COMP 6231
Aymeric Grail
#26810020
aymeric.grail@gmail.com

2nd assignment: Documentation

Loan Management System - CORBA

General notes

This documentation is related to the code submitted electronically for the 2nd Assignment (via the EAS system).
The code can be found in the same zip file as this documentation electronic version.
1 hard copy has been produced.

Note that this documentation is only complementing the 1st assignment documentation. It will only highlight the new decisions and challenges.
For more details, please refer to the 1st assignment documentation (an electronic copy is provided in the documentation folder of the archive).

The **java Documentation** has been redacted and is available electronically in the document folder at the root of the sources folder.
No hard copy has been produced.

Note that the project has been developed and tested using IntelliJ IDE.
Corba files (stubs, skeletons, …) have been generated using eclipse (then imported to IntelliJ).
Installing the "MultiRun" plugin on Intellij IDE allows to run the multirun configurations provided in order to start all 3 instances of the server at once.
Folder Root of the project: 6231-02-Loan-Corba

Challenges

In general, challenges encountered during this assignment were simpler than in previous assignment. Mostly due to the fact that most of the strategies for decoupling (interfaces), concurrency (lock factory) and messaging (UDP) were already in place.

Nevertheless, here are some of the challenges encountered during this assignment.

CORBA:

Certainly, most of the challenges came from the CORBA middleware implementation as a matter of fact.

- Learning the details of the Interface Definition Language. How to define complex types, arrays, or throw exceptions
- The plugin to generate the CORBA code for helpers, stubs, skeletons etc. is not very verbose when failing. Neither does it have support from the community (last update was in 2004)
- Managing generated code is tedious. Especially when refactoring the IDL interface. Need to regenerate everything and re-import it in the project, so forth and so on.
- Some refactoring was needed in order to adapt the solution to a new middleware. The choice has been made to not support both middleware at the moment but a simple strategy pattern could be implemented at the client's service layer to support it.
- Mapping objects is tedious. In order to keep the middleware layer decoupled from the rest of the application, all entities generated for CORBA have been mapped to their equivalent Data Type at the application level.
  This mapping was not complicated but it was tedious.

UDP:

- concurrency was handled for data access but not for UDP. Therefore, only when I started testing concurrent Loan Transfer did I realize that UDP answers ended up being mixed up. Simply reused the Lock Factory to handle it.
- the udp client only supported one type of message prior to this assignment. Needed to implement a strategy to support different types of operations.

Testing:

- prior to this assignment, tests were situated in the server class and were directly performed when starting the server.
  This has been corrected so that test can now be run separately.
  NOTE: servers have to be running before running the test class.

Testing

In order to run the tests, start all 3 instances of the bank server then run the "Tests.BankService.BankServiceTest" class.

All 4 prior operations were already covered by unit tests. Those tests helped a lot while refactoring the solution to support CORBA instead of Java RMI, ensuring nothing was

broken on server side.

The only test that has been added for this assignment is the testTransferMultipleLoans() method that covers the Tranfer Loan functionality.
This tests the function in a concurrent manner, including cross UDP communication between different banks and clients in order to maximize the concurrency coverage. It's only when I started crossing messaging that I discovered (and then fixed) UDP concurrency errors.

Source Control

GitHub has been used as a source control. The github repository is publicly accessible at the following address:
https://github.com/erkkiperkele/6231-02-Loan-Corba

Process documentation

The process for this assignment was similar to the one of the prior assignment. The main difference was at the beginning of this assignment. Instead of sketching the high level architecture of a new application, I had to plan the minor but needed refactoring in order to be able to properly adapt and extend the application to a new middleware and feature.