CS 402: Mobile Development

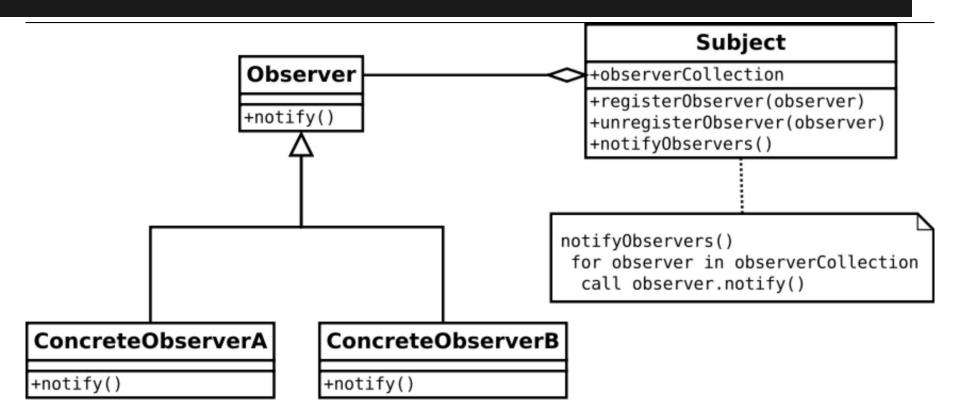
Michael Ziray - michaelziray@boisestate.edu

Observer Design Pattern

Observer Pattern

The **observer pattern** is a software design pattern in which an object, called the subject, maintains a list of its dependents, called observers, and notifies them automatically of any state changes, usually by calling one of their methods.

Observer UML



Observer Libraries

<u>LocalBroadcastManager</u>

Otto by Square Up

EventBus by GreenRobot

Otto BusProvider

```
public final class BusProvider {
    private static final Bus BUS = new Bus();
    public static Bus getInstance() {
        return BUS;
    private BusProvider() {
        // No instances.
```

Otto Registering

```
@Override
protected void onResume() {
   super.onResume();
   setUpMapIfNeeded();
   BusProvider.getInstance().register(this);
}
```

Otto Broadcast

```
// Broadcast that a pin was added
BusProvider.getInstance().post(new AddLocationEvent());
```

AddLocationEvent Class

```
public class AddLocationEvent{
  // That's it!
  // ... or you can add fields to pass data
}
```

AddLocationEvent Class

```
public class AddLocationEvent{
  public Boolean isSuccess;
  public String errorMessage;
}
```

Otto Receiving Events

```
@Subscribe
public void locationsUpdated( AddLocationEvent addLocationEvent )
{
    // Retrieve locations from LocationsController
    // ... or, if they're in the addLocationEvent, pull them out
}
```