

## Re: OO's best feature survey results

**Source:** <http://coding.derkeiler.com/Archive/General/comp.object/2003-11/0024.html>

---

**From:** Topmind ([topmind\\_at\\_technologist.com](mailto:topmind_at_technologist.com))

**Date:** 11/01/03

Date: 1 Nov 2003 01:09:27 -0800

>>> *If you have a very flat class structure, you can completely equate  
>>> classes to relational tables, etc.*  
>>  
>>*Not if you have dynamic OOPL's, in which each "record" (object) can  
>>have as many or few attributes as it wants. This goes against  
>>relational as we know it, which "centralizes" the schema. However, it  
>>would be interesting to try dynamic relational, which is sort of the  
>>dynamically-typed (or "scriptish") way of seeing things applied to the  
>>relational model.*  
>  
> *OOPL is always a superset.*

Well, I guess I have to agree, but being a superset is not necessarily better. Sticking to a certain rigor has its advantages. That is why planned towns are generally more functional than shanty-towns, which abide by less rules and regulations.

But, dynamic relational would open up more possibilities and should be explored.

>  
> *OTOH, relational was the first language/system to have dynamic  
> resultsets in response to dynamic SQL queries!*  
>  
>>> *But, there is no reason to ever do  
>>> such a thing! I'm with you on the relational data model, and proper  
>>> use of relational databases.*  
>>  
>>*Then would you be comfortable saying that relational is generally  
>>better for modeling the domain itself and OOP for the networking and  
>>OS interfaces (where polymorphism is not hampered by dynamic interface  
>>changes as much)?*  
>  
> *Er, can't really parse that question.*  
>  
> *Relational is good because it gives you very nearly the mechanical,*

comp.object: Re: OO's best feature survey results

- > *meat-grinder approach to modelling that everyone wants, at least down*
- > *to 3NF. However, there are entire classes of apps that don't model*