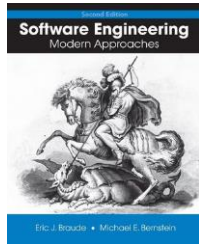


# Software Engineering

Modern Approaches



Eric Braude and Michael Bernstein

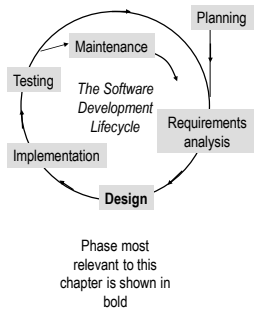
© 2010 John Wiley & Sons Ltd.

1

## Principles of Software Design

© 2010 John Wiley & Sons Ltd.

2



### Learning Goals of This Chapter

- What are the goals of software design?
- How do you use various design models for a single application?
- What are the use case model, the class models, the data flow models, and the state models?
- How are frameworks used in design?
- What are the IEEE standards for expressing designs?
- How does a team prepare for design in practice?

© 2010 John Wiley & Sons Ltd.

3

## Goals of Software Design

- **Sufficiency:** handles the requirements
- **Understandability:** can be understood by intended audience
- **Modularity:** divided into well-defined parts
- **Cohesion:** organized so like-minded elements are grouped together
- **Coupling:** organized to minimize dependence between elements

© 2010 John Wiley & Sons Ltd.

4

## Goals of Software Design

- **Robustness:** can deal with wide variety of input
- **Flexibility:** can be readily modified to handle changes in requirements
- **Reusability:** can use parts of the design and implementation in other applications

© 2010 John Wiley & Sons Ltd.

5

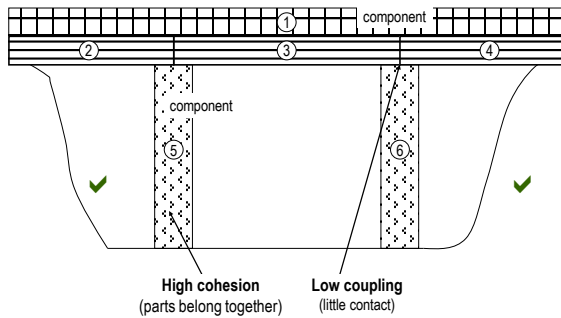
## Goals of Software Design

- **Information hiding:** module internals are hidden from others
- **Efficiency:** executes within acceptable time and space limits
- **Reliability:** executes with acceptable failure rate

© 2010 John Wiley & Sons Ltd.

6

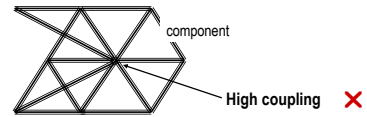
### Cohesion and Coupling: Bridge Example



© 2010 John Wiley & Sons Ltd.

7

### Questionable Architecture: High Coupling in Truss



© 2010 John Wiley & Sons Ltd.

8

### Aspects of Flexibility

- **Obtaining more or less of what's already present**

Example: handle more kinds of accounts without needing to change the existing design or code

- **Adding new kinds of functionality**

Example: add *withdraw* to existing *deposit* function

- **Changing functionality**

Example: allow withdrawals to create an overdraft

© 2010 John Wiley & Sons Ltd.

9

### Types of Reuse

We can reuse ....

- **Object code** (or equivalent)

Example: sharing dll's between word processor and spreadsheet

- **Classes** – in source code form

Example: Customer class used by several applications

Thus, we write generic code whenever possible

- **Assemblies of Related Classes**

Example: the java.awt package

- **Patterns** of Class Assemblies

© 2010 John Wiley & Sons Ltd.

10

### Types of Reuse

We can reuse ....

- **Object code** (or equivalent)

- **Classes** – in source code form

- **Assemblies of Related Classes**

- **Patterns** of Class Assemblies

© 2010 John Wiley & Sons Ltd.

11

### Integrating Design Models

Architecture drawings of a Office Building comprise:

- Front elevator
- Side elevator
- Electrical plan
- Plumbing plan
- etc

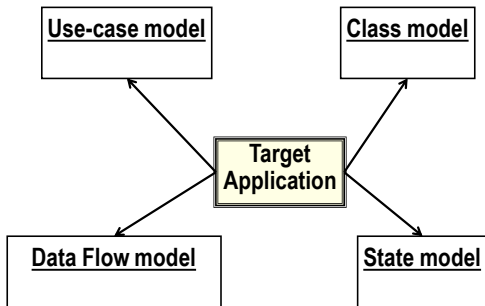
- Different views are required to express a building's architecture

- Several different views are also required to express a software design. They are called models

© 2010 John Wiley & Sons Ltd.

12

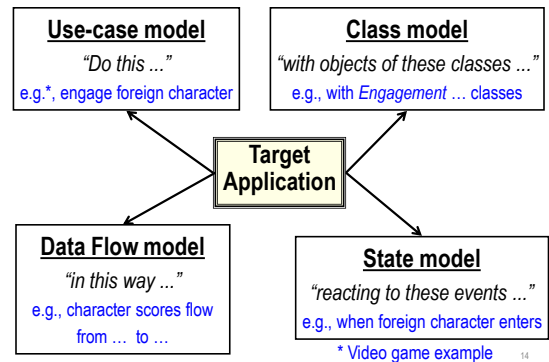
**Models** To express requirements, architecture & detailed design



© 2010 John Wiley & Sons Ltd.

13

**Models**

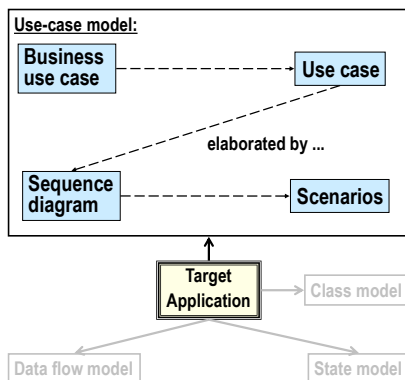


© 2010 John Wiley & Sons Ltd.

14

### Role of Use Case Models

Express what the application is supposed to do

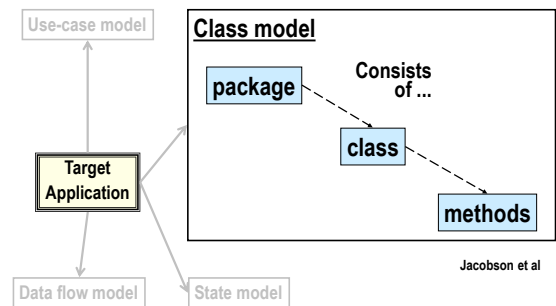


© 2010 John Wiley & Sons Ltd.

15

### Role of Class Models

Classes are building blocks of design



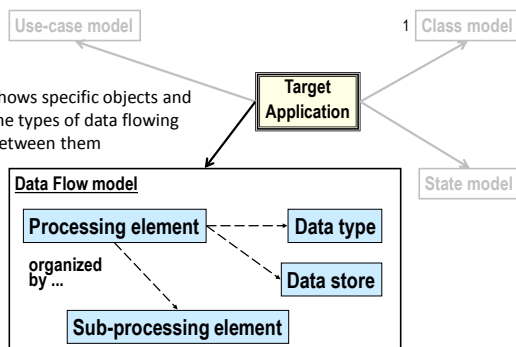
Jacobson et al

© 2010 John Wiley & Sons Ltd.

16

### Role of Data Flow Model

Shows specific objects and the types of data flowing between them



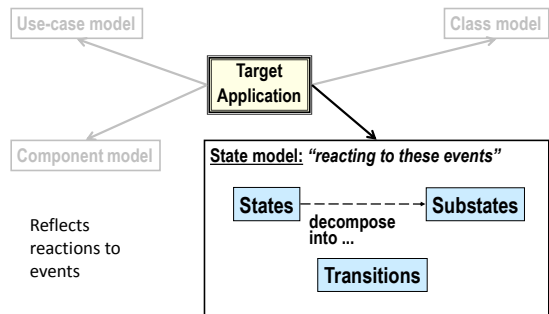
© 2010 John Wiley & Sons Ltd.

Jacobson et al

17

### Role of State Models

Reflects reactions to events

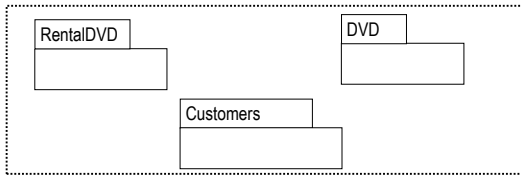


© 2010 John Wiley & Sons Ltd.

Jacobson et al

18

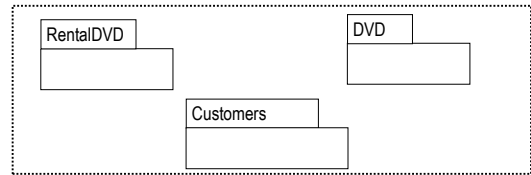
### Rental Framework



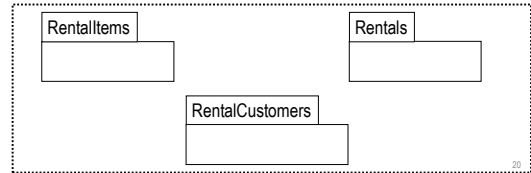
What do you think about these design considering reuse?

© 2010 John Wiley & Sons Ltd.

### Rental Framework

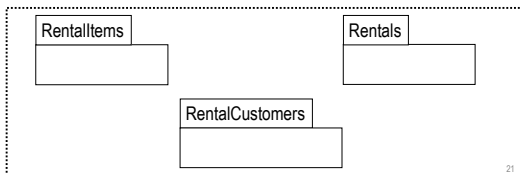


Is this other better?



© 2010 John Wiley & Sons Ltd.

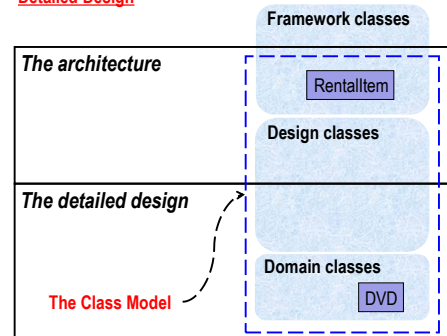
### Rental Framework



- Classes in a framework may be related
- Classes in a framework may be abstract or concrete
- Application may use them by means of inheritance, aggregation, or dependency
- A framework may feel like a generic application that we customize by inserting our own parts

© 2010 John Wiley & Sons Ltd.

### Class Model vs. Architecture and Detailed Design



© 2010 John Wiley & Sons Ltd.

### Design Comparison Example

Candidates	AI Pruitt's	Parallel Communicating Processes	Event Systems	Layered
<b>Qualities</b>				
<b>Sufficiency:</b> handles the requirements	1	1	2	2
<b>Understandability:</b> can be understood by intended audience	0	2	1	2
<b>Modularity:</b> divided into well-defined parts	0	0	1	2
<b>Cohesion:</b> organized so like-minded elements are grouped together	1	0	2	2

© 2010 John Wiley & Sons Ltd.

23

### Design Comparison Example ctd.

<b>Coupling:</b> organized to minimize dependence between elements	0	1	0	1
<b>Robustness:</b> can deal with wide variety of input	1	0	2	1
<b>Flexibility:</b> can be readily modified to handle changes in requirements	1	0	1	1
<b>Reusability:</b> can use parts of the design and implementation in other applications	0	0	1	2

© 2010 John Wiley & Sons Ltd.

24

### Design Comparison Example, concluded

<b>Information hiding:</b> module internals are hidden from others	1	1	2	2
<b>Efficiency:</b> executes within acceptable time and space limits	1	2	0	1
<b>Reliability:</b>	0	1	1	2
<b>TOTALS</b>	<b>6</b>	<b>8</b>	<b>13</b>	<b>18</b>