4TH EXERCISE ON "ENTERPRISE APPLICATION INTEGRATION"

in summer term 2013
Prof. Dr. Herbert Kuchen

Due: 12.07.2013, $23^{\underline{59}}$ 28.06.2013

Message Oriented Middleware (50 points)

Scenario

Using the statistics provided by the head office system, JAVA and C.SHARPE could not yield insights on improving replenishment cycle times. As part of a retrospective that involved all their employees, they found out that ordering is often forgotten as day-to-day business at pharmacies can often be busy. Therefore, JAVA and C.SHARPE decide to have clerks at the head office take care of timely ordering. To guarantee data consistency between their head office and subsidiaries, they want you to utilise Message Oriented Middleware.

Use Cases

You are commissioned to extend the existing systems. The following use cases need to be covered:

- Creation and maintenance of replenishment orders at subsidiary also creates and updates its equivalent at the head office
- The head office may exclusively...
  - Transition to *posting*, *cancelled*, and *ordered*
  - Enter the expected delivery date before transitioning to *ordered*
- The entry of the actual delivery date and transition to *finished* should be done at the respective subsidiary only
- Replenishment orders are shared between subsidiary and head office
  - JaVa ↔ head office
  - C.Sharpe ↔ head office

Organisational Issues

The tutorial on JMS and .NET and the accompanying example explain how to connect a .NET application with a JMS provider. As described in the tutorial, you will need the modified DLLs of the Apache.NMS project.

Existing orders do not need to be migrated. Jackie and Cabe gave their OK to purge existing orders from their systems. Furthermore, they do not want you to convert the delivery deviation statistics to use the now local data: even though you tried talking them into it, they fear it would cost them too much.

Please supply your queues or topics through a hornetq-jms.xml file as presented in the example. Your group number has to be part of their names.

The example uses the `@Resource` annotation with the parameter `lookup` to let the container inject certain JMS objects. In order to do the same in your projects, you have to change the order of the build path so that the JBoss runtime stands in front of the JRE library. In Eclipse, right-click on the respective project, choose *Build Path >Configure Build Path ...* and arrange the items on the *Order and Export* tab accordingly.