

## Re: Opinons on code generators and 4GLs

---

*Source:* <http://newsgroups.derkeiler.com/Archive/Comp/comp.databases.pick/2005-08/msg00114.html>

---

- *From:* Glen <[nospamwebmaster@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:nospamwebmaster@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx)>
  - *Date:* Tue, 09 Aug 2005 23:58:29 -0400
- 

On Tue, 09 Aug 2005 15:48:32 -0700, Tony Gravagno  
<[g6q3x9lu53001@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:g6q3x9lu53001@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx)> wrote:

>While posting to my thread on physician practice management software,  
>I decided to take some OT thoughts to a new thread. I warn you, this  
>might just be a ramble. :)

>

>I am not particularly fond of code written by code generators unless  
>it comes from one of the currently maintained products, and even then  
>I don't usually like to have my hands tied by someone else's notion of  
>what I should be able to do in code or how I should do it. In the  
>back of my mind I'm fighting to avoid older products like Wizard,  
>Forge (sorry Chandru), or even SB+ paragraphs. I like ATGUI but only  
>if the business rules are truly divorced from the UI.

>

There's no such thing as a split model in this market simply because  
no one wants to maintain a UI and a logic model on separate levels.  
The skill here is in dataBASIC and that's where everyone wants full  
control. That ideology is changing, but I feel it's taking way too  
long to keep up with the rest of the DB world.

>I'm trying to not lump all of the GUI/RAD/IDE products together in  
>this, including SB+, DesignBais, OSMOSiS, Nucleus, Visage, Symbion,  
>WebWizard, or even FlashCONNECT and Coyote which are toolkits with no  
>UI. These products facilitate development, and as long as some part  
>of the code written in these products is portable, I have no problem  
>with them. Each one, however does tie you in with product-specific  
>nuances and our decision to use them includes having some idea of how  
>long they'll be around. Product differentiation is of course  
>unavoidable, but I personally prefer products with a lighter footprint  
>(no so much product-specific code required in the app) to those, like  
>Coyote for example that essentially change the language and coding  
>paradigm to support a new UI. The question any developer has to ask  
>is "what is the benefit I derive now vs the pain some tool might cost  
>later?" In most cases, "later" isn't a consideration, and most  
>developers assume that as long as we're working with something we like  
>now it doesn't matter who gets stuck with it later. And then there is  
>the idea, as with Coyote for example, that the product actually does

## Re: Opinions on code generators and 4GLs

- >have longevity and there will always be someone around to maintain app
- >code written around it, so who cares how it ties into the app?

That's where OSS comes in handy. OSS source code will always be available for maintenance and enhancement. If the open source development is centralized and kept updated, then there's no end-of-life for it. There are always things to consider when looking at OSS code for commercial development, including the license the code is released under. I think that, above all things, is the brick wall that make so many commercial developers wince and therefore shy from OSS integration. Well, you can't have an immortal product without opening it up for the public to maintain it. Opening up an interface module for your app is a small price to pay for a product that you know will always be around for enhancement and possibly redesign.

- >
- >What I really don't enjoy is getting a call from an end-user who needs
- >assistance and after discussion of all the great things we can do with
- >their MV system I find out their code is written with some commercial
- >or home-grown tool that isn't maintained anymore. So now the options
- >are to contract with someone else who has skills with the old code, to
- >manually rework the generated code, or to re-write. We pride
- >ourselves on working with a platform that is easy to maintain but then
- >we stick end-users with unmaintainable code. Often their only
- >recourse is to migrate. My business solution to this to focus on
- >mainstream products rather than picking up orphans. There are plenty
- >of people who enjoy the diversity of working with these orphan sites,
- >learning the code and becoming the new IT resource.

- >
- >When writing code, I'm particularly careful these days to try and
- >separate UI from rules, so that the rules can be used by any UI or
- >development tools that come along. This development style is a
- >choice, not absolutely required, and honestly there are few Pick shops
- >that will pay for the extra time required to implement this, so the
- >reality is that a lot of end-user code I write has some measure of
- >rules mixed with the UI code. Same goes for code that abstracts data
- >access and data structures from the rules. Good design and forward
- >thinking are nice ideals, but very few companies will pay for it.

That depends on how you present it. That typical lack of forward thinking is the reason we're in so many predicaments to start with. Being the lone IT 'expert' for a private company, I see this argument from both sides. When you're running a business on low margins, you don't want bells and whistles. Most companies aren't going to pay for future expansion possibilities, especially if growth and technology can not be projected far enough in the future to perform an ROI.

- >
- >Maybe this is further OT but it's interesting that no one in our
- >market writes material like "Design Patterns for MV" – of course I'm

Re: Opinons on code generators and 4GLs

- >sure few people would buy material like that too, unlike in the
- >mainstream world where this is a very important topic. I'm wondering
- >how other people deal with the concepts above and if we can derive any
- >best practices from the various coding techniques and business
- >decisions that relate to them.
- >

The survey from PickSource a few years ago showed that a good (80% + I think) percentage of developers and VARs would be willing to participate in a developer's guild or organization. I have way too many external projects right now to even consider starting such a thing, but I honestly believe that developers are hungry for info and ideas to help them make design decisions. If a couple of commercial figures could be convinced in funding such an organization, then we'd be one step closer to bringing the MV world together. Right now, there's still way too much seperation between flavor audiences. The only way it's gonna happen full-tilt is to bring in logos to show commerical support from all the major players.

Glen

<http://picksource.com>

<http://mvdevcentral.com>

--

Posted via NewsDemon.com – Premium Uncensored Newsgroup Service

----->>>>>>><http://www.NewsDemon.com><<<<<<<-----

Unlimited Access, Anonymous Accounts, Uncensored Broadband Access

.

- 
- Prev by Date: ***Re: Physician Practice Management Software?***
  - Next by Date: ***Re: Opinons on code generators and 4GLs***
  - Previous by thread: ***Dictionary Locking issues***
  - Next by thread: ***Re: Opinons on code generators and 4GLs***
  - Index(es):
    - ◆ ***Date***
    - ◆ ***Thread***