

Advanced Software Engineering

CSE870, Spring 2007

Miniproject

Due: April 26, 2007

The purpose of this assignment is to give you hands-on experience with building whitebox, graybox, and blackbox object-oriented frameworks. For this assignment, you are to work in pairs to develop a framework for a *family* of network protocols. We use the term *client* to refer to a developer who will use a framework to build an application. Most frameworks provide ways for clients to use both built-in and modified (or newly defined) functionality. Typically, clients use a framework's built-in functionality by instantiating classes and calling their member functions (in adherence to interfaces). Clients extend or modify a framework's functionality by deriving new classes and overriding member functions.

The clients for your framework are developers of protocols for systems such as MISYS. Your goal is to simplify and facilitate a client's use of a framework when building an application. Clients should be able to define their own functionalities that correspond to their specific *protocol functionality*.

Hint: You may wish to use a visitor pattern in the design of your framework that will allow you to define a new operation without changing the classes. The operations provided to the user might change over time; for example, new security requirements might need to be satisfied.

In general, it should be very clear in the documentation of the framework how a client is to use the framework that you construct, including what subclasses can/should be introduced or what the definitions of the interfaces are. Moreover, you must point out to the client the potential pitfalls of your framework if clients adopt particular design decisions. For example, if there exist constraints on the usage of parts of the framework then you, as the framework builder, must identify those constraints.

Assignment

Select 1 *cluster of protocols* (e.g., 3 protocols that perform similar functions, such as 3 file transfer protocols, FTP, SFTP, TFTP) from the list of protocols provided below (total of 9 clusters of protocols). Please note that we have only provided suggested families of protocols. If you discover other variations for a given family of protocols, please contact the instructor and the TA for approval for use. Then

1. Whitebox Framework:

- a. **Develop a Whitebox Framework:** Study all 3 protocols to look for commonalities and differences. Commonalities should be captured in the base classes of your framework. Differences should be achievable by extending the base classes. Your framework should include the use of at least one design

pattern. Develop a class diagram by incorporating relevant design patterns to depict the framework hierarchy.

- b. **Instantiate Whitebox Framework:** Implement all 3 protocols using your whitebox framework.. As you implement each protocol, you may need to refactor to make the framework more general. The final whitebox framework must support all 3 protocols.

2. Graybox Framework:

- a. **Develop a Graybox Framework:** Generalize the whitebox framework to use a combination of composition (use through interfaces) and extend by inheritance to use the framework. As before, use a class diagram to represent the design of your framework. Indicate clearly the extension points and the interfaces to be used during instantiation.
- b. **Instantiate Graybox Framework:** Implement all 3 protocols using your whitebox framework.. As you implement each protocol, you may need to refactor to make the framework more general. The final whitebox framework must support all 3 protocols.

3. Blackbox Framework:

- a. **Develop a Blackbox Framework. :** Generalize the graybox framework to support the use of the framework only through composition (use through interfaces. As before, use a class diagram to represent the design of your framework. Indicate clearly the extension points and the interfaces to be used during instantiation.
- b. **Instantiate Graybox Framework:** Implement all 3 protocols using your whitebox framework.. As you implement each protocol, you may need to refactor to make the framework more general. The final whitebox framework must support all 3 protocols.

4. Develop a state diagram for each class that corresponds to a protocol (total of 9 state diagrams).

5. **Extra credit:** You will receive extra credit if you use any of the three frameworks developed above to support functionality in **MISYS** in appropriate and reasonable places. Augment the class diagram appropriately. And submit the new implementation. Note that all the protocol types have application in **MISYS**.

Assignment Deliverables

The deliverables for this assignment include the following:

1. The name of the projects you choose in the development of your framework.
2. Class Diagrams of the three frameworks. If you attempt the extra credit, then you must also submit the new class diagram and new implementation, highlighting the derived classes of **MISYS** system. (Framework classes and derived classes should be explicitly identified).
3. State diagram of classes of corresponding protocols.
4. All files for testing the framework classes and the final application programs. These files will include the MAKEFILE for compiling and running your program. You will also include a README file describing how to use your final application programs.
5. Project report.

Intermediate Submissions

You are to report on the progress of your modeling and development gradually. In particular, you need to meet the following deadlines:

1. (**April 3rd**) Report on design and implementation of the whitebox framework. This report must include (1) the class and state diagram, including derived classes to develop the selected protocols, (2) description of interfaces that implement the common functionality of three protocols, and (3) source code of your implementation. Your report will be submitted in class and your source code should be sent to borzoo@cse.msu.edu. Make sure you include prose guidelines for how to run your code.
2. (**April 12th**) Design and implementation of graybox framework. Your report must include the same items mentioned above. Make sure you include the class diagram of your whitebox framework as well.

Final Submission

(**April 24th, 26th**) In-class presentations of projects.

(**April 26th**) Your final project report will include the following (in the order specified):

1. Name of your group members.
2. Table of Contents.
3. Class diagram of frameworks (respectively, **MISys** in case you attempt the extra credit), including derived classes to develop the selected protocols and data dictionary.

Note: Data dictionary should be easy to peruse, examples of how you can make data dictionary easy to perused are alphabetically ordering and extra table of content for data dictionary.

1. A description of your solution for the framework. You should describe the problems faced in keeping the client interaction at a minimum and your solutions. The report should also explain how (the specific process and steps within the process) a developer would use the framework (for developing protocols, for example).
 2. Any assumptions you made about the interpretation of specifications that might have been unclear.
4. 3 state diagrams corresponding to each protocol.

Grading

If you only develop a whitebox framework and instantiate to implement all 3 protocols, you will be eligible for 70% of the total grade, 80% if you also implement and instantiate a graybox framework, and 100% if you also implement and instantiate a blackbox framework. Note that the penalty of having a too-specific framework can be greater than 10% of the total grade. Moreover, each framework will be graded based on the following criteria:

1. **(20%)** Class diagram and state diagram.
2. Project Report
 - a. **(10%)** Description of framework.
 - b. Include descriptions of the original applications that you used to build the framework (e.g., descriptions of their similarities and differences between the 3 protocols of each type, etc.)
3. Evaluation of Framework
 - a. **(40%)** Framework
 - b. **(15%)** Instantiation
4. **(15%)** Commenting of Code

(15 pts) Extra credit

		<i>Protocol</i>	<i>Why needed?</i>	<i>Resource</i>
1	Transaction	1-phase commit	Several participants issue a transaction on the same database record.	http://citeseer.ist.psu.edu/83540.html
		2-phase commit		http://en.wikipedia.org/wiki/Two-phase_commit_protocol
		3-phase commit		http://en.wikipedia.org/wiki/Three-phase_commit_protocol
2	Email Transfer	IMAP2	To retrieve e-mail messages from a remote server over a TCP/IP connection (pull protocol)	http://tools.ietf.org/html/rfc1176
		POP3		
		IMAP4rev1		http://tools.ietf.org/html/rfc3501
3	Authentication	Kerberos	Strong authentication for client/server applications	http://web.mit.edu/Kerberos/
		Otway-Rees protocol	Authentication protocol designed for use on insecure networks.	http://en.wikipedia.org/wiki/Otway-Rees
		Needham-Schroeder protocol	Mutual authentication between two parties communicating on a network	http://en.wikipedia.org/wiki/Needham-Schroeder
4	Authorization	DIAMETER	Protecting computer resources by only allowing those resources to be used by resource consumers that have been granted authority to use them.	http://en.wikipedia.org/wiki/DIAMETER
		RADIUS		http://tools.ietf.org/html/rfc2866
		TACACS+		http://tools.ietf.org/html/draft-grant-tacacs-02
5	Accounting	RADIUS	Tracking the consumption of network resources by users.	http://tools.ietf.org/html/rfc2866
		TACACS+	Provides access control for routers and other networked computing devices via one or more centralized servers.	http://tools.ietf.org/html/draft-grant-tacacs-02
		DCCA	Provides real-time credit Control which is not provided in basic RADIUS	http://tools.ietf.org/html/rfc4006
6	File Transfer	TFTP	Simple file transfer	http://www.isi.edu/in-notes/ien/ien133.txt
		SFTP	Secure file transfer.	http://rfc.net/rfc913.html
		FTP	Non-secure file transfer	http://www.ietf.org/rfc/rfc0959.txt

7	Remote Access	RLogin	Provides remote access to UNIX machines	http://en.wikipedia.org/wiki/RLogin
		Telnet	General purpose remote access protocol	http://tools.ietf.org/html/rfc854
		SSH	Secure shell	http://tools.ietf.org/html/rfc4251
8	Hypertext Transfer	HTTP	Web-based remote access.	http://www.ietf.org/rfc/rfc2616.txt
		HTTPS		http://tools.ietf.org/html/rfc2818
		SOAP		http://en.wikipedia.org/wiki/Simple_Object_Access_Protocol
9	Proxy	SOCKS4	Clients behind a firewall, needing to access exterior servers	http://en.wikipedia.org/wiki/SOCKS#SOCKS_5_protocol
		SOCKS4a		
		SOCKS5		