CS 314: Project 3

Caleb Tebbe & Zach Kaplan

April 22nd, 2013

E131

Generally, true adapters are used to link a preexisting class whose methods have the same function but different names of a class you are using. Our adapters in the OCSF framework differ in that they are already specifically built to adapt to OCSF classes so they have same names and parameters where true adapters would have to deal with the messy differences between the classes they are adapting to.

E132

Observable Layer advantages:

- Observable layer provides concrete implementations of the Abstract classes to be easily inherited and implemented.
- All necessary service methods in Observable layer simply delegate up to Adaptable layer.
- Developers need to know very little about Observable layer making it simple to tack on different application subsystems.

Observable Layer disadvantages:

- Could make debugging more difficult. In the observer design pattern the flow of control is between observer(s) and observable. In the case of OCSF, if different observers are linked it could be difficult to determine where the problem originated.
- Could add unneeded obfuscation if there is no need for multiple observers, or no plans to expand upon the application.

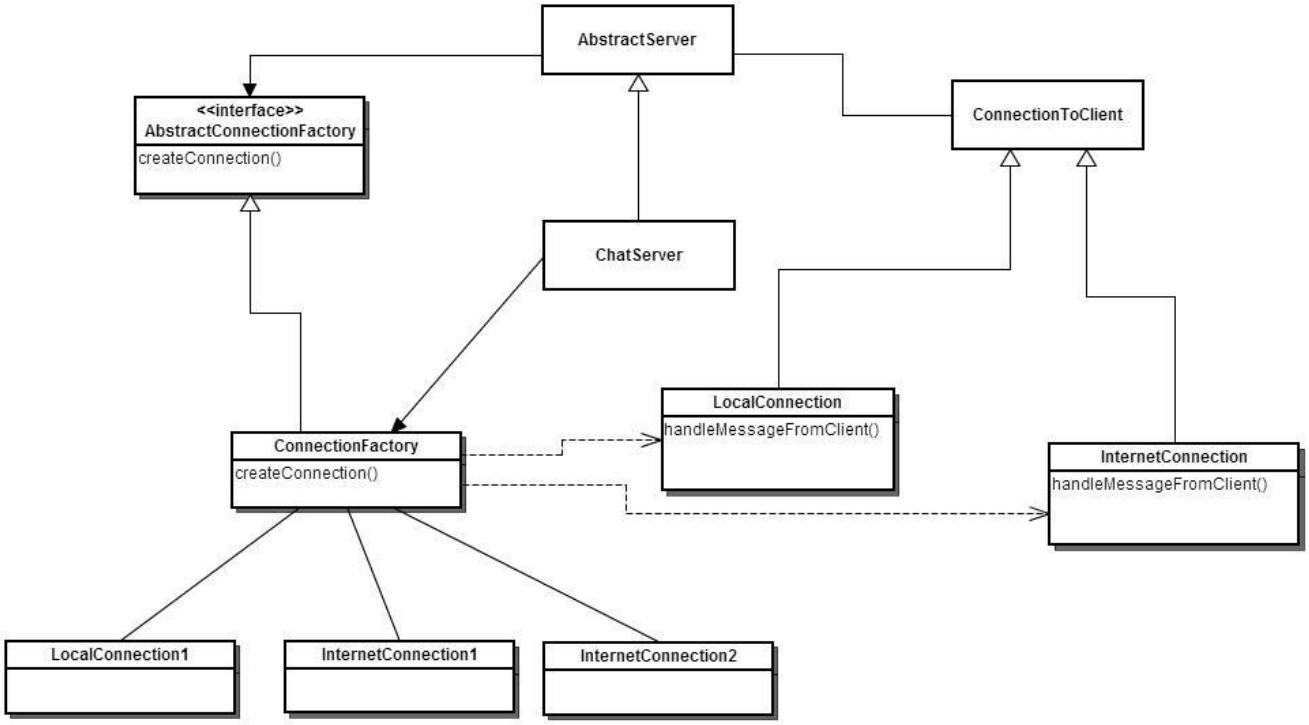
Creating subclass of ConnectionToClient advantages:

- One may not want to have all message handling processed from the server; instead it could be handled by a ConnectionToClient subclass and still be synchronized.
- May want to have different handleMessageFromClient methods in different subclasses of ConnectionToClient to handle different types of connections (such as local or Internet connections).

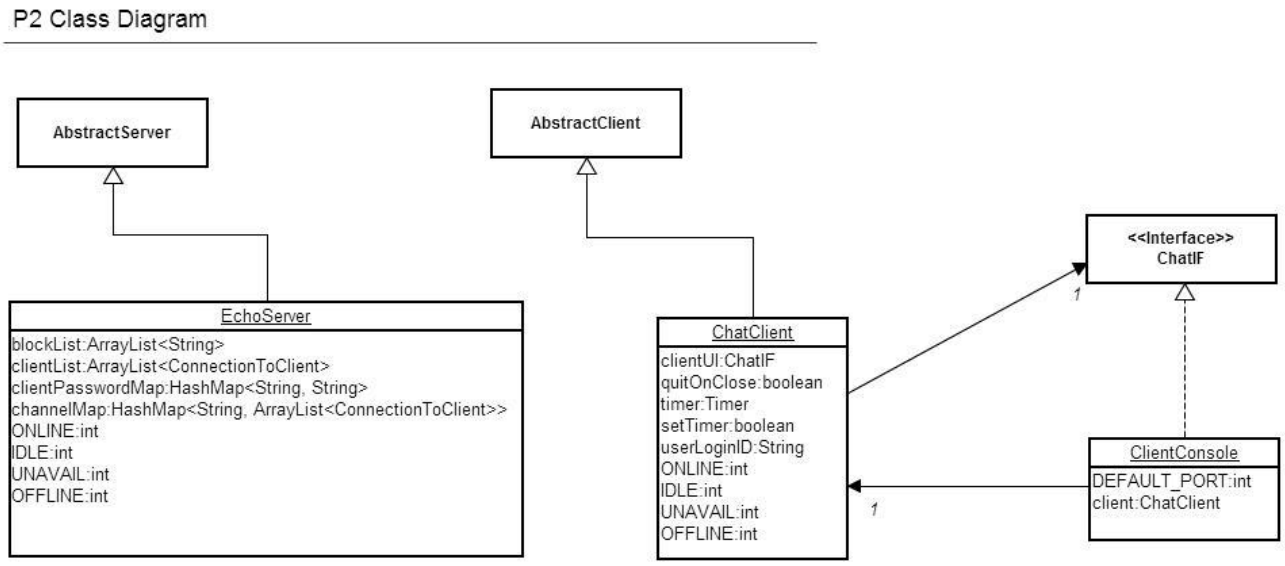
Creating subclass of ConnectionToClient disadvantages:

- Unnecessary if server can handle processing of messages (small scale program)
- May not have need for different handleMessageFromClients. This may be due to there being only one connection type.

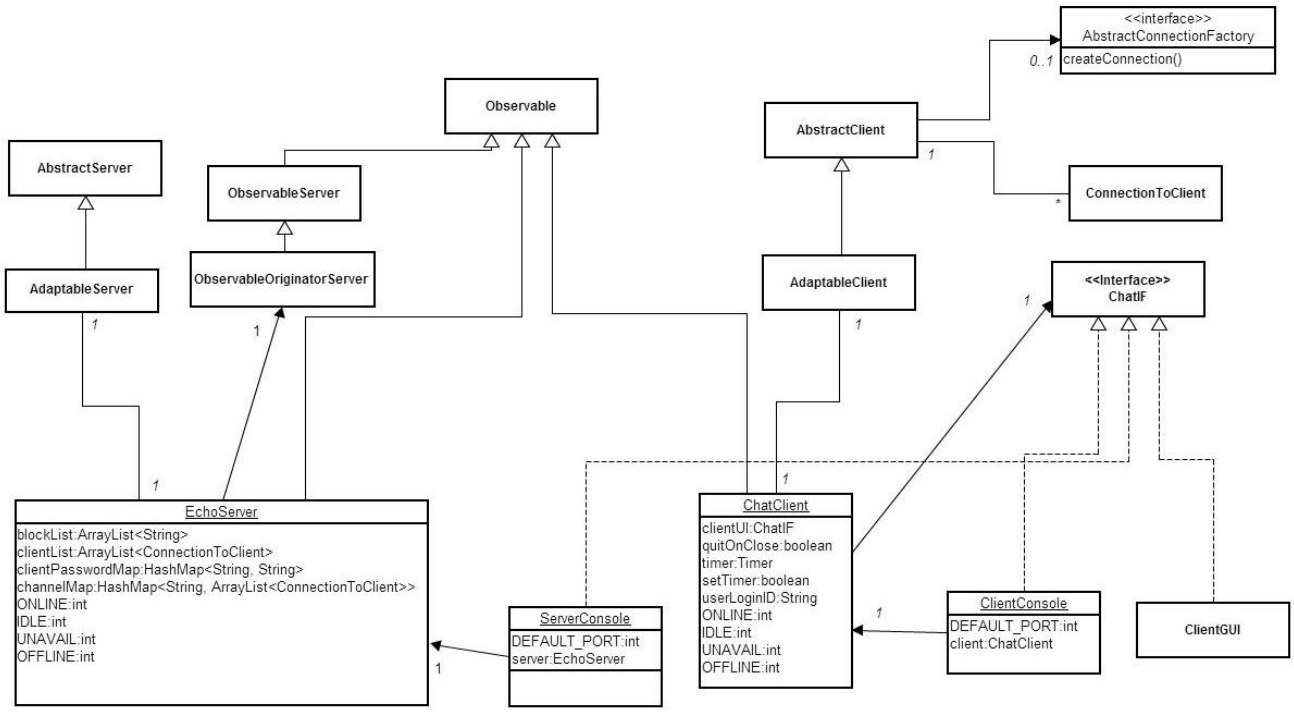
E133



P2 UML Class Diagram



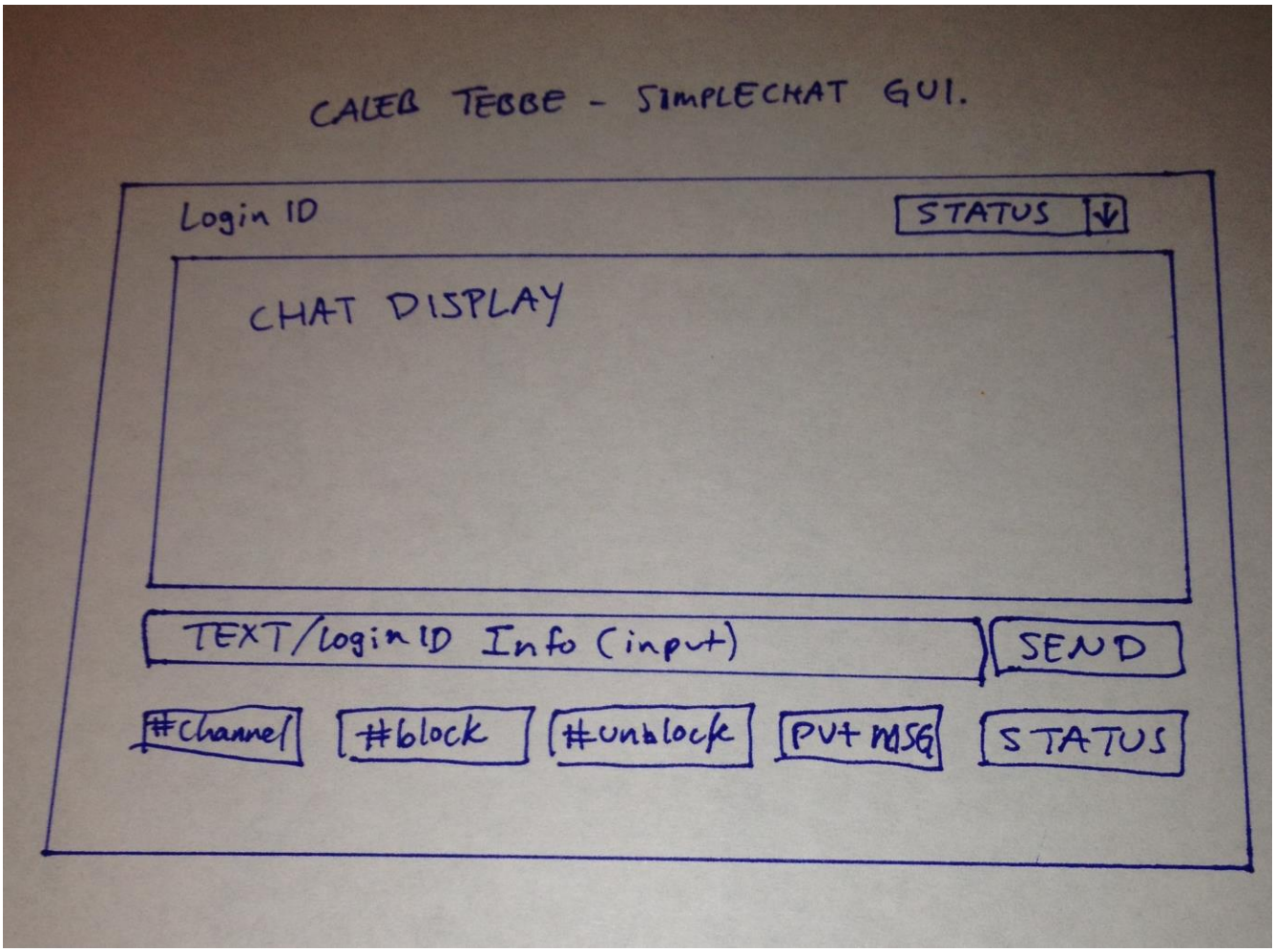
P3 UML Class Diagram



P2 and P3 differences

- Server side
  - changed EchoServer to implement Observer
  - changed EchoServer constructor to accept a UI and ObservableOriginatorServer.
  - Added observer in constructor
  - changed and inherited methods to use ObservableOriginatorServer (sendToAllClients).
  - Moved all stop, start, port/host functionality to EchoServer to be used by ObservableOriginatorServer.
  - Added update method from Observer.
  - added ChatIF interface
  - removed server necessary code from ServerConsole
  - added ObservableOriginatorServer creation to ServerConsole constructor.
- Client side
  - made ChatClient implement Observer.
  - ChatClient constructor adds ObservableClient to be updated.
  - Made all client to server commands use the ObservableClient reference.
  - Added update method
  - Added the creation of ObservableClient to ClientConsole constructor to be passed to the ChatClient

E148


CALEB TEBBE - SIMPLECHAT GUI.
Login ID
STATUS
CHAT DISPLAY
TEXT/login ID Info (input)
SEND
#Channel
#block
#unblock
PVT MSG
STATUS