Subject: Re: GUI Builder limited to Windows platform (LONG)
Posted by Struan Gray on Mon, 16 Nov 1998 08:00:00 GMT
View Forum Message <> Reply to Message

Mirko , mirko_vukovic@notes.mrc.sony.com writes:

> On a related thread, while the programming features of IDL
> are growing, I would hope that the interactive features
> would do the same. Insight may be one attempt, but still
> very limited.

For me, the real benefit of the object orientation is actually during interactive use. The fact that objects are globally accessible and encapsulated makes it much easier to mix fully developed, 'production' code and new routines which are still at the brainstorming stage.

I still do a lot of ad-hoc programming for one-off cases, but anything that does get used more than a few times quickly migrates from being a 'temp.pro' file to a full-blown widget, especially if there are parameters I want to play with interactively (and these can be algorithmic parameters as well as functional ones). Not surprisingly, other people are much more willing to use my routines if I provide them in a packaged form with widget front ends, and now I can use those widgets myself while mucking about at the command line with the same data.

I like IDL and try to evangelise it in my lab. RSI deserve credit for staying multi-platform in a non-trivial way and they do seem to try and address users' concerns. Prices are always emotive, and personally I would love to see a cheaper way of distributing completed applications, but on the other hand I am aware of the dangers of insufficient cashflow to small businesses who are trying to innovate and I also can't help wondering how much of the price of IDL is made up of fixed licensing fees for things like OpenGL that can't be reduced.

My biggest worry is that there is a tendency for known bugs to persist forever. My favourite whinge is the lack of tools to monitor memory usage, but the immortal contour plot problems and the non-drawing of large offshore islands in direct graphics are other examples of hugely annoying issues that seem to have been simply ignored on the way to implementing sexier new features. There are similar unncessary limitations built into the current object graphics routines (the IDLgrTrackball object is the least object-like object I've ever seen and a total pain in the arse to use), and I fear that they too will be ignored or left to web-archives of known workarounds to sort out rather than being fixed properly.

Such casual disregard for 'minor bugs' is less of a problem in a community of IDL gurus who are essentially programming for themselves, but tilts the learning curve to unnecessarily steep angles for novice and casual users, which makes life harder for both myself and the RSI sales force.

Struan