

CS 635 Advanced Object-Oriented Design & Programming
Spring Semester, 2011
Doc 18 Facade & Flyweight
April 20, 2011

Copyright ©, All rights reserved. 2011 SDSU & Roger Whitney, 5500 Campanile Drive, San Diego, CA 92182-7700 USA. OpenContent (<http://www.opencontent.org/opl.shtml>) license defines the copyright on this document.

References

Design Patterns: Elements of Reusable Object-Oriented Software, Gamma, Helm, Johnson, Vlissides, 1995, pp. 185-194, 195-206

This Car Runs on Code, <http://spectrum.ieee.org/green-tech/advanced-cars/this-car-runs-on-code>

Source lines of code, http://en.wikipedia.org/wiki/Source_lines_of_code



Size

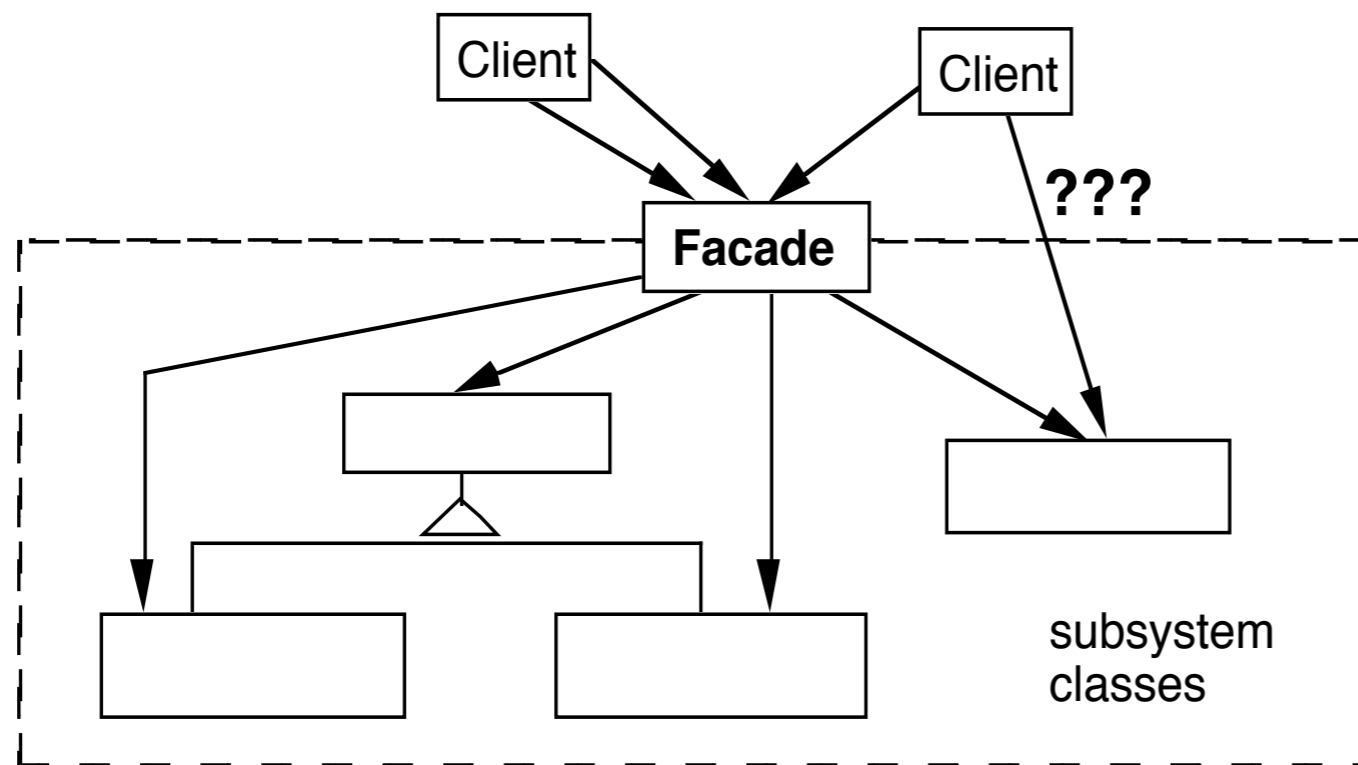
Item	Source Lines of Code (Millions)
F-22 Raptor US jet fighter	1.7
Boeing 787	6.5
S-class Mercedes-Benz radio & navigation system	20
Mac OS 10.4	86
Premium class automobile	~100
Debian 4.0	283

Design Patterns text contains under 8,000 lines

The Facade Pattern

Create a class that is the interface to the subsystem

Clients interface with the Facade class to deal with the subsystem



Consequences of Facade Pattern

It hides the implementation of the subsystem from clients

It promotes weak coupling between the subsystems and its clients

It does not prevent clients from using subsystem classes directly, should it?

Facade does not add new functionality to the subsystem

Public versus Private Subsystem classes

Some classes of a subsystem are

- public

 - facade

- private

Compiler Example

The VisualWorks Smalltalk compiler system has 75 classes

Programmers only use Compiler, which uses the other classes

Compiler evaluate: '100 factorial'

```
| method compiler |  
method := 'reset'  
"Resets the counter to zero"  
count := 0.'
```

```
compiler := Compiler new.  
compiler  
  parse:method  
  in: Counter  
  notifying: nil
```


Flyweight

Use sharing to support large number of fine-grained objects efficiently

Text Example

A document has many instances of the character 'a'

Character has

- Font

- width

- Height

- Ascenders

- Descenders

- Where it is in the document

Most of these are the same for all instances of 'a'

Use one object to represent all instances of 'a'

Java String Example

```
public void testInterned() {  
    String a1 = "catrat";  
    String a2 = "cat";  
    assertFalse(a1 == (a2+ "rat"));  
  
    String a3 = (a2 + "rat").intern();  
    assertTrue(a1 == a3);  
    String a4 = "cat" + "rat";  
    assertTrue(a1 == a4);  
    assertTrue(a3 == a4);  
}
```

public String intern()

Returns a canonical representation for the string object.

A pool of strings, initially empty, is maintained privately by the class String.

Intrinsic State

Information that is independent from the object's context

The information that can be shared among many objects

So can be stored inside of the flyweight

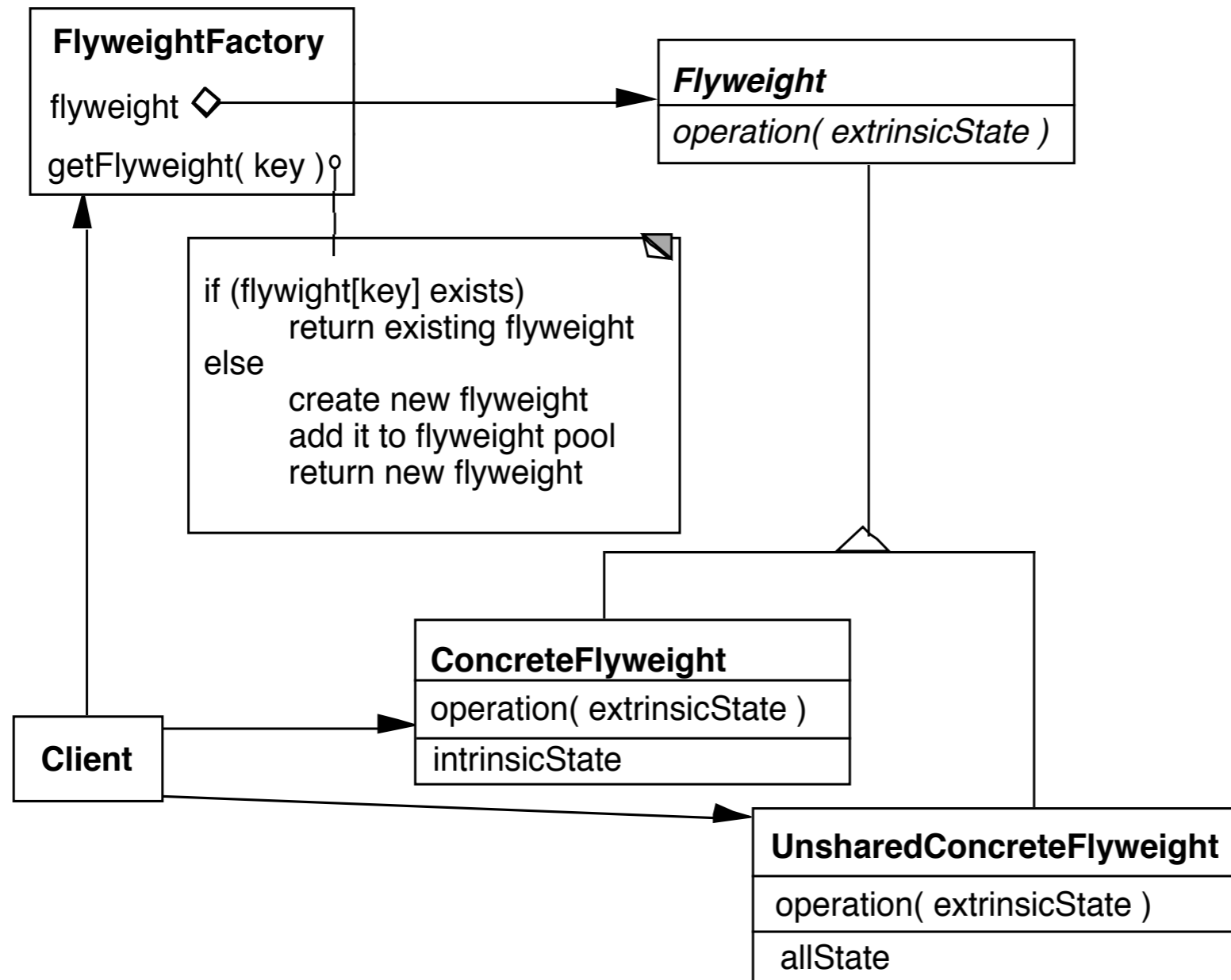
Extrinsic State

Information that is dependent on the object's context

The information that can not be shared among objects

So has to be stored outside of the flyweight

Structure



The Hard Part

Separating state from the flyweight

How easy is it to identify and remove extrinsic state

Will it save space to remove extrinsic state

Example Text

Run Arrays

aaaaabaaaaaaaaaaaaaaaaaaaaa



a b a
5 1 20

Text Example

Lexi Document Editor

Uses character objects with font information
(To support graphic elements)

"A Cat in the hat came **back** the very next day"

Use run array to store font information (extrinsic state)

Normal	Bold	Normal
22	4	18