

Builder Design Pattern



Frank Dziubak
University of Chicago
CSPP 51023
OO Architecture
Winter 2011

Builder - an Object Creational Pattern

Creational Patterns:

- abstract the Instantiation Process
- help make a system independent of object creation, composition, and representation

“Object” Creational Patterns:

- delegate instantiation to another object
- becoming more important; systems are coming to depend more on object composition than on class inheritance



Builder

The Builder

What is it Good For?

Managing Complexity



Builder



Builder

Use it when :

- You must build a complex object
- You must build different varieties
- Design has to be independent of parts and assembly

Builder has Four Classes

Builder

- the Abstract Interface

Concrete Builder

- the Implementation
- An object able to construct other objects
- Constructs and assembles the parts



Builder

Director

- Manages correct sequence of object creation
- Receives a Concrete Builder as parameter
- Executes the necessary operations.

Product

- the End Result; the Goal; the Output; the *Raison d'être*



But we have to get serious.

How serious?

War



War...huh...
What is it good for?
Absolutely nothing
Uh ha haa ha

“War”

written by [Norman Whitfield](#)
and [Barrett Strong](#)

for Motown Records

1969

Recorded by Edwin Starr

. . . None-the-less . . .

The Navy needs ships “Product Class”



Client

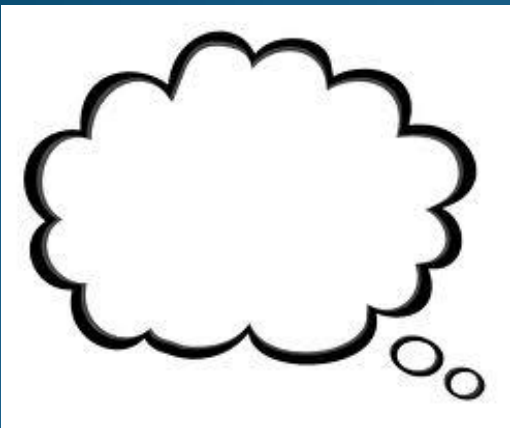
I'll have one
aircraft carrier, a
cruiser, and a side of
Marines.



“Director”



**Commander, Naval Sea Systems Command
(ComNavSeaSysCom)
“We Build Ships!”**





I'll use the
"Builder"



“Builder” Class Shipyard (Let’s Build ‘em, boys!)



Who's in Our 4 Classes?

Product: Warship

Director: ComNavSeaSysCom (Curly)

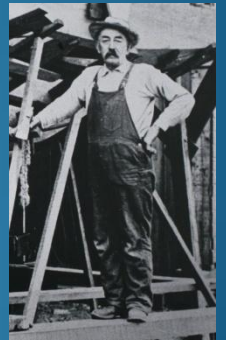
Builder: Shipyard

ConcreteBuilders:

IowaShipyard (Philadelphia)

TicoShipyard (Pascagoula)

NimitzShipyard (Newport News)



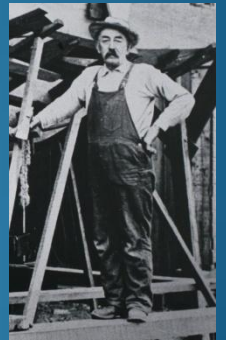
Builder

What's in Our Parts List?

- Engines
- Main Battery (Weapons)

We'll call these "Parts" also:

- ShipType (cruiser, carrier, etc.)
- ShipClass (Nimitz, Iowa, etc.)

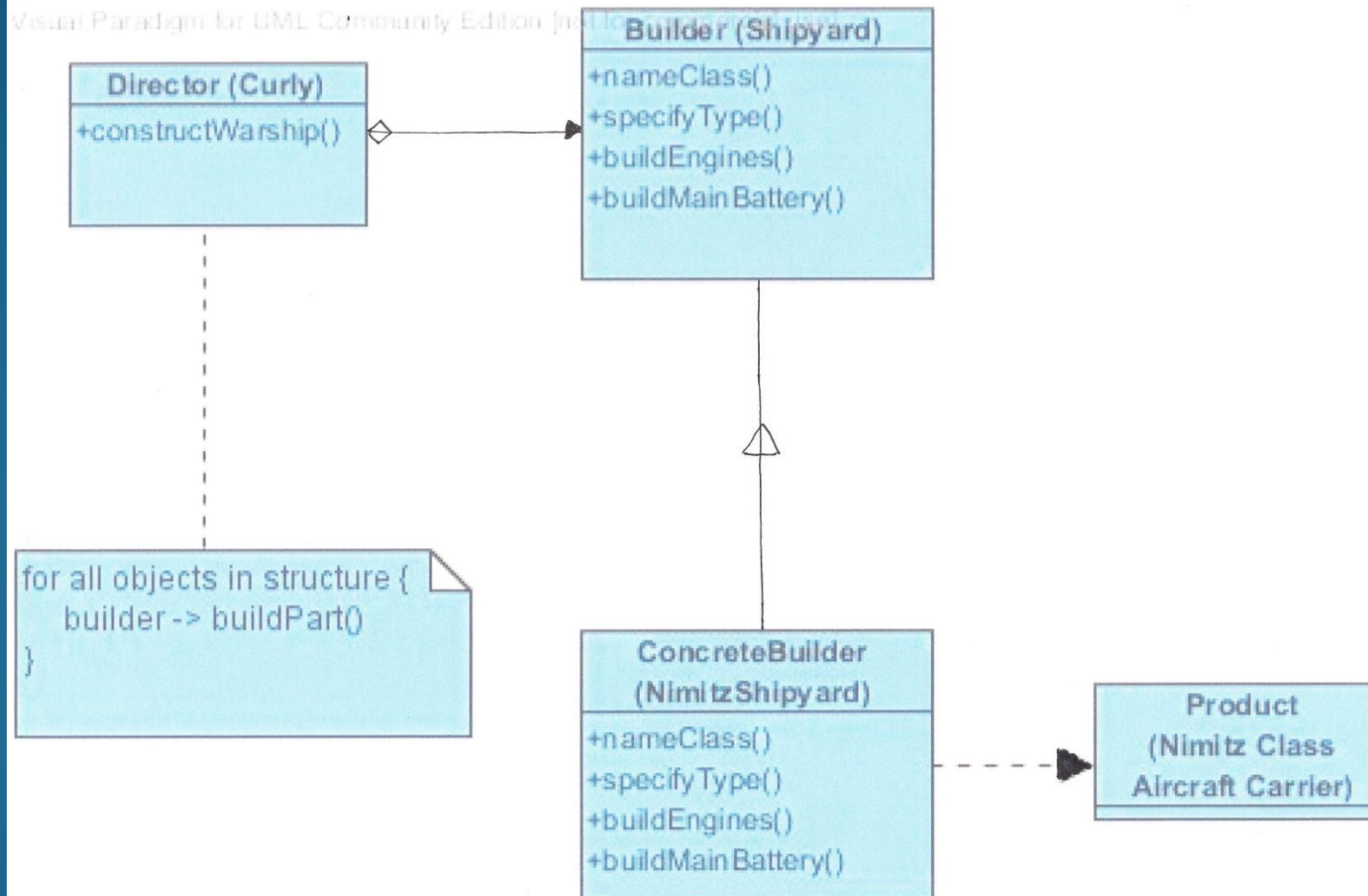


Builder

Structure



Builder



```

/*          "Product"          */

class Warship {
    // Private variables
    private String className;
    private String type;
    private String engines;
    private String battery;

    // Mutators (setter) methods
    public void setClassName(String className) {
        this.className = className; }
    public void setType(String type) {
        this.type = type; }
    public void setEngines(String engines) {
        this.engines = engines; }
    public void setBattery(String battery) {
        this.battery = battery; }

    // Accessors (getter) methods
    public String getClassName() {
        return className; }
    public String getType() {
        return type; }
    public String getEngines() {
        return engines; }
    public String getBattery() {
        return battery; }
}

```



Product

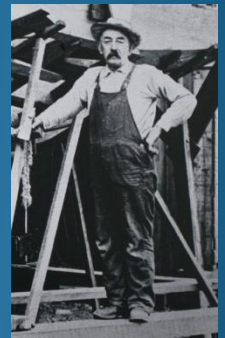

```
/*          "Builder Class"          */
/*          (Abstract Interface)      */

abstract class Shipyard {
    protected Warship warship;

    public Warship getWarship() {
        return warship;
    }

    public void createNewWarshipClass() {
        warship = new Warship();
    }

    // abstract methods
    public abstract void nameClass();
    public abstract void specifyType();
    public abstract void buildEngines();
    public abstract void buildMainBattery();
}
```

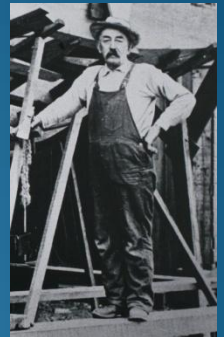


Builder


```
/*      "ConcreteBuilder Class"      */
/*      (the Implementation)         */

class IowaShipyard extends Shipyard {

    public void nameClass() {
        warship.setClassName("Iowa");
    }
    public void specifyType() {
        warship.setType("Battleship");
    }
    public void buildEngines() {
        warship.setEngines("8 steam turbines");
    }
    public void buildMainBattery() {
        warship.setBattery("9 x 16-inch guns");
    }
}
```



Builder

```
/*          "Director"          */

class ComNavSeaSysCom{
    private Shipyard shipyard;

    public void setShipyard(Shipyard sh) {
        shipyard = sh; }

    public Warship getShip() {
        return shipyard.getWarship();}

    public void buildWarship() {
        shipyard.createNewWarshipClass();
        shipyard.nameClass();
        shipyard.specifyType();
        shipyard.buildEngines();
        shipyard.buildMainBattery();
    }
}
```



Director

```
public class Driver {  
    public static void main(String[] args) {  
  
        ComNavSeaSysCom curly = new ComNavSeaSysCom();  
  
        Shipyard IngallsPascagoula = new TicoShipyard();  
  
        curly.setShipyard(IngallsPascagoula);  
        curly.buildWarship();  
  
        Warship valleyForge = buildDirector.getShip();  
        System.out.println("Shiloh (CG-71)");  
        christenShip(valleyForge);  
  
        } // end main  
  
    public static void christenShip(Warship thisShip){  
        System.out.println("    Class:      " + thisShip.getClassName());  
        System.out.println("    Type:      " + thisShip.getType());  
        System.out.println("    Propulsion:  " + thisShip.getEngines());  
        System.out.println("    Main battery: " + thisShip.getBattery());  
        System.out.println(); }  
}
```

Christen Ship



Product

USS Missouri (BB-63)

Class: Iowa

Type: Battleship

Propulsion: 8 steam turbines

Main battery: 9 x 16-inch guns

Product



USS Shiloh (CG-67)

Class: Ticonderoga

Type: Cruiser

Propulsion: 4 gas turbine engines

Main battery: 122 Vertical-launch missiles

Product



USS Theodore Roosevelt (CVN-71)

Class: Nimitz

Type: Aircraft Carrier

Propulsion: 2 nuclear reactors

Main battery: F/A-18 Super Hornet fighter jets

Product



USS Enterprise (NCC-1701)

Class: Constitution
Type: Starship
Propulsion: Matter/anti-matter warp drive
Main battery: Photon torpedoes / phasers

**This presentation
has been brought to you by the
Department of the Navy**

