***Title      Homework 4: Automated Teller Machine***

***Author     Kalpan Bhatt***

***Date         11/12/2012***

***Overview***

**Purpose**

Develop the back end of a distributed Automated Teller Machine (ATM) system using basic Java networking.

**Brief description**

We are developing a distributed Automated Teller Machine (ATM). The ATM will be hosted in a different process than the client application. The client will connect to the server using Java networking and communicate with the server using a protocol.

**Functional Requirements**

The ATM will simulate a real world automated teller machine.

The ATM must support the following operations:

* deposit: add some dollar amount to account balance
* withdrawal: deduct some dollar amount from account balance
* balance inquiry: get current account balance

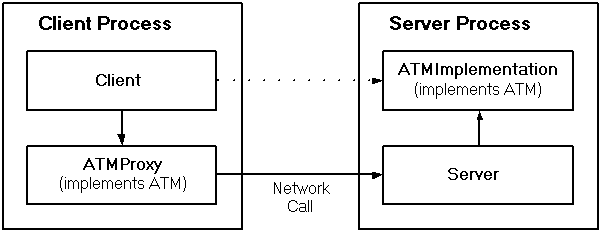
The ATM will run in its own process and will handle remote requests from a client over a socket connection running in some other process.

**Design**

In the ATM system, the client and ATM will be running in different processes. Since the client cannot reference memory in the server process, it cannot get an actual reference to the real ATM object running on the server. Instead the client will use a proxy that presents all the behavior that characterizes an ATM. In fact, the client doesn't even need to know if the actual ATM is local or remote. This is achieved using a Java interface.

**Architecture**

The following diagram illustrates how the client operates on an ATM which is actually an ATMProxy that communicates over the network to the server and dispatches the call to the ATMImplementation.



**Goals**

1. All classes should be in package cscie.160
2. Create an ATM interface
3. Create a ATMImplementation class that implements ATM interface. This will have information about only one account for now.
4. Create an Account class. It will hold information about its balance.
5. Create an ATMProxy class. It will also implement ATM interface, but it will be invoked at the client end.
6. Create a server and Client class.
7. Client class should have main method for the demonstration.

**See Also**

[**http://courses.dce.harvard.edu/~cscie160/hw4-06.htm**](http://courses.dce.harvard.edu/~cscie160/hw4-06.htm)

**Assumptions**

1. *Currently the system supports only one Account.*

**Risks**

NA

**Current procedure/functionality**

1. Server process will listen for connections from clients on port 1099.
2. When a client will be started on the same host and sends a request on port 1099, the request will be received by ATMProxy class.
3. ATMProxy has implementation of public interface ATM. The same interface is also implemented by server side ATMImplementation class.
4. ATMProxy will call methods of the ATMImplementation.
5. ATMImplementation will have methods to get information about Account balance.
6. There are currently 3 methods implemented to Deposit, Withdraw and QueryBalance.

**How to run the assignment**

*Unzip the contents of the jar file HW4.jar*

*Run the Ant tool at the folder level of build.xml*

*Use the jar file created in build folder to run server and client files.*