

Re: Database type independence

Source: <http://coding.derkeiler.com/Archive/General/comp.object/2004-08/1792.html>

From: Topmind (topmind_at_technologist.com)

Date: 08/23/04

Date: 23 Aug 2004 00:59:02 -0700

fredrik_bertilsson@passagen.se (Fredrik Bertilsson) wrote in message news:<31f7e57d.0408220607.5bcbd0a8@posting.google.com>...

- > *Many OO evangelists argue very hard for making the persistence layer*
- > *independent of the database type (RDB, OODB, LDAP, etc). They want to*
- > *be able to change the database from a RDB to an OODB without changing*
- > *the business logic.*
- >
- > *But the cost for doing this is very high. The developer is forced to*
- > *implement (or configure) an extra transformation/mapping layer. The*
- > *persistence frameworks becomes unnecessary complex. This requirement*
- > *about database type independence is one of the main reasons that EJB*
- > *is so very complex. Besides, data-aware components are not possible to*
- > *use because of this requirement.*
- >
- > *My question is, is it really worth the cost to make the architecture*
- > *independent of database type. On the software market, there are a huge*
- > *demand for relational database vendor independent applications. Most*
- > *customers want to be able to run the applications they buy on their*
- > *favorite dbms system like Oracle, SQL Server, MySQL etc. But how many*
- > *customers want the application to be able to run on their OODB? I have*
- > *never heard about it anyway.*

Perhaps sell it the other way around: make the database be programming paradigm independent. Since data tends to last longer than languages, this is goal is at least just as worthy. If OO is the final pennicle of paradigms or Java is the end-all-be-all of languages, I shall eat my bloomers with hot sauce.

- >
- > *Relational databases have proven to be the superior database type.*
- > *They have almost 100% percent market share and this fact does not seem*
- > *to change in the near future. In fact relational databases is more*
- > *likely to live longer than most programming languages used today*
- > *(Java, C#, C++, PHP, VB etc). There have been numerous examples of*
- > *applications changing programming languages (from Cobol and Fortran to*
- > *Java or VB/C#) and the database remains the same. But the examples of*
- > *changing database type and keeping the application the same is*

> *extremely few.*

>

> *Is it really reasonable to invest money in a feature (database type
> independence) that in almost no cases are needed? I think that if IT
> managers really knew that when they are investing in mainstream OO
> architecture, they are also spending money on things they will never
> need, they would make different decisions.*

The industry is full of fads and hype. Sun is not a RDBMS vendor, so bashes RDBMS to cause managers to invest in Sun-bound stuff. Why the DB vendors don't fight back, I don't know. Perhaps because they have OODBMS products in their back room just in case, and will make money off of conversions if it happens. Also, the RDBMS vendors are guilty of not sticking with standards, and may have to admit this if they make swappability an issue.

>

> *It would be interesting if it is someone out there that actually uses
> something else but relational databases for persistence? And in that
> case, what type of applications and what database vendor. It would be
> even more interesting to know if someone actually have changed from
> relational database to OO database (or something else), whithout
> having to rewrite the business logic.*

OODBMS have certainly fell out of style. Most are being turned into some kind of hybrid or bridge product. Besides, what exactly an OODBMS is, was never clearly defined. Maybe that is its definition: "A bunch of independent records that don't have to follow the rules of relational". In other words, lack of a definition *is* the definition.

>

> /Fredrik

By the way, pitting OO versus relational causes nasty fights to break out.

Take care,
-T-
oop.ismad.com