

Slides #17

# Design Patterns & Implementation Issues

Sections 7.2 – 7.4

CMPT 276  
© Dr. B. Fraser

15-03-16

1

## Topics

- 1) What are design patterns?
- 2) What are some general implementation issues?
- 3) What is open-source development?

15-03-16

2

## Design patterns

15-03-16

3

## Design patterns

- A pattern is:
  - It should be sufficiently abstract to be reused...
  - Pattern descriptions usually make use of object-oriented characteristics such as inheritance and polymorphism.

15-03-16

4

# Observer pattern motivation

For  
billionaires!

- Imagine you are writing an automatic day-planner:
  - It can take the user's interests, plus information about the world, and suggest what they should do.
- Possible design idea:
  - You want to use different objects for cultural planning, sports planning, and sight-seeing.
  - Some objects bring in information about the world; your planning-objects use these other objects.
- Challenge:
  - All of these objects need to know the weather.
  - Your weather object gets updates now and then.
  - How do you tell..

15-03-16

5

# Possible Idea

- Have the weather object call each other object:

```
void newDataUpdate() {
    String weatherData = ...;
    culturePlanner.update(weatherData);
    sportsPlanner.update(weatherData);
    sightseeingPlanner.update(weatherData);
    // Change here EVERY time you get a new planner.
}
```
- Bad because:
  - Weather object is...
  - Every new planner you get, you'll have to change the weather object's code, recompile, and re-run.

15-03-16

6

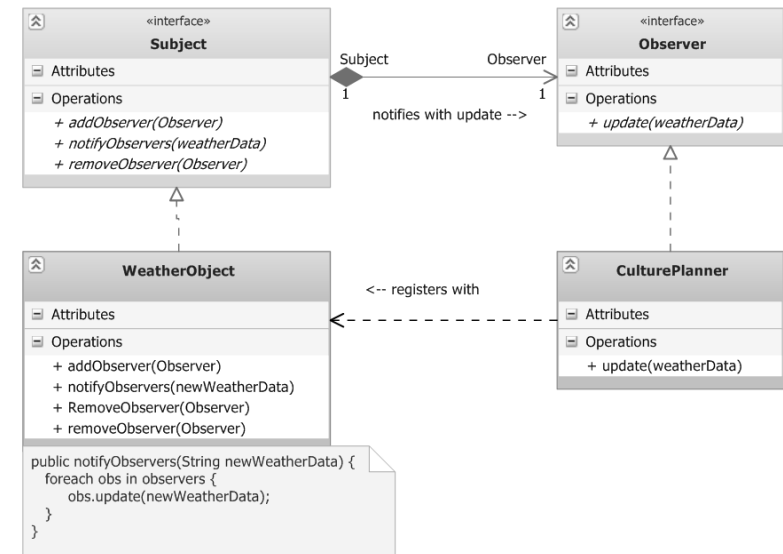
# The observer pattern

- Observer Pattern:
- Produces a one to many relationship:
  - one object observed (called the subject)
  - many objects observing (called the observers).
- Great because it loosely couples objects:
  - Object with something to report does not need a hard-coded list of who to tell; ...

15-03-16

7

# The observer pattern – weather data

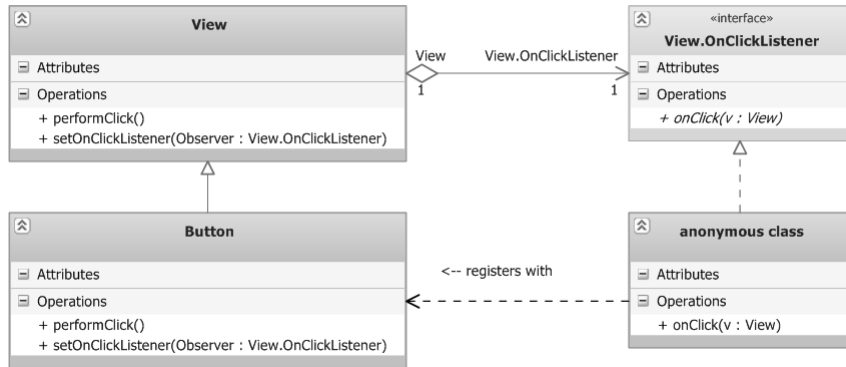


15-03-16

8

## The observer pattern – Android button

```
// Handle yes button
Button yesButton = (Button) findViewById(R.id.Button_Yes);
yesButton.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        // do something useful!
    }
});
```

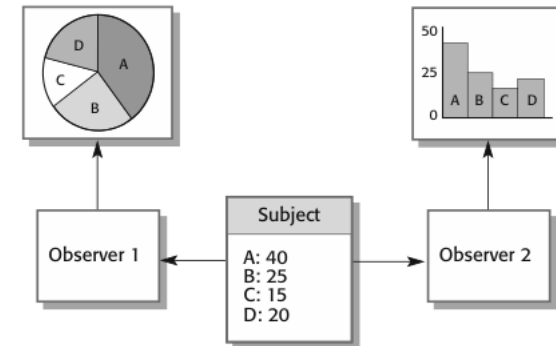


15-03-16

9

## Multiple observers

- Often have multiple observers listening to the same subject.



15-03-16

10

## Other design patterns in Android

- Android framework uses many design patterns:
  - Observers – Button presses
  - Singleton objects: Only ever one object.

```
SharedPreferences settings =
    getSharedPreferences( GAME_PREFERENCES,
        MODE_PRIVATE);
```

- Factory:
  - Runtime decision as to which exact class to create.
    - Ex: Intents
- ... Many more.

15-03-16

11

## Implementation issues

15-03-16

12

## Implementation issues

- We are not going to cover how to code; just mention some issues that apply to software engineering.
- - How can we use existing code to create our new system?
- - Revision control, component version control (releases...)
- - How can we develop on one style of machine and run on another? (Ex: PC to Android).

15-03-16

13

## Levels of reuse

- The abstraction level:
  - Reuse the... in the design of the software.
  - Ex: Design patterns, architectural patterns.
- The object level
  - Reuse vs rewriting the code.
  - Ex: Library objects: XML parsing, container classes.
- The component level
  - Reuse a collection of objects such as...
  - Ex: Android UI framework.
- The system level
  - Reuse... (COTS=Commercial off-the-shelf)

15-03-16

14

## Reuse cost

- Cost of Software Reuse:
  - time to find and evaluate COTS for reuse.
  - \$ to buy the software; can be expensive!
  - \$ to adapt, configure and integrate components.
- Reusing well tested component can...
- However:
  - Many disasters caused by reusing software which had an unknown bug.
  - We tend not to test them well enough because..

15-03-16

15

## Caution on reuse

- Therac-25: Canadian made radiation therapy machine. Failure...
  - Reused buggy software that \*relied\* on hardware safeties, which were left out in the later version.
- Ariane 5 rocket: Initial test flight...
  - Reused a module from Ariane 4 which converted a floating point number to a 16bit integer.
  - Ariane 4 rocket never encountered an error.
  - Exception handling was turned off for efficiency.
  - Both primary and backup computers encountered the error at the same time and shutdown.
- 

15-03-16

16

## Configuration management

- - Version control/source-code control.
  - Ex: Git
- - Control the version of components used to build releases of the system.
    - Select the Linux kernel version.
  - Ex: Unix make, Java ANT, Gradle
- - Track bugs: bug entry, priority, assignment, investigation, fix, verification.
  - Ex: Trac, Bugzilla, Phabricator

15-03-16

17

## Host-target development

- Most software is developed on one computer... but runs on a different target platform...
- A platform is more than just hardware.
  - Includes operating system and database management system.
- Host machine usually has:
  - different... than target;
  - different... than target.

15-03-16

18

## Development platform (host) tools

- IDE (integrated development environment) :  
Android Studio
- Compiler:
  - Called a... if compiling on one machine for another.
- A language debugging system : DDMS
- Emulation tools: Android emulator.
- Testing tools: Junit for automatic unit-testing.
- Version control tools: Git, SVN, CVS, ...

15-03-16

19

Open source development

15-03-16

20

## Open source development

- Open source development is:  
software development where the source code of a system is published and volunteers are invited to participate in its development.
- Open source systems
  - Linux operating system:
    - Used in servers, developers, mobile phones, etc
  - Android, Apache web server, mySQL (database), LibreOffice.

15-03-16

21

## Open source issues

- Open Source Issues:
  - Should our product...
  - Should an open source approach be...
- More companies are using open source development.
  - Business model is not reliant on selling software but on...
- Possible advantages of open source:
  - developed cheaper and faster,
  - creates a community of users for the software.

15-03-16

22

## Open source licensing

- Open-source =
  - Does not mean that anyone can do as they wish with that code.
- Developer (company or individual) still owns the code and can...
- Carefully consider the license of
  - Ex: File-system, network "stacks", audio decoders, etc.

15-03-16

23

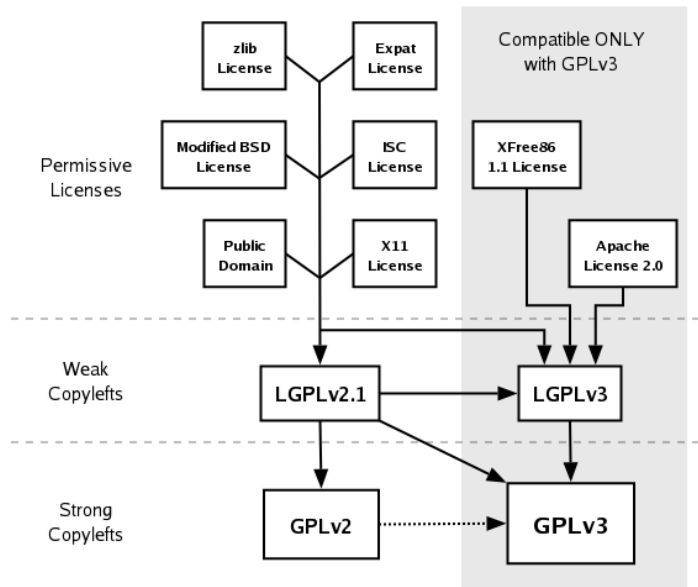
## License models

- GNU General Public License (GPL)  
"Reciprocal" license, "copyleft", "Viral open source"
  - If your program includes any GPL code, then..
- GNU Lesser General Public License (LGPL)
  - If you statically-link to LGPL code, it too must be LGPL
  - If you dynamically link to the code (like a DLL), it need not be LGPL (could have any licence).
- Berkley Standard Distribution (BSD) License
  - Non-reciprocal license...
  - Code may be included in proprietary systems that are sold for profit (closed-source).

15-03-16

24

# Copyleft Licences



15-03-16

25

# Summary

- Use observer pattern to decouple views from data.
- Consider possible reuse of existing software: components, services or complete systems.
- Use configuration management to control system development.
- Open source development allows others to see and change the code
  - Can add complex licensing issues.

15-03-16

26