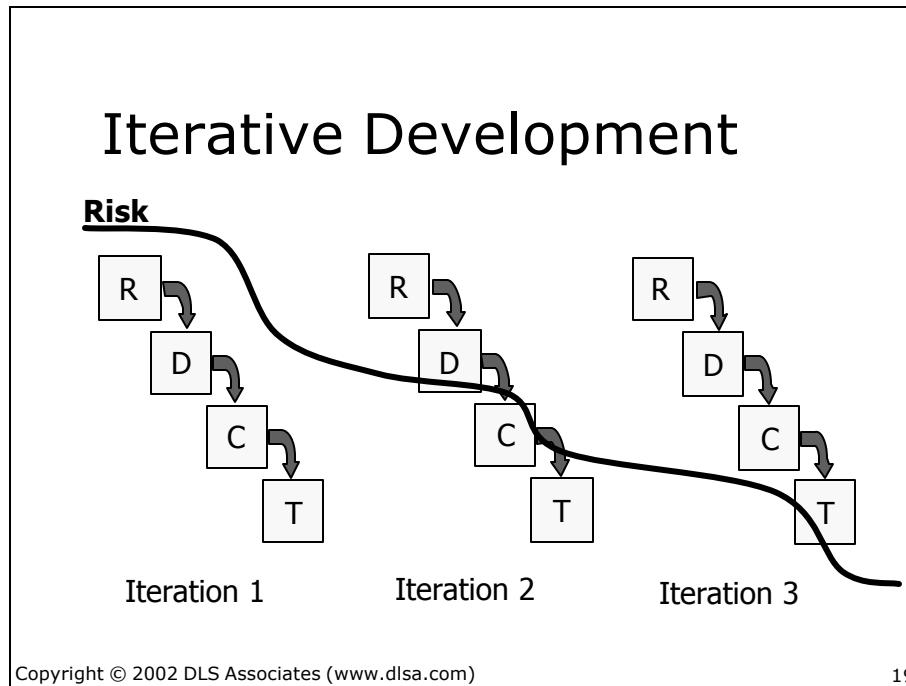
Waterfall Development

Risk



Copyright © 2002 DLS Associates (www.dlsa.com)

18



Baseline Features

- What will be delivered in X months?
- Driven primarily by:
 - Customer priority



Copyright © 2002 DLS Associates (www.dlsa.com)

21

Iteration Planning

- Driven primarily by risk
- Something that might happen, and if it does, will have a negative impact
- Sample risks
 - Technology risks
 - Requirements risks
 - Market risks

Copyright © 2002 DLS Associates (www.dlsa.com)

22

Development Phases

- Inception: project scope
- Elaboration: eliminate risk
- Construction: build the product
- Transition: deploy the product

Copyright © 2002 DLS Associates (www.dlsa.com)

23

Inception Phase

- Define project scope
 - Baseline
 - Initial estimate
 - Initial risk list
 - Project plan
- Milestone: go/no go decision

Copyright © 2002 DLS Associates (www.dlsa.com)

24

Elaboration Phase

- Eliminate risk
- Milestone: approx. 20% functionality
- Demonstrate working architecture
 - Core functionality
 - Shared services
 - Technical exposure

Copyright © 2002 DLS Associates (www.dlsa.com)

25

Elaboration Milestone

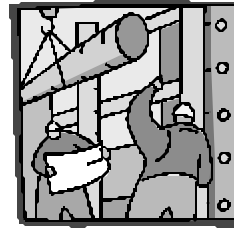
- Approximately 25% of budget spent project-to-date
- Approximately 40% of calendar time elapsed
- Go/no go: invest remaining 75% with high confidence

Copyright © 2002 DLS Associates (www.dlsa.com)

26

Construction Phase

- Build the Product: additional 80%
- Leverage Resources
(The Deadline by Tom DeMarco)
- Milestone: Release Candidate
(beta version)



Copyright © 2002 DLS Associates (www.dlsa.com)

27

Transition Phase

- Deployment
 - Installation
 - Documentation
 - Training
- Milestone: Product Release (GA)

Copyright © 2002 DLS Associates (www.dlsa.com)

28

Clash with Past

- Managers/Customers:
 - Expect “frozen” artifacts
 - Not accustomed to incremental review
 - Shocked by rework
 - View iterations as “fumbling around”

Copyright © 2002 DLS Associates (www.dlsa.com)

29

Example: Rework

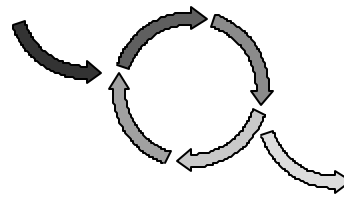
- Expected Rework:
 - Inception, 40%-100% - throwaway, exploratory prototypes
 - Elaboration, 25%-60% in early iterations; less than 25% in later iterations
 - Construction, 10% or less per iteration and 25% total.
 - Transition, less than 5%.

Copyright © 2002 DLS Associates (www.dlsa.com)

30

Iterative Development

- Reduce risk early
- Better control for project management
- Close match to requirements



Copyright © 2002 DLS Associates (www.dlsa.com)

31



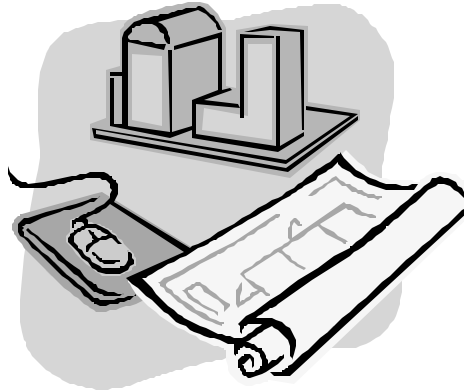
Exercise: Iteration Planning

Use Cases	Priority	Difficulty	Risk	Stability
Use Case 1: Save and restore sort and filter criteria.	Med High	Low	Low	High
Use Case 2: Save a report as a Word document.	Med High	Low	Low	High
Use Case 3: View and undo deleted requirements in a view window.	Medium	Med High	Medium	Medium
Use Case 4: Generate "complex" document from "include," "extend," and "generalized" use cases.	Medium	Medium	Med Low	Medium
Use Case 5: Automatically generate HTML from standard reports.	Med High	Medium	Medium	Med High
Use Case 6: Select requirement in a view and jump to requirement in Word document.	Med High	Medium	Medium	Med High
Use Case 7: Display a requirement's attribute in the text of the requirement's document.	Medium	Medium	Medium	Med High
Use Case 8: New project wizard	Med High	High	Med High	Medium
Use Case 9: Fast creation of a requirement (avoid the requirement dialog on creation).	Med High	Med Low	Med Low	High
Use Case 10: Autosave project (project archive).	Medium	Med Low	Medium	Medium
Use Case 11: Change one or more attributes for a selected set of requirements.	Medium	Med High	Medium	Medium
Use Case 12: Clone a project's structure to allow users to easily create new projects from old.	High	Medium	Medium	Low
Use Case 13: Print cross-reference and treeview with requirement identification.	High	Med High	Medium	Med High
Use Case 14: Windows2000 Port.	High	Medium	High	High

Copyright © 2002 DLS Associates (www.dlsa.com)

32

Software Architecture



Copyright © 2002 DLS Associates (www.dlsa.com)

33

Traditional Architect

- Scale model
- Design standards
- Infrastructure services
- Tools and strategies



Copyright © 2002 DLS Associates (www.dlsa.com)

34

Software Architecture

- Modeling (scale model)
- Guidelines (standards)
- Mechanisms (services)
- Tools and strategies

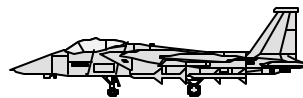
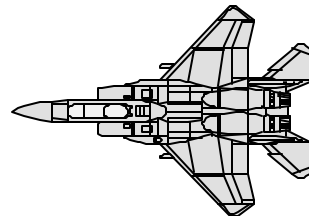


Copyright © 2002 DLS Associates (www.dlsa.com)

35

What is a model?

- Abstraction of reality
 - People
 - Processes
 - Things
- Examples:
 - Entity-relationship model
 - Data model
 - Object model

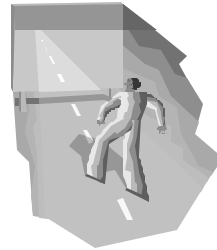


Copyright © 2002 DLS Associates (www.dlsa.com)

36

What Is Abstraction?

- Representation of real thing
- Captures what is important for the problem being solved



Copyright © 2002 DLS Associates (www.dlsa.com)

37

Example: Abstraction



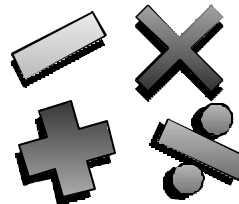
Student



Professor



Course Offering (9:00 AM,
Monday-Wednesday-Friday)



Course (e.g. Algebra)

Copyright © 2002 DLS Associates (www.dlsa.com)

38

Object Modeling (UML)

- Main models
 - Use Case Model
 - Logical View
(class and interaction diagrams)
 - Implementation View
 - Deployment View
 - Test Model
 - Data Model

Copyright © 2002 DLS Associates (www.dlsa.com)

39

Guidelines

- Naming conventions
- Programming standards
- Interface standards
- Documentation standards and styleguides
- User Interface standards

Copyright © 2002 DLS Associates (www.dlsa.com)

40

Mechanisms

- Security
- Persistence
- IPC
- Transactions
- Error Handling/Reporting

Copyright © 2002 DLS Associates (www.dlsa.com)

41

Tools and Strategies

- Technologies
 - EJB, CORBA, COM
- Tools
 - Modeling
 - Development Environment
- Database
- Configuration Management

Copyright © 2002 DLS Associates (www.dlsa.com)

42

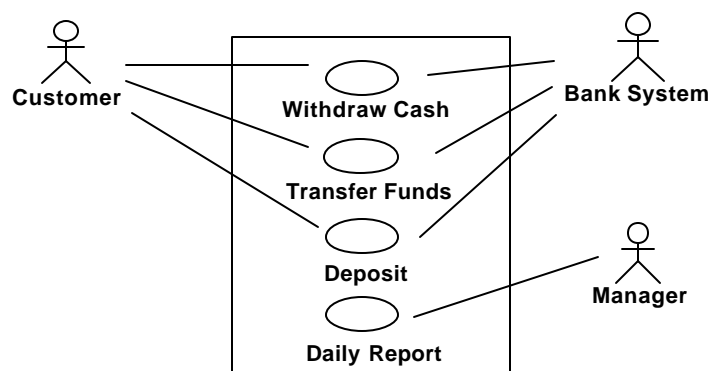
Advantages of Software Architecture

- Visual Modeling
- Design for Re-Use
- Integration Consistency
- Leverage Resources

Copyright © 2002 DLS Associates (www.dlsa.com)

43

Use Case Driven Development



Copyright © 2002 DLS Associates (www.dlsa.com)

44

Why Use Cases?

- Focus on users' perspective
- Communicate to everyone
 - Users
 - Designers
- Translate well to object model

Copyright © 2002 DLS Associates (www.dlsa.com)

45

Building Use Cases

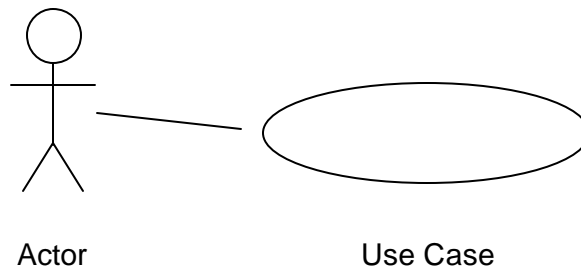
- How do we get started building use cases?



Copyright © 2002 DLS Associates (www.dlsa.com)

46

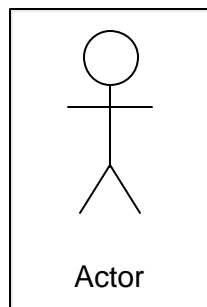
Use Case Model Elements



Copyright © 2002 DLS Associates (www.dlsa.com)

47

Actor



An actor represents an external entity that interacts with the system.

Copyright © 2002 DLS Associates (www.dlsa.com)

48

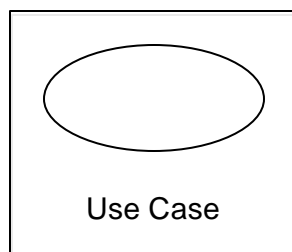
Actor Examples

- Users (e.g., customer, employee, etc.)
- External Systems (legacy systems, other recent systems)
- Devices (especially in embedded systems)
- Special actor: system clock

Copyright © 2002 DLS Associates (www.dlsa.com)

49

Use Case Definition

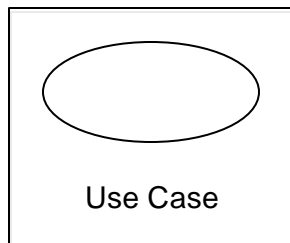


A **use case** defines a **sequence of actions** performed by a system that yields an **observable result of value** to an actor.

Copyright © 2002 DLS Associates (www.dlsa.com)

50

Use Case Key Concepts



Sequence of actions

Initiated by an actor

Result of Value

Focus on customer

Observable

Verifiable

Testable

Copyright © 2002 DLS Associates (www.dlsa.com)

51

Use Cases in Inception Phase

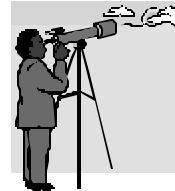
- Use Case Model Survey
- Initial Estimates
- Ranking (priority, risk, stability, cost)
- Gain Agreement on Scope

Copyright © 2002 DLS Associates (www.dlsa.com)

52

Inception: Use Case Model Survey

- Summarizes Use Case Model
- Includes:
 - Use Case Model Diagram
 - Actor Descriptions
 - Use Case Descriptions



Copyright © 2002 DLS Associates (www.dlsa.com)

53

Inception: Use Case Outline

Withdraw Cash

Brief Description

The bank customer uses the ATM machine to get cash in convenient locations, without the need for a bank teller. The customer may request a specific amount of cash, or may request "fast cash" which tells the ATM to dispense a pre-set amount, and immediately return cash, card, and receipt to the customer.

Outline for Main ("Happy Day") Flow of Events

1. Customer inserts card
2. System prompts for PIN
3. Customer enters PIN
4. System prompts for Withdraw, Transfer, or Deposit
5. Customer selects Withdraw

...

Copyright © 2002 DLS Associates (www.dlsa.com)

54

Ranking Use Cases

Use Cases	Priority	Difficulty	Risk	Stability	Baseline	Order
Use Case 1: Save and restore sort and filter criteria.	Med High	Low	Low	High		
Use Case 2: Save a report as a Word document.	Med High	Low	Low	High		
Use Case 3: View and undo deleted requirements in a view window.	Medium	Med High	Medium	Medium		
Use Case 4: Generate "complex" document from "include," "extend," and "generalized" use cases.	Medium	Medium	Med Low	Medium		
Use Case 5: Automatically generate HTML from standard reports.	Med High	Medium	Medium	Med High		
Use Case 6: Select requirement in a view and jump to requirement in Word document.	Med High	Medium	Medium	Med High		
Use Case 7: Display a requirement's attribute in the text of the requirement's document.	Medium	Medium	Medium	Med High		
Use Case 8: New project wizard	Med High	High	Med High	Medium		
Use Case 9: Fast creation of a requirement (avoid the requirement dialog on creation).	Med High	Med Low	Med Low	High		
Use Case 10: Autosave project (project archive).	Medium	Med Low	Medium	Medium		
Use Case 11: Change one or more attributes for a selected set of requirements.	Medium	Med High	Medium	Medium		
Use Case 12: Clone a project's structure to allow users to easily create new projects from old.	High	Medium	Medium	Low		
Use Case 13: Print cross-reference and treeview with requirement identification.	High	Med High	Medium	Med High		
Use Case 14: Windows2000 Port.	High	Medium	High	High		

Copyright © 2002 DLS Associates (www.dlsa.com)

55

Gain Agreement

- Walkthrough with users
 - Confirm that system does what they need
 - Storyboard to help refine
 - Revisit if necessary when iteration is available
- Walkthrough with designers
 - Confirm that use cases provide enough detail to build system

Copyright © 2002 DLS Associates (www.dlsa.com)

56

Use Cases in Elaboration Phase

- Detail Use Cases
- Develop Test Cases
- Build Iteration
- Demonstrate and re-work

Copyright © 2002 DLS Associates (www.dlsa.com)

57

Elaboration: Use Case Details

Withdraw Cash

1. Brief Description

The bank customer uses the ATM machine ...

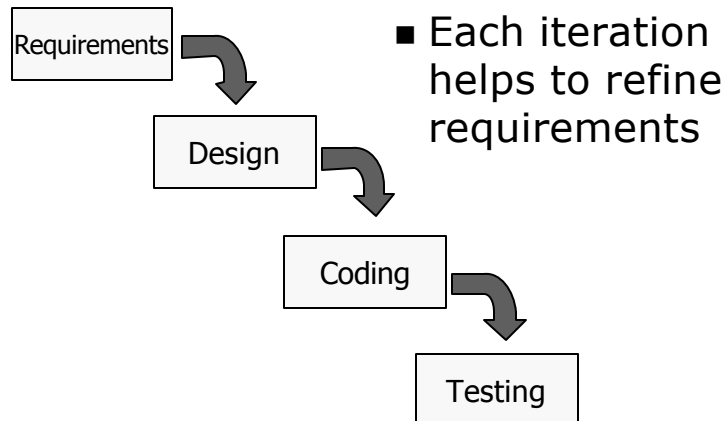
2. Main ("Happy Day") Flow of Events

- 2.1 The use case begins when the customer inserts their ATM card into the ATM machine.
- 2.2 The system verifies that the card is valid and prompts for a PIN number.
- 2.3 The customer enters their PIN number and the system verifies that the PIN is correct.
- 2.4 The system asks the customer if they want to: a) withdraw cash, b) transfer funds, or c) make a deposit.
- 2.5 The customer chooses "withdraw cash."
- 2.6 The system asks which account type (checking or savings) from which to withdraw.
- 2.7 The customer chooses "checking."
- 2.8 The system asks how much the customer wishes to withdraw.

Copyright © 2002 DLS Associates (www.dlsa.com)

58

Mini-Waterfall

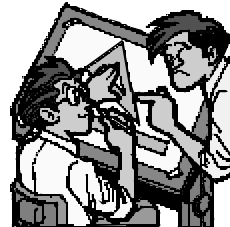


Copyright © 2002 DLS Associates (www.dlsa.com)

59

Translates to Object Model

- Analysis
 - Restate requirements as object model
 - Translate from use case view (English) to logical view (Object Model)



Copyright © 2002 DLS Associates (www.dlsa.com)

60

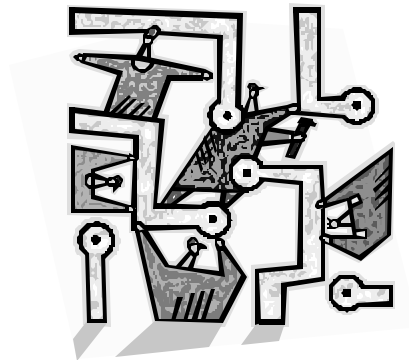
Use Case Advantages

- Focus on results of value
- Focus on communication
 - Understandable to customer
 - Understandable to designer
- Encourages continuing customer contact
- Translates well to Analysis and Design Model

Copyright © 2002 DLS Associates (www.dlsa.com)

61

Analysis and Design

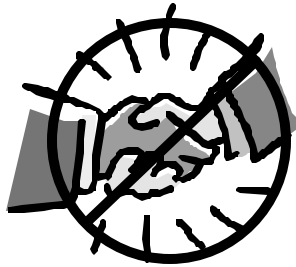


Copyright © 2002 DLS Associates (www.dlsa.com)

62

Why Do Projects Fail?

- “They didn’t build what we told them.”
- “They didn’t tell us what we needed.”

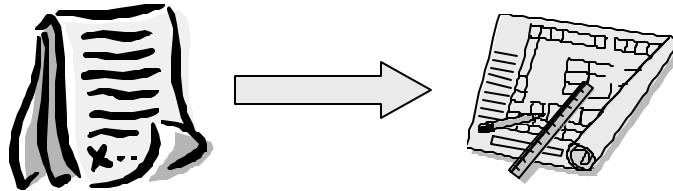


Copyright © 2002 DLS Associates (www.dlsa.com)

63

Analysis

- Translating from requirements model to logical model



Restate the requirements in the form of an object model

Copyright © 2002 DLS Associates (www.dlsa.com)

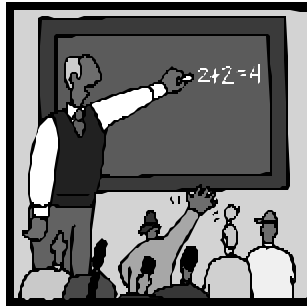
64

Abstraction: A Class

Class CourseOffering

Properties

Name
Location
Days offered
Credit hours
Start time
End time



Behavior

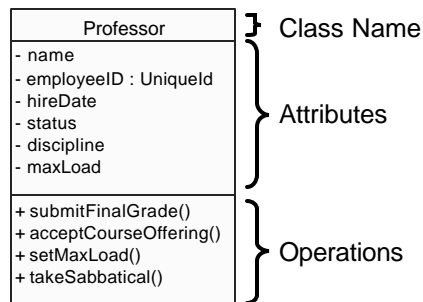
Add a student
Delete a student
Get course roster
Determine if it is full

Copyright © 2002 DLS Associates (www.dlsa.com)

65

Representing Classes in the UML

- A class is represented using a rectangle with compartments.



Professor J Clark

Copyright © 2002 DLS Associates (www.dlsa.com)

66

Object

- An object is an abstraction of a real-world thing, which is represented as a rectangle with an underlined name.



Professor J Clark

: Professor

Unnamed Object

J Clark

J Clark :
Professor

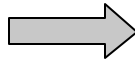
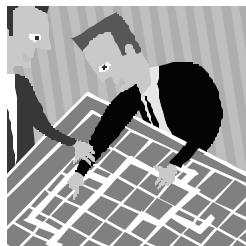
Named Object

Copyright © 2002 DLS Associates (www.dlsa.com)

67

Classes and Objects

- A class is a blueprint
- An object occupies space

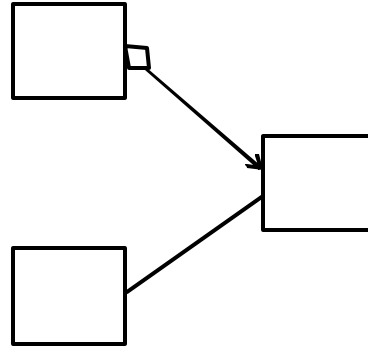


Copyright © 2002 DLS Associates (www.dlsa.com)

68

Class Diagrams

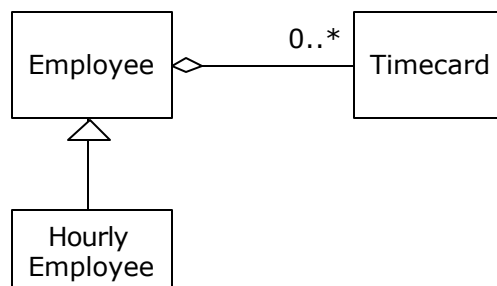
- Business abstractions represented in software
- Relationships among business abstractions



Copyright © 2002 DLS Associates (www.dlsa.com)

69

Class Diagram: Example

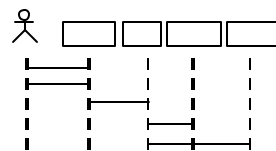


Copyright © 2002 DLS Associates (www.dlsa.com)

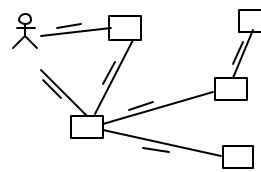
70

Modeling Behavior

- An interaction diagram shows an interaction, consisting of a set of objects and their relationships, including the messages that may be dispatched among them.
- It models the dynamic aspects of a system.



Sequence Diagrams

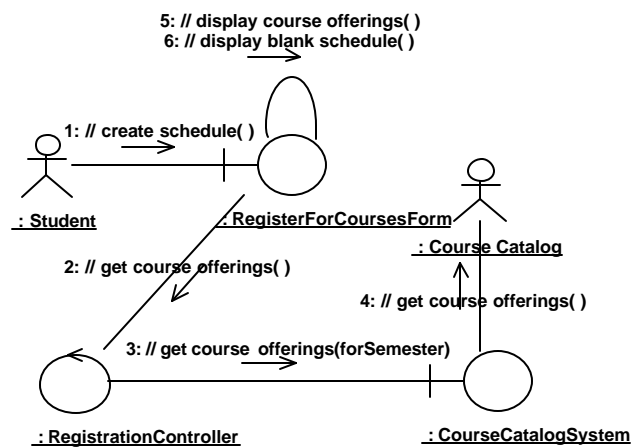


Collaboration Diagrams

Copyright © 2002 DLS Associates (www.dlsa.com)

71

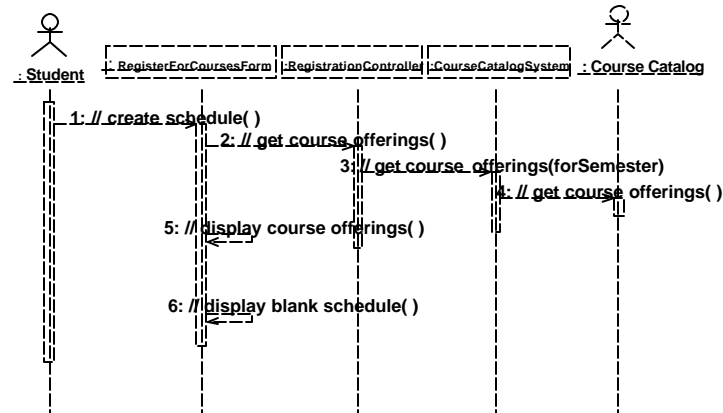
Example: Collaboration Diagram



Copyright © 2002 DLS Associates (www.dlsa.com)

72

Example: Sequence Diagram



Copyright © 2002 DLS Associates (www.dlsa.com)

73

Sequence and Collaboration Diagram Differences

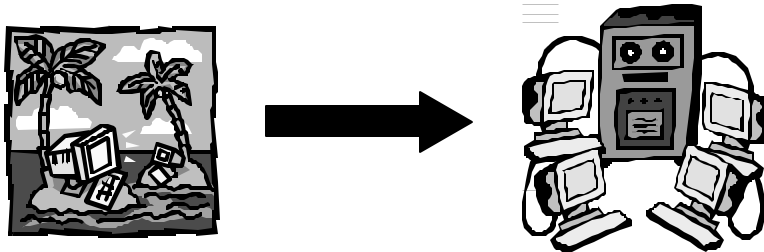
- Collaboration diagrams
 - Show relationships in addition to interactions
 - Better for visualizing patterns of collaboration
 - Better for visualizing all of the effects on a given object
 - Easier to use for brainstorming sessions
- Sequence diagrams
 - Show the explicit sequence of messages
 - Better for visualizing overall flow
 - Better for real-time specifications and for complex scenarios

Copyright © 2002 DLS Associates (www.dlsa.com)

74

Design

- Making the analysis model work in the “real world”



Copyright © 2002 DLS Associates (www.dlsa.com)

75

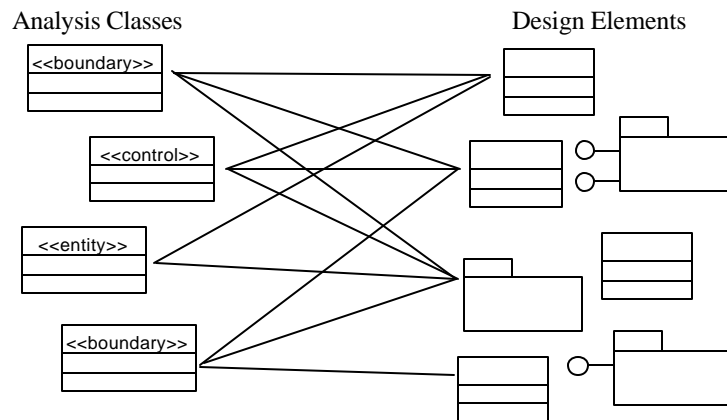
Analysis Versus Design

- | ■ Analysis | ■ Design |
|--------------------------------------|---------------------------------------|
| ■ Focus on understanding the problem | ■ Focus on understanding the solution |
| ■ Idealized design | ■ Operations and Attributes |
| ■ Behavior | ■ Performance |
| ■ System structure | ■ Close to real code |
| ■ Functional requirements | ■ Object lifecycles |
| ■ A small model | ■ Non-functional requirements |
| | ■ A large model |

Copyright © 2002 DLS Associates (www.dlsa.com)

76

Analysis to Design

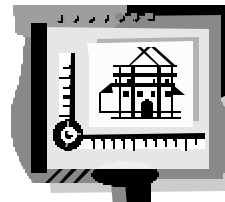


Copyright © 2002 DLS Associates (www.dlsa.com)

77

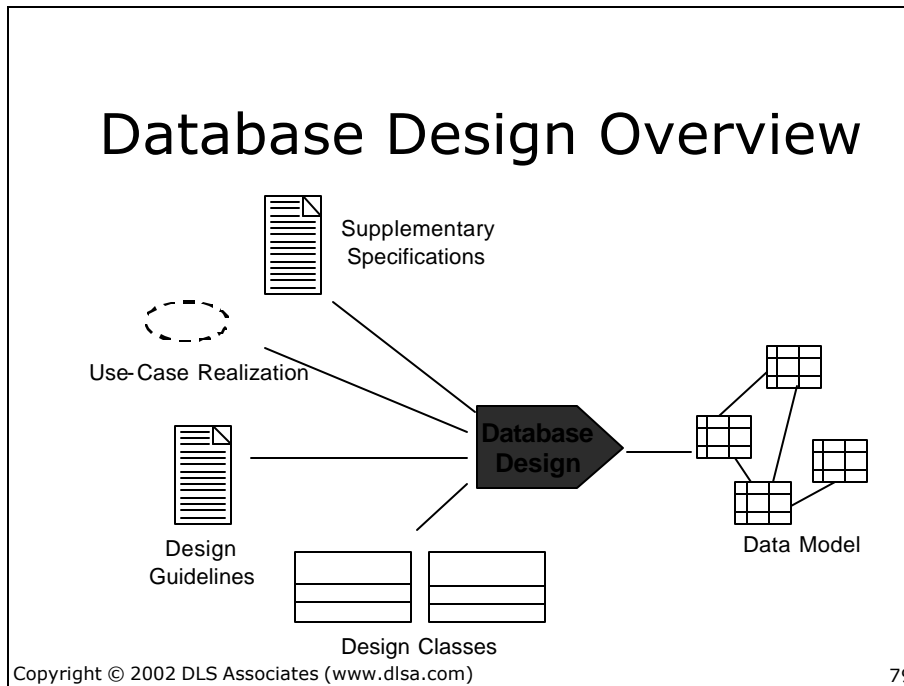
Design Activities

- Use Case Design
- Subsystem Design
- Class Design
- Database Design



Copyright © 2002 DLS Associates (www.dlsa.com)

78



RDBMS and Object Orientation Are Not Entirely Compatible

■ RDBMS

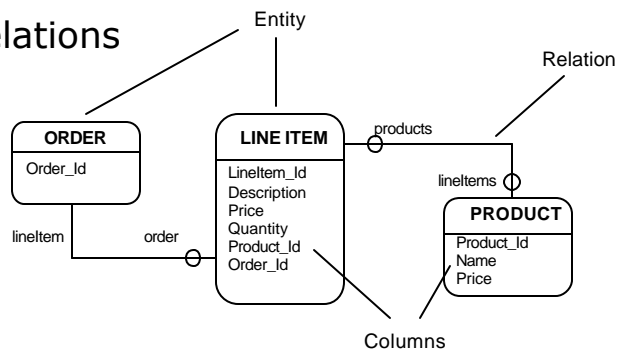
- Focus is on data
- Better suited for ad-hoc relationships and reporting application
- Expose data(column values)

■ Object Oriented system

- Focus is behavior
- Better suited to handle state-specific behavior where data is secondary
- Hide data(encapsulation)

The Relational Data Model

- Relational model is composed of
 - Entities
 - Relations

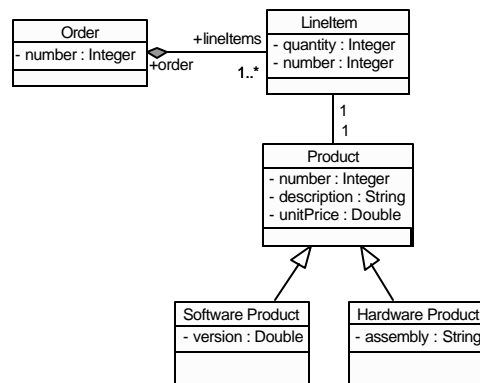


Copyright © 2002 DLS Associates (www.dlsa.com)

81

The Object Model

- The object model is composed of
 - Classes(attributes)
 - Associations

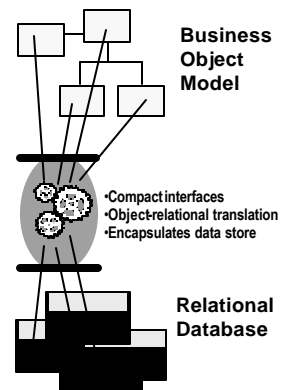


Copyright © 2002 DLS Associates (www.dlsa.com)

82

Persistence Frameworks

- The challenge
 - Changes shouldn't break the model
- The solution: An object-relational framework
 - Insulate application from physical storage
- The importance of the framework
 - 30% of development time is spent in accessing an RDBMS
 - Maintenance can be 60% of total cost



Copyright © 2002 DLS Associates (www.dlsa.com)

83

Object-Relational Framework: Characteristics

- Performance
 - Decomposing objects to data
 - Composing objects from data
- Minimize design compromises
 - Limit changes to object and relational models
- Extensibility
 - 15%-35% of the framework needs to be designed as an extensible framework

Copyright © 2002 DLS Associates (www.dlsa.com)

84

Emerging Standards

- J2EE Services (EJB containers)
 - Persistence
 - Transactions
 - Communications
- .NET Services
 - Data-awareness: ADO .NET

Copyright © 2002 DLS Associates (www.dlsa.com)

85

Extend the Framework

- Many to Many Mappings
- Inheritance
- Performance Considerations



Copyright © 2002 DLS Associates (www.dlsa.com)

86

Map Class Behavior to Stored Procedures

- Determine if any operations can be implemented as a stored procedure
- Candidate operations
 - Deal with persistent data
 - Any operations where a query is involved in a computation
 - Need to access the database to validate data

Copyright © 2002 DLS Associates (www.dlsa.com)

87

Data Modeling Expertise Applied

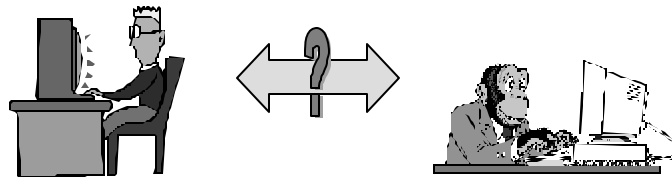
- Business modeling
- Functional requirements gathering
- Persistence mapping and framework development
- Extending framework for performance

Copyright © 2002 DLS Associates (www.dlsa.com)

88

Other Opportunities: Data Modeler ➡ Object Modeler

- Advantages
 - Understand value of modeling
 - Understand data as an asset
- Challenges
 - Shift from entity-focus to behavior-focus
 - Programming



Copyright © 2002 DLS Associates (www.dlsa.com)

89

Questions?

DLS Associates
Email: DavidS@dlsa.com
<http://www.dlsa.com>
(314) 236-5023



Copyright © 2002 DLS Associates (www.dlsa.com)

90