

Re: OO vs. RDB challenge

Source: <http://coding.derkeiler.com/Archive/General/comp.object/2005-03/0674.html>

From: bruno modulix (*onurb_at_xiludom.gro*)

Date: 03/15/05

Date: Tue, 15 Mar 2005 14:53:57 +0100

Alfredo Novoa wrote:

> *On Mon, 14 Mar 2005 18:12:47 +0100, bruno modulix <onurb@xiludom.gro>*

> *wrote:*

>

>>>*If something can not enforce the business logic then it is not a DBMS.*

>>

>>*Ho, yes ?-) So no one of us never ever worked with a DBMS, cause such a*

>>*thing does not exists (by your definition).*

>

> *What a nonsense!*

>

> *Of course they exist: Oracle, DB2, SQL Server, etc.*

>

>>*May I remind you that a DBMS is a program that has it's own business*

>>*logic (it's business being, as the acronym implie, to manage data), and*

>>*that a not-so-small part of this logic (the one dealing with internal*

>>*data and indexes representation, IO access, request optimisation etc*

>>*etc...) cannot be enforced by a DBMS ?*

>

>

> *What a nonsense!*

What ? Stating that a DBMS is a program ? Stating that it's business is

to manage data ? Or stating that a DBMS cannot rely on a DBMS to

enforce (at least part of) the rules that pertain to it's business ?

> *You don't know what business logic is.*

I don't know (yet) what *you* mean by business logic.

>

>*That is not business logic!*

Depends on how you define business logic... let's see :

>>*Unless of course you define 'business logic' as 'any rule my favorite*

>>*RDBMS can enforce, and no other', which would make your assertion a*

>>*tautology – and would be a bit unfair too...*

- >
- >
- > *Of course my statement was tautology!*

of course...

- > *Business logic are: database structure, database constraints and*
- > *database derivation rules.*

Lol.

This (obviously biased) definition implies that there is no business logic in an application that dont use a database, so an application that dont use a database can only handle presentation logic.

As you like to say : "What a nonsense!".

- > *And an RDBMS is a software system that can enforce all that.*

I didn't say that a RDBMS could *not, never, in any way* be used to enforce some kinds of business rules, did I ?

- > *But it is not my definition, it is the definition you can find on any*
- > *book or web about business rules.*
- >
- > <http://www.businessrulesgroup.org/>
- > <http://www.brcommunity.com/>
- > <http://www.computerworld.com/printthis/2003/0,4814,84500,00.html>

Nothing I saw on theses pages supported your statement that business logic was directly tied to DBMS. If I missed something, please provide links to more accurate documentation.

From what I saw, it's all about defining the rules that the application (however it's implemented) must enforce to solve the customer's problem.

From this POV, I think it's perfectly valid to state that the 'business' of a DBMS program is to manage user's data and enforce user's specified rules (users in this context being the DBA and the application programmer). One exemple of the business rules for a RDBMS program is that a primary key is unique – I let you think about the others...

If it's the usage of the 'business' word in that context that bother you, please feel free to propose another – from my POV, it's equal.

- >
- >>>*Of course, but if we compare OODBMS's to RDBMS's then OODBMS's don't*
- >>>*have any technical pro.*
- >>
- >>*What you mean 'technical' pro ?–)*
- >

>
> *An objective pro.*

<troll>
Then what do you mean by 'objective' ?-)
</troll>

Ok, I'm not here to troll, and I pretty well understand what you mean.

Now what I meant here is that there is no such thing as an 'absolute' superiority in this domain. The 'best' tool *for a given problem* is the one that best solves *that* problem.

It's obvious that writing my programs in C could allow me to have much more efficiency at runtime than what I can get from a Python interpreter.

But the fact is that for what I'm doing actually (web applications), C would be a worst choice in terms of productivity – and my customer's priority is more about productivity (how much time will it take to implement this particular feature) than absolute efficiency (given that actual perfs are considered good enough by the customer itself).

>
>>*I've worked (and still have to) with RDBMS based CMS. At least half the*
>>*code is related to connecting to the RDBMS, building SQL queries,*
>
> *Your testimonial is irrelevant.*

Of course, since it doesn't go your way.

> *It is very probable that you were not*
> *using the SQL DBMS correctly,*

Of course, I must be a very bad and illiterate programmer, not educated enough to even have a clue about how to use a DBMS. This is of course the only possible explanation to the fact I observed.

When facts comes in contradiction with theory, you can dismiss facts or revise the theory...

> *and SQL DBMS's are a lot worse than*
> *RDBMS's.*

Would you categorize SQL Server, Oracle and Postgres as 'SQL DBMS' or 'RDBMS' ? and in the first case could you be kind enough to tell us about existing RDBMS ?

Alfredo, I know that you know much more than me about RDBMS, and I would not pretend being a better (or more expiremented or whatever) programmer than you are (which doesn't implies that I know nothing about RDBMS or programming...). Too bad you're so biased as to not even accept that

comp.object: Re: OO vs. RDB challenge

there exists a whole world of applications outside RDBMS based apps, and that RDBMS are **not** the alpha and omega of application programming – just a pretty nice and useful **tool**.

Have nice day !-)

Regards

```
--  
bruno desthuilliers  
ruby -e "print 'onurb@xiludom.gro'.split('@').collect{|p|  
p.split('.').collect{|w| w.reverse}.join('.')}.join('@)"
```