

## **ODBMS.ORG User Report No. 18/08**

Editor Roberto V. Zicari- ODBMS.ORG [www.odbms.org](http://www.odbms.org)

October 2008.

*Category:* **Industry**

*Domain:* Finance/Banking

*User Name:* **Peter Train**

*Title:* *Architect*

*Organization:* **Standard Bank Group Limited, South Africa.**

After a career change from music, Peter Train has been working in Information Technology since 1994. He has extensive experience with Enterprise Application Integration, software design and programming with C++ and Java. He spent several years convinced that the Object Management Group's Model Driven Architecture initiative was the real "Silver Bullet". After waiting some time for the dream to be delivered, his hopes were finally shattered when he read Umberto Eco's *The Search for the Perfect Language* and Douglas Hofstadter's *Gödel, Escher, Bach*. He is now looking to the Agile Methods community for a more pragmatic approach.

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

**Peter Train:** I have an architectural role within the operations area of the Corporate Investment Banking division of Standard Bank. The application domains within our area include confirmation and settlement, payments, reconciliation, reference data, international trade and custody. I do have some exposure to other areas within the investment banking division and my answers will not necessarily be specific to the operations area.

*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?*

**Peter Train:** Many of the legacy systems in our environment have been developed using a structured programming paradigm and relational databases. I don't think that the "impedance mismatch" problem as stated above applies to these.

Many of our newer systems are off the shelf products and while I know how some of these manage persistence, I won't discuss these.

Some of our newer systems have been developed in-house using an object oriented paradigm.

Most (if not all) of these use Relational Database systems to store data and the "impedance mismatch" problem does apply.

With the exception of Meta Object Facility metadata repositories (which I presume would qualify as Object Database systems), I am not aware of any systems in our environment that have been developed using Object Database systems.

*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)?*

**Peter Train** The most common scenarios that I am aware of being used in our environment are as follows:

- Java Enterprise Java Beans using container managed persistence to manage object to relational mapping
- Java using Hibernate to manage object to relational mapping
- Java using Java Persistence API to manage object to relational mapping
- C# using in-house hand coded layer to manage object to relational mapping

The most common problems that we have experienced with object Relational mapping tools are:

- The effort required to define mappings between the object and the relational models
- Difficulty in understanding how the mapping will be implemented at runtime and how this might impact performance and memory utilization.

In some cases, a great deal of effort is spent tweaking configurations to achieve satisfactory performance.

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

**Peter Train:** - My mind is not made up on this matter.

At the moment, I lean towards the belief that Object Database systems may be appropriate in certain specific cases but not as a general solution to all problems.

I believe that the most significant problems with Object Database systems are as follows:

- For many Object Database systems, the system is tightly coupled to a particular programming language.
- A far more significant issue was alluded to by Grady Booch in *Object Oriented Analysis and Design*: There is no universally correct way to model objects.

Of course the perspective of the designer plays a role in determining how objects are structured. However a more subtle issue is that depending on the role objects play within a particular application domain, considerations for that domain may lead to different ways of structuring the objects.

This issue has been dealt with explicitly by Subject Oriented Programming. This issue tends to become more and more significant as we try to achieve greater degrees of integration between different application domains.

I am sure that this issue applies equally to both object oriented modeling and relational modeling.

Further more, I believe that it is usually preferable to try and avoid using shared data as a way to achieve integration between different applications. However, being able to take a generalized relational model and using object relational mapping tools to provide one or more specific object oriented abstractions of this data often proves useful.

In addition, having the data stored in a relational form that can easily be exploited by our current generation of data warehousing and OLAP tools also often proves useful.

In short, I am not sure that substituting object relational mapping with Object Databases would help us to solve the problem of integration across the enterprise (which is possibly the most significant hurdle we are facing at the moment).

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

**Peter Train:** On the whole, I am quite happy with the state of object relational mapping techniques and tools and I don't have

any particular interest in Object Databases. However, I would be interested in seeing case studies of Object Databases used within the financial industry.

In particular, I would be interested in seeing how they can be exploited within an environment where data is used both to support operational processes and decision support processes.

#