

ODBMS.ORG User Report, No. 26/08

Editor: Roberto V. Zicari- www.odbms.org

November 2008

Category: **Industry**

Domain: **Finance**

User Name: **Serena Pizzi**

Title: **Responsible Application Management Back End**

Organization: **Banca Fideuram SpA, Italy.**

Q1. Please explain briefly what are your application domains and your role in the enterprise.

Serena Pizzi is responsible for Application Management Back End systems of Bank Fideuram (in cooperation with SOFTLAB).

Q2. When the data models used to persistently store data (whether file systems or database management systems) and the data models used to write programs against the data (C++, Smalltalk, Visual Basic, Java, C#) are different, this is referred to as the "impedance mismatch" problem. Do you have an "impedance mismatch" problem?

At present, in our division, ADP/AGP, we have an “impedance mismatch” problem: this is due to the use of JDBC to access data (using relational databases: DB2 and Oracle), and to the fact that we did not use so far any persistent framework such as ORM. We have found hard when mapping an object conceptual model to the relational model. We solved this by manually programming the mapping ourselves.

Q3. What solution(s) do you use for storing and managing persistence objects? What experience do you have in using the various options available for persistence for new projects? What are the lessons learned in using such solution(s)?

We have a considerable experience in using framework for object persistence such as Hibernate.

In particular, for Hibernate, we have:

- PRO : Using Hibernate we obtained a considerable simplification to the data access layer, making it almost transparent, and reducing data access to small/medium query of limited complexity.
-
- CONS : We have found some difficulties in mapping objects to large relational databases.

Q4. Do you believe that Object Database systems are a suitable solution to the "object persistence" problem? If yes why? If not, why?

Surely ODBMS are a better solution for object persistence. However, we should consider the following: ODBMS are object oriented and therefore improve the standard of development and quality of the code (control of object persistence at compile time, data type safety, etc.). However, this focus is also a limitation, especially when they are used in domain where "objects" are not mainstream (such report generation, data warehouse or datamart, access to heterogeneous data). In this cases, the performance of ODBMS is not comparable with the one of relational databases, and this is also true for the level of knowledge of the consultants who work in this area.

Q5. What would you wish as new research/development in the area of Object Persistence in the next 12-24 months?

We are experimenting internally with a test project with goal to "transform" data access of existing applications from standard JDBC to an ORM (probably using Hibernate).