Software Architecture

Prof. Dr. Bertrand Meyer
Till G. Bay
Lecture 7: Introduction to Patterns, Model View Controller Pattern
What is a Pattern? (me, 10 min)
Model View Controller Pattern (me, 10 min)
Apply MVC to your Project (you, 5 min)
Pattern Categorization (me, 20 min)
ESDL installation (we, second lesson)
“A design pattern names, abstracts, and identifies the key aspects of a common design structure that make it useful for creating a reusable object-oriented design.”

Erich Gamma et al., *Design Patterns: Elements of Reusable Object-Oriented Software*, 1995, p 3.
A **design pattern** is given by one or more of

- A description of the pattern’s intent
- Use cases
- A software architecture for typical implementations
Model View Controller

Views

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>50</td>
<td>30</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>3</td>
<td>30</td>
<td>60</td>
</tr>
</tbody>
</table>

A = 50%
B = 30%
C = 20%

Model
Model View Controller (2/2)

**Model**
- Encapsulates application state
- Exposes application functionality
- Notifies view of changes

**View**
- Renders the model
- Sends user gestures to controller
- Allows controller to select view

**Controller**
- Defines application behavior
- Maps user actions to model updates
- Selects view for response
- One for each functionality

- Change Notification
- State change
- View selection
- User gestures
- Feature calls
- Events
GoF’s description of a design pattern

- Pattern name and classification
- Intent
- Also known as
- Motivation
- Applicability
- Structure
- Participants
- Collaborations
- Consequences
- Implementation
- Sample code
- Known uses
- Related patterns
The GoF design patterns

- **Creational**
  - Abstract Factory
  - Builder
  - Factory Method
  - Prototype
  - Singleton

- **Structural**
  - Adapter
  - Bridge
  - Composite
  - Decorator
  - Façade
  - Flyweight
  - Proxy

- **Behavioral**
  - Chain of Responsibility
  - Command
  - Interpreter
  - Iterator
  - Mediator
  - Memento
  - Observer
  - State
  - Strategy
  - Template Method
  - Visitor
Creational design patterns (1/2)

- **Creational**
  - Abstract Factory
  - Builder
  - Factory Method
  - Prototype
  - Singleton

- **Structural**
  - Adapter
  - Bridge
  - Composite
  - Decorator
  - Façade
  - Flyweight
  - Proxy

- **Behavioral**
  - Chain of Responsibility
  - Command
  - Interpreter
  - Iterator
  - Mediator
  - Memento
  - Observer
  - State
  - Strategy
  - Template Method
  - Visitor
Creational design patterns (2/2)

- **Goal:**
  - Put more flexibility into the instantiation process

- **How:**
  - Through inheritance or delegation

- **What:**
  - Defer parts of object creation
Structural design patterns (1/2)

- **Creational**
  - Abstract Factory
  - Builder
  - Factory Method
  - Prototype
  - Singleton

- **Structural**
  - Adapter
  - Bridge
  - Composite
  - Decorator
  - Façade
  - Flyweight
  - Proxy

- **Behavioral**
  - Chain of Responsibility
  - Command
  - Interpreter
  - Iterator
  - Mediator
  - Memento
  - Observer
  - State
  - Strategy
  - Template Method
  - Visitor
Structural design patterns (2/2)

- **Goal:**
  - Compose software elements into bigger structures

- **How:**
  - Through inheritance (static binding) or composition (flexibility)
Behavioral design patterns (1/2)

- **Creational**
  - Abstract Factory
  - Builder
  - Factory Method
  - Prototype
  - Singleton

- **Structural**
  - Adapter
  - Bridge
  - Composite
  - Decorator
  - Façade
  - Flyweight
  - Proxy

- **Behavioral**
  - Chain of Responsibility
  - Command
  - Interpreter
  - Iterator
  - Mediator
  - Memento
  - Observer
  - State
  - Strategy
  - Template Method
  - Visitor
Behavioral design patterns (2/2)

- **Deal with:**
  - Algorithms
  - Assignment of responsibilities between objects
  - Communication between objects

- **How:**
  - Through inheritance or composition
Exercise 5

- Start implementing your project now
- We will put an updated version of Exercise 5 on the web
Manually Deinstalling ESDL (1/2)

- Remove Environment Variables
  - ESDL
  - GOBO
  - EWG
  - SDL
  - SDL_HEADER
  - ISE_EIFFEL
  - ISE_C_COMPILER
  - GOBO_CC
  - GOBO_EIFFEL
Manually Deinstalling ESDL (2/2)

- Delete ESDL directory
- Delete dll’s in Windows System Directory
  - jpeg.dll
  - libpng1.dll
  - SDL.dll
  - SDL_image.dll
  - SDL_mixer.dll
  - sdlgfx.dll
  - zlib.dll
- Reboot
Installing ESDL 0.6.0

- Download EiffelStudio 5.5
- [http://se.inf.ethz.ch/download/games/developer](http://se.inf.ethz.ch/download/games/developer)
  - Download esdl_bcb_0.6.0.exe and install
End of lecture 7