CSC7203 : Advanced Object Oriented Development

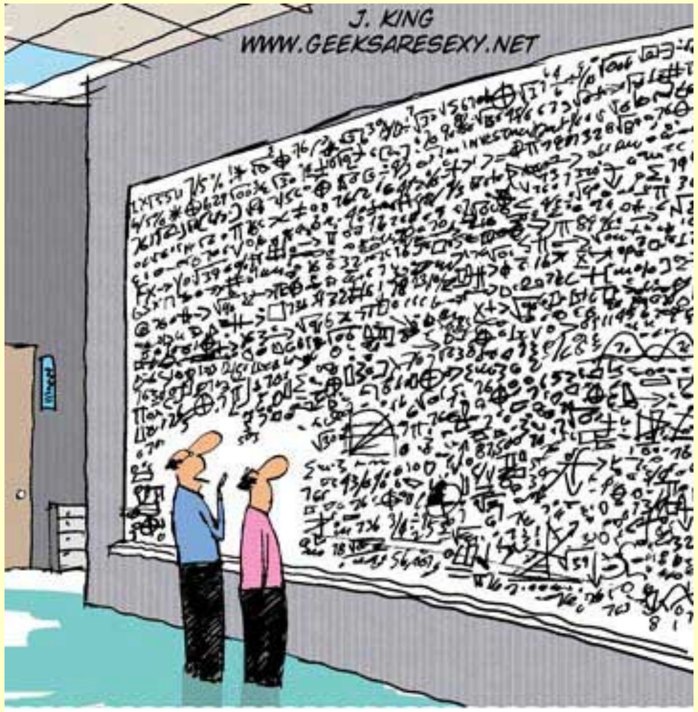
J Paul Gibson, D311

paul.gibson@telecom-sudparis.eu

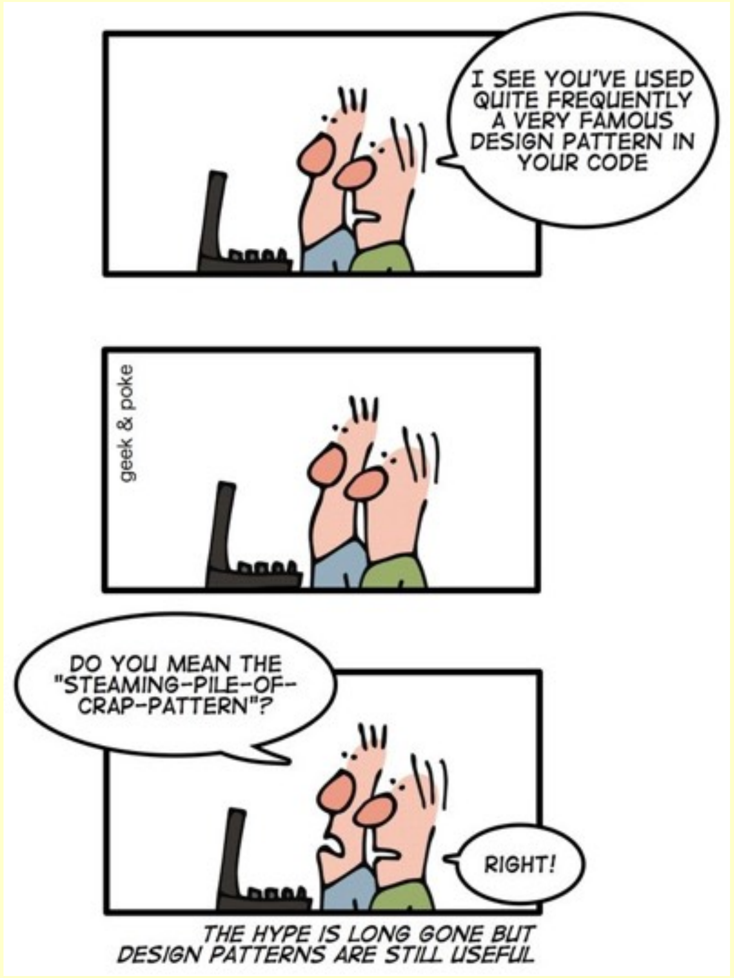
http://www-public.tem-tsp.eu/~gibson/Teaching/CSC7203/

Design Patterns

…/~gibson/Teaching/CSC7203/CSC7203-AdvancedOO-L2.pdf



"...And that, in simple terms, is what's wrong with your software design."


I SEE YOU'VE USED QUITE FREQUENTLY A VERY FAMOUS DESIGN PATTERN IN YOUR CODE
geek & poke
DO YOU MEAN THE "STEAMING-PILE-OF-CRAP-PATTERN"?
RIGHT!
THE HYPE IS LONG GONE BUT DESIGN PATTERNS ARE STILL USEFUL

Introduction

pattern noun **1** a model, guide or set of instructions for making something • *a dress pattern*. **2** a decorative design, often consisting of repeated motifs, eg on wallpaper or fabric. **3** a piece, eg of fabric, as a sample. **4** any excellent example suitable for imitation. **5** a coherent series of occurrences or set of features • *a pattern of events*. verb (***patterned***, ***patterning***) (*usually* **pattern something on another thing**) to model it on another type, design, etc.

ETYMOLOGY: 14c as *patron*; French, from Latin *patronus* example or defender.

SOURCE - Chambers 21st Century Dictionary

Introduction

design verb (***designed***, ***designing***) **1** to develop or prepare a plan, drawing or model of something before it is built or made. **2** *formal* to plan, intend or develop something for a particular purpose. noun **1** a plan, drawing or model showing how something is to be made. **2** the art or job of making such drawings, plans, etc. **3** the way in which something has been made. **4** a picture, pattern, arrangement of shapes, etc used eg as decoration. **5** a plan, purpose or intention. **designable** adj. **designedly** adverb intentionally; on purpose. **designing** adj, *derog* using cunning and deceit to achieve a purpose. **designingly** adverb. **by design** intentionally. **have designs on someone** or **something** to have plans to appropriate them or it.

ETYMOLOGY: 16c: from French *désigner*.

SOURCE - Chambers 21st Century Dictionary

Introduction

design pattern
programming
A description of an object-oriented design technique which names, abstracts and identifies aspects of a design structure that are useful for creating an object-oriented design. The design pattern identifies classes and instances, their roles, collaborations and responsibilities. Each design pattern focuses on a particular object-oriented design problem or issue. It describes when it applies, whether it can be applied in the presence of other design constraints, and the consequences and trade-offs of its use.
Home (http://st-www.cs.uiuc.edu/users/patterns/patterns.html)
["Design Patterns: Elements of Reusable Object-Oriented Software", Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides].
(1997-07-21)

SOURCE – Free On-Line Dictionary Of Computing (http://foldoc.org/)

Introduction

1970’s - **Christopher Alexander** (an architect) worked on « *Pattern Languages* » which were applied in many domains:

a structured method of describing good design practices within a field of expertise.

Using a pattern language permits non-experts in a field to successfully solve very large, complex design problems.

A single problem is documented with its typical place (the syntax), and use (the grammar) with the most common and recognized good solution seen in the real world, like the entries seen in dictionaries.

Such an entry is considered to be a single design pattern; and a rich set of patterns form a language.

Introduction

Early research:

THING-MODEL-VIEW-EDITOR an Example from a planning system, Trygve Reenskaug, 1979.

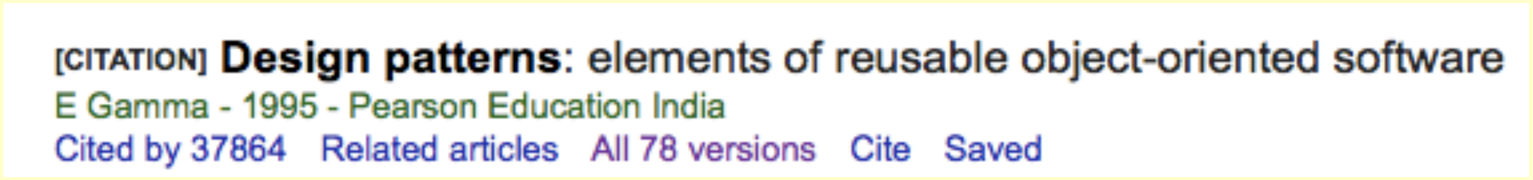
Using Pattern Languages for Object-Oriented Programs,Kent Beck and Ward Cunningham, 1987.

A Cookbook for Using View-Controller User the Model-Interface Paradigm in Smalltalk-80, Krasner and Pope, 1988.

Introduction

Breakthrough into mainstream: Gang of Four (GOF)

Design Patterns: Abstraction and Reuse of Object-Oriented Design, Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides, 1993


[CITATION] **Design patterns**: elements of reusable object-oriented software
E Gamma - 1995 - Pearson Education India
Cited by 37864 Related articles All 78 versions Cite Saved

Introduction

Further reading:

•*Design Patterns for Object-Oriented Software Development*, Pree and Sikora, 1997

•*The origins of pattern theory: the future of the theory, and the generation of a living world*, C. Alexander, 1999

•*Software Factories Assembling Applications with Patterns, Models, Frameworks and Tools*, Greenfield and Short, 2003.

•*The Model-View-Controller (MVC) Its Past and Present*, Trygve Reenskaug, 2003.

•*What Is Web 2.0: Design Patterns and Business Models for the Next Generation of Software*, O'Reilly, 2007

Definitions

Alexander :

As an element in the world, each pattern is a relationship between a certain context, a certain system of forces which occurs repeatedly in that context, and a certain spatial configuration which allows these forces to resolve themselves.

As an element of language, a pattern is an instruction, which shows how this spatial configuration can be used, over and over again, to resolve the given system of forces, wherever the context makes it relevant.

Each pattern describes a problem that occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice.

Definitions

Erich Gamma (GoF):

Patterns provide you with tools that help you with design problems. They do so not by giving a pat solution but by explaining trade-offs. Even though patterns are abstracted from concrete uses, they also provide you valuable implementation hints. From my perspective it is the fact that patterns are implementable that makes them so valuable.

Patterns are distilled from the experiences of experts. They enable you to repeat a successful design done by someone else. However, since patterns enable many implementation variations you still have to keep the brain turned on.

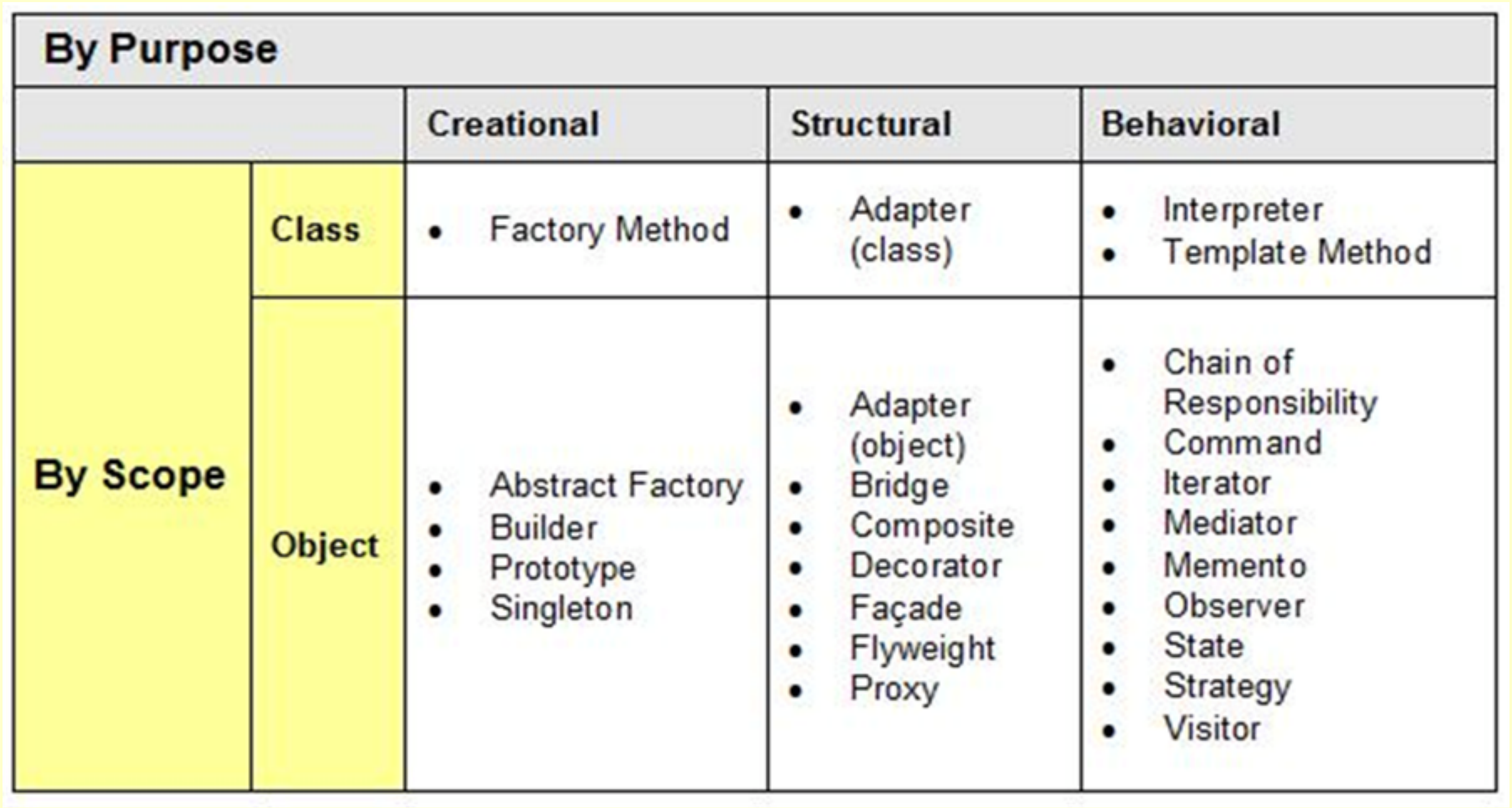
Since patterns provide you with names for design building blocks they provide you with a vocabulary to describe and discuss a particular design.

I think patterns as a whole can help people learn object-oriented thinking: how you can leverage polymorphism, design for composition, delegation, balance responsibilities, and provide pluggable behavior.

Elements of Design Patterns

- Pattern Name
- Problem statement -context where it might be applied
- Solution - elements of the design, their relations, responsibilities, and collaborations (including a template of the solution)
- Consequences - Results and trade-offs

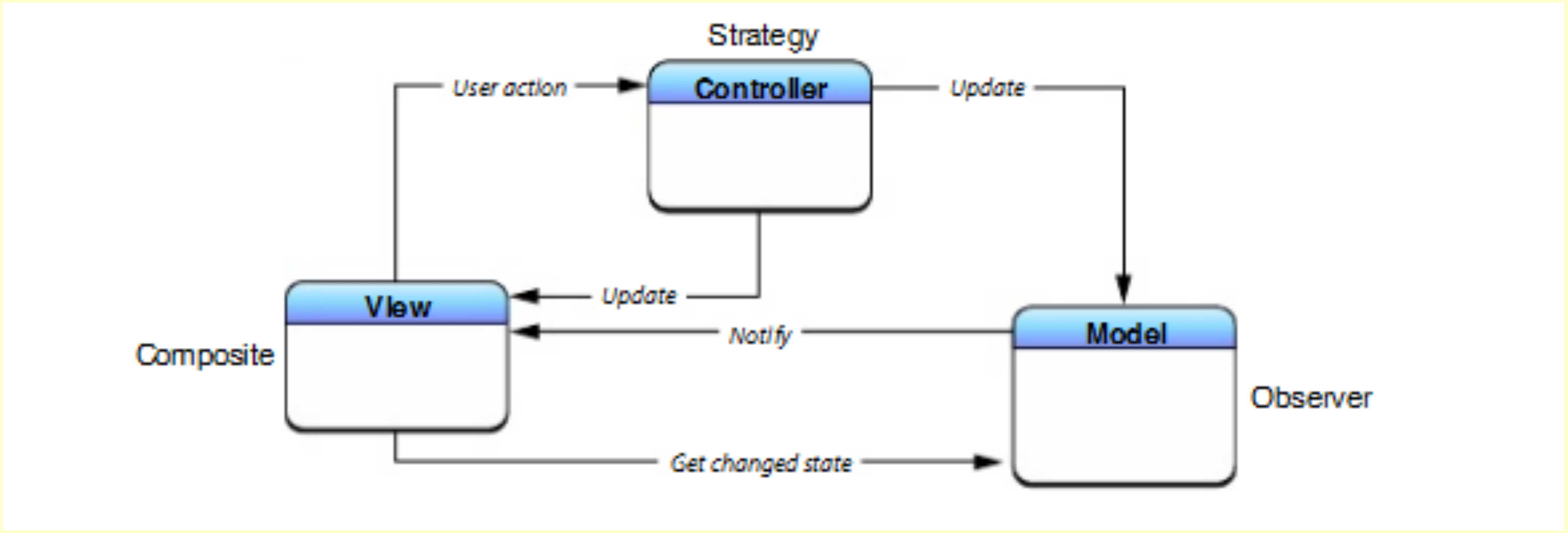
The 23 GoF patterns are generally considered the foundation for all other patterns. They are categorized in three groups: *Creational, Structural, and Behavioral.*



By Purpose				
		Creational	Structural	Behavioral
By Scope	Class	• Factory Method	• Adapter (class)	• Interpreter • Template Method
	Object	• Abstract Factory • Builder • Prototype • Singleton	• Adapter (object) • Bridge • Composite • Decorator • Façade • Flyweight • Proxy	• Chain of Responsibility • Command • Iterator • Mediator • Memento • Observer • State • Strategy • Visitor

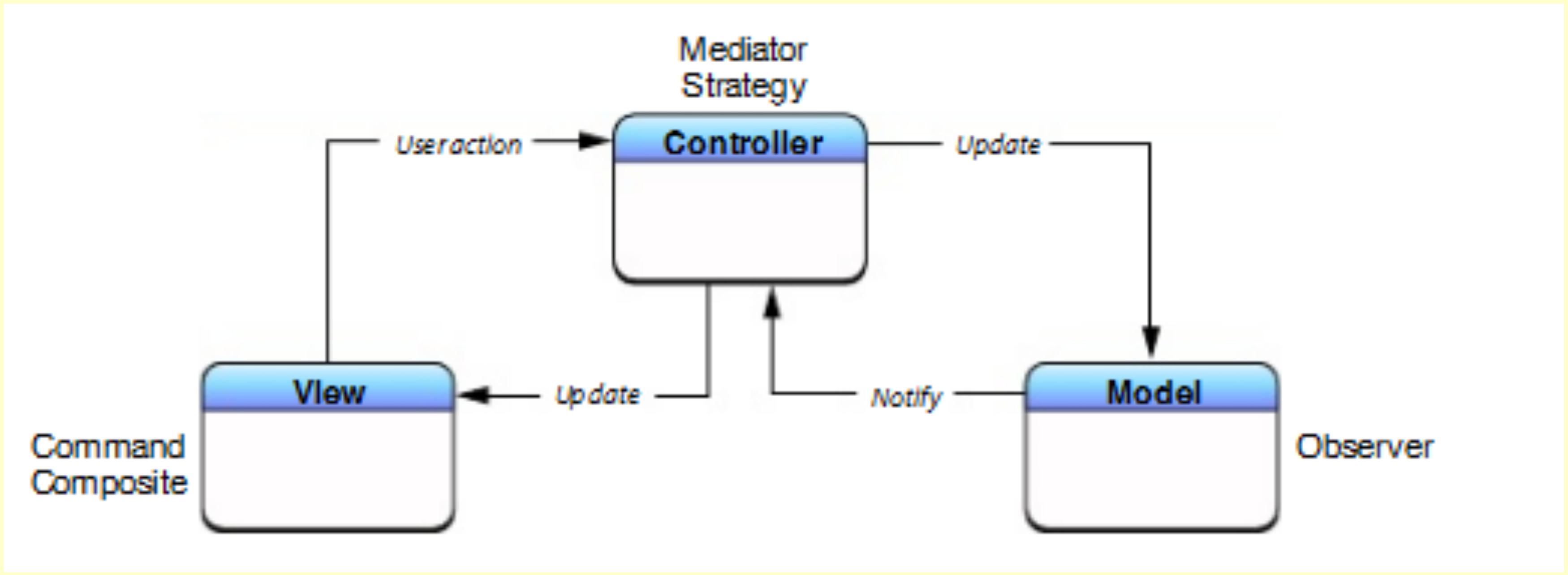
Design pattern composition: MVC example

Traditional version of MVC as a compound pattern



Design pattern composition: MVC example

Alternative version of MVC as a compound pattern



Learning by PBL

•The best way to learn about patterns is to look at examples.

•We shall do these in UML/Java, but any (OO) language can be used to model/implement/re-use a pattern.

•It is best if you discover patterns yourself, rather than being shown them – but this is not guaranteed to happen!

•You can also read about them. A good web site is:

http://sourcemaking.com/design_patterns