ODBMS.ORG User Report No. 10/08

Editor Roberto V. Zicari- ODBMS.ORG www.odbms.org
July 2008.

Category: **Industry**
Domain: Defense/intelligence area**.**
User Name: **Mike Card**
Title: Principal engineer
Organization: **Syracuse Research Corporation (SRC), USA**

Mike Card works with Syracuse Research Corporation (SRC) and is involved in object databases and their application to challenging problems, including pattern recognition. He chairs the ODBT group in OMG to advance object database standardization.

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

Mike Card: Our application domain is in the defense/intelligence area, my role is as principal engineer in a research and development group.

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?

Mike Card: Presently no, as we are using an object-oriented database

Q3. What solution(s) do you use for storing and managing persistence objects?

Mike Card: We use a home-made Java object database

What experience do you have in using the various options available for persistence for new projects?

Mike Card: I have done evaluation and testing of most major commercial Java persistence solutions, including db4o, Versant, Objectivity, ObjectDB, JPOX, and Hibernate

What are the lessons learned in using such solution(s)?

Mike Card: Many of these products are very easy to use. However, all of them use a form of so-called "transparent activation" which ultimately means that to navigate a graph of connected persistent objects the entire graph must be cached in RAM. For very large data sets, this is not acceptable. Some products give you options to try to fix this problem such as a setting for activation depth, options for manual activation and de-activation of objects etc.
All of these options, however, cause your application to become cluttered up with memory management code that is difficult to maintain and quite prone to error. That is why we ended up building our own object database that uses a different approach for activation, and this allows us to have very large object graphs without requiring that everything fit in RAM and without requiring manual activation or de-activation of objects, setting of activation depths, etc.

Now admittedly we are working in a very specialized application domain, and most users would not have this problem.

The transparent activation scheme used by the products we tested is a great general solution because it maximizes ease-of-use for the developer, it's just that it unfortunately doesn't work well for our particular problem area.

Q4. Do you believe that Object Database systems are a suitable solution to the "object persistence" problem?

Mike Card. yes

If yes why?

Mike Card. because we are currently using one successfully!

If not, why?

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

Mike Card: I would like to see object programming languages build in persistence as a true language concept. As I have said earlier, Ruby comes the closest from the languages I have played with but I think much more could be done in this area.