IN2013, Week 7 – Components and Deployment

Dr Peter Popov

Model Answers

8th November 2018

Scenario

Consider that the BAPERS system is deployed as a web application as follows:

- A single node called BAPERS server (modeled as a device node) is used as hardware with operating system Linux Fedora Core 10 (modeled as execution environment);
- BAPERS itself is developed as a web application which may consist of many servlets and Java Server Pages (JSP), which are packaged together as a single archive file, BAPERS.war (an artifact), as required by the J2EE specification for web application. BAPERS.war is then deployed on BAPERS server, in Tomcat to be more precise.
- A DB server (DB server device) is also used for the deployment with operating system Linux Fedora Core 10. The database deployed on DB server is MySQL 5.1. The database file itself (where the databases tables are stored) is called bapers.dbi and is modeled as an artifact deployed on MySQL.
- The communication between Tomcat running the web-application and the database is via a JDBC driver (another artifact, e.g. mysql.jar, deployed also in Tomcat 6.0)
- The web application is accessed by a browser of any device and operating system communicating with Tomcat. For the purpose of this exercise, assume that there is an Employee PC (as a device) with operating system Windows 8 (as execution environment) in which a browser, e.g. Firefox, is run (another execution environment).
- The communication protocols used in the deployed system are as follows:
  o tcp/ip between the devices
  o http between a browser and Tomcat (Tomcat listens on port 8080)
  o 'mysql protocol' between the execution environments Tomcat and MySQL (MySQL listens on port 3306).

Question 1.

Develop a deployment diagram for BAPERS.

Model answer

A possible solution is presented below.
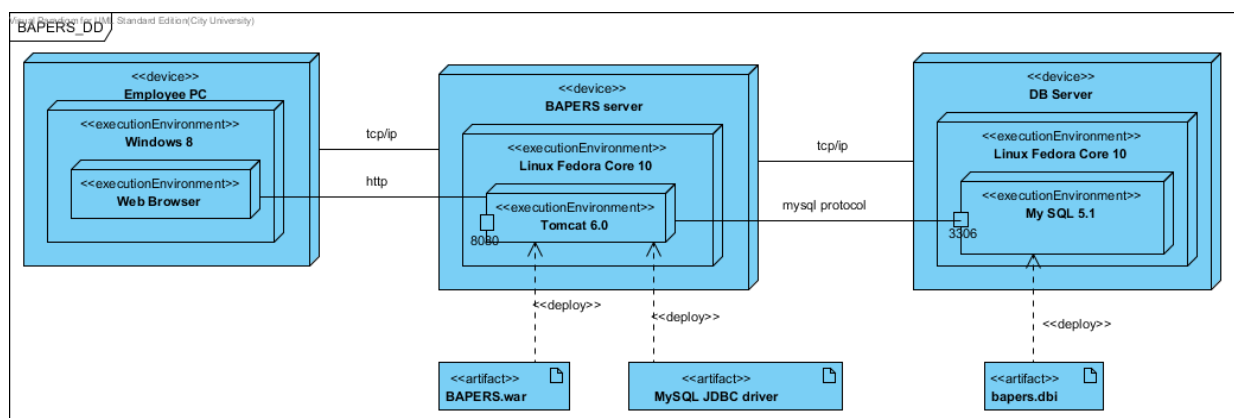
Note that the devices may contain nested execution environments: the operating system is nested in the devices (which represent the hardware).

Tomcat is an execution environment in which one can run servlets and Java Server pages (JSP), i.e. java code on a server (in a similar way as php is run on web server).

MySQL is also modeled as an execution environment as it offers execution of SQL queries and even more advanced processing such as store procedures and triggers which are important for enforcing referential integrity between the tables in the database (primary/secondary keys).

The web application BAPERS.war is an archive (similar to zip and jar archives) which has a specific structure. This file, the databases file, bapers.dbi, and the JDBC driver are modeled as artifacts and are deployed in their respective environments – Tomcat and MySQL, respectively.

The diagram also shows the communication protocols between the devices and between the execution environments. The devices communicate via tcp/ip; the browsers and Tomcat use http; the access to MySQL (from Tomcat is via mysql protocol (which is implemented by the respective JDBC driver).

Finally, I added the ports used in the communication with Tomcat and MySQL. These details are not needed, but have been added to illustrate that the deployment diagrams allow for showing such details. 8080 is the default port on which Tomcat listens; 3306 is the default port on which a MySQL server listens for SQL queries.

Question 2.

Consider that BAPERS is accessed from a browser using a javascript library such as jQuery. Model the deployment of jQuery (as an artifact) in the browser.

Model answer

The modification in comparison with the solution to Q1 is minor: now the jQuery.js is added as an artifact deployed in the web browser: this could be any javascript enabled browser including browsers run in mobile devices.
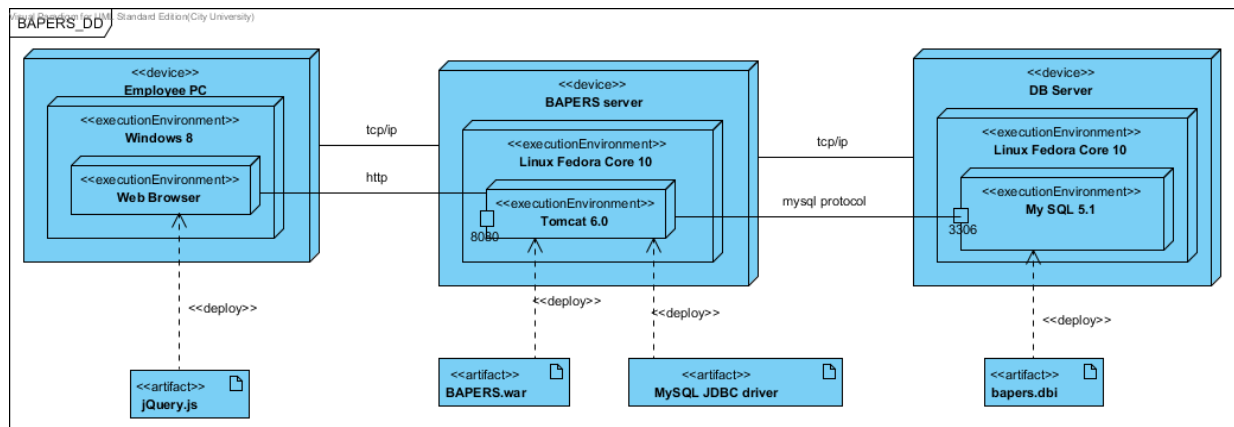
Question 3.

Extend the diagram to model that BAPERS can be accessed from mobile devices.

Model answer

Not provided, but would include adding mobile devices similar to Employee PC with their OS, e.g. Android, etc. The rest of the deployment diagram will remain unaffected.

Question 4.

Consider a possible layered architecture of BAPPERS, which assumes that a specific RDBMS will be used, e.g. MySQL.
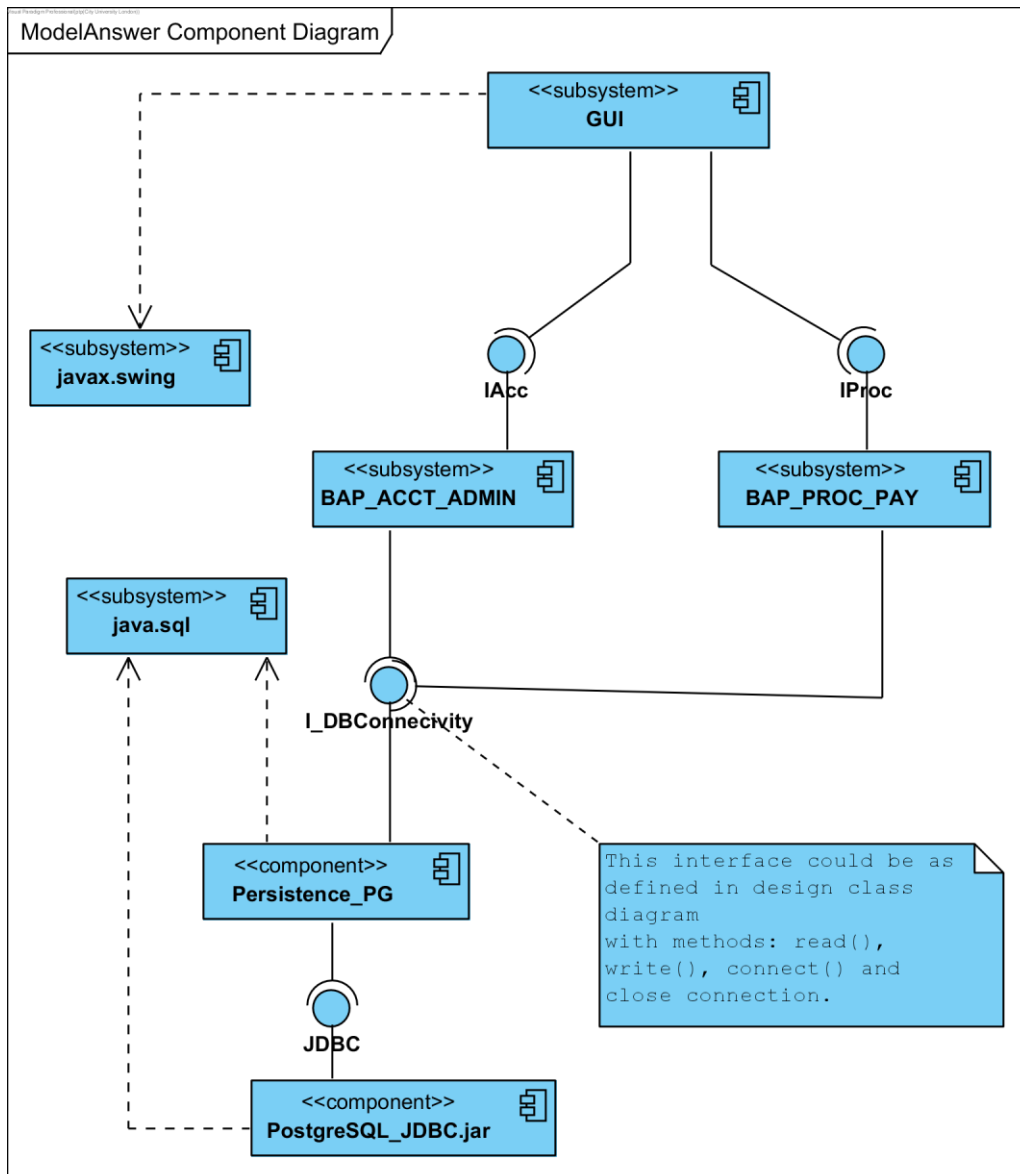
A possible component diagram is shown below.

Implementation Model BAPPERS Component Diagram
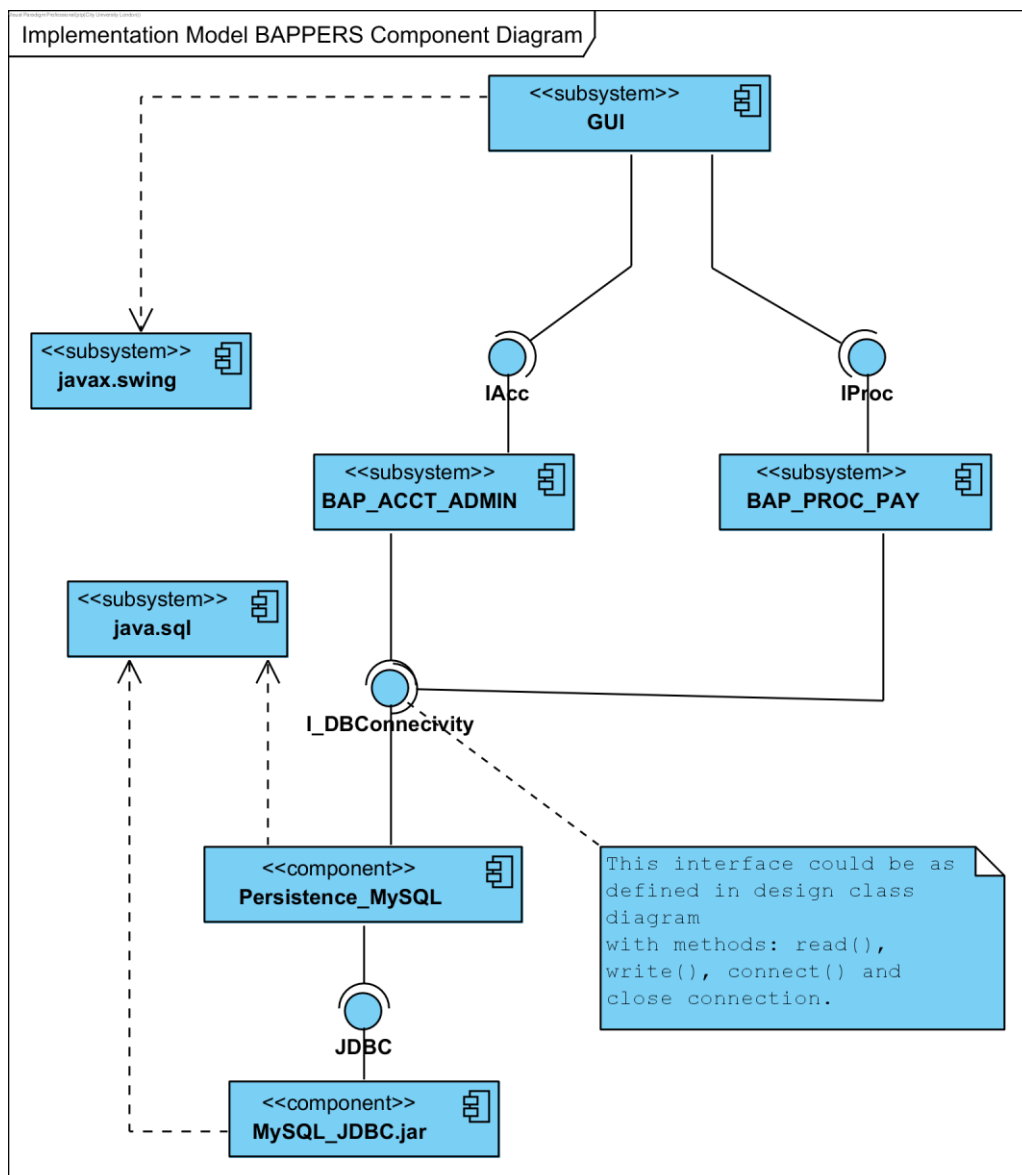
Now consider a possible change of the RDBMS from MySQL to a different product, e.g. PostgreSQL.

a) How will the BAPPERS architecture change? Draw a component diagram to reflect the change.
b) Discuss the implications of the change from MySQL to a different RDBMS for the application code. Will it be affected by the change? If so, discuss ways of minimizing the changes.

Model answer

a) *Clearly, the JDBC driver will have to change, so that BAPPERS can communicate with PostgreSQL. The architecture then will be as shown in the diagrams below:*

ModelAnswer Component Diagram

<<subsystem>>
GUI

<<subsystem>>
javax.swing

IAcc

IProc

<<subsystem>>
BAP_ACCT_ADMIN

<<subsystem>>
BAP_PROC_PAY

<<subsystem>>
java.sql

I_DBConnecivity

<<component>>
Persistence_PG

```
This interface could be as
defined in design class
diagram
with methods: read(),
write(), connect() and
close connection.
```

JDBC

<<component>>
PostgreSQL_JDBC.jar

Note that in this diagram I changed two of the components: the JDBC driver itself (PostgreSQL_JDBC.jar), but also Persisitence_PG component. The latter will contain the implementation of I_DBConnectivity interface.

b) *Whether this component will be different from the component Persistence, used in the assignment, will depend on how the persistence component has been implemented. If Persistence component relied on features of MySQL which are not present in PostgreSQL, then Persistence_PG would be different from persisitence. Examples of such difference will be "show tables" statement, which is supported by MySQL, but does not have a direct counterpart in PostgreSQL. For further details, check:*
http://www.linuxscrew.com/2009/07/03/postgresql-show-tables-show-databases-show-columns/
There are many other examples of "proprietary extensions" of SQL which vendors of different RDBMS have added to their products. Relying on such extensions may be convenient but creates difficulties for porting an application to a different RDBMS. This is a very well-known problem. Designing an application for "portability" to a different RDBMS could be achieved by either using a minimal subset of SQL (e.g. SQL – 92 entry level), which is widely supported by many RDBMS-s, or using object relational mapping frameworks (e.g. Hibernate discussed at the lecture).

If a minimal subset of SQL has been used in the application with MySQL, then it may be possible to retain Persistent unchanged.

4th November 2018