Subject: IDL verses other interpertative languages (tcl/tk, khouros, pv_wave, etc).
Posted by Stuart E. Murray on Tue, 16 Dec 1997 08:00:00 GMT
View Forum Message <> Reply to Message

I am working on a Satellite Sensor Simulator and will require that I display several windows of data. I am trying to sort out what display package might be the best bet. My group currently uses IDL and tcl/tk and I have been asked to considered them since they are in house. I have talked with these folks about both languages and both have strong view points. It is obvious to me that this discussion group will also. But, for the sake of argument. I will pose this question on the tcl discussion group as well.

Has anyone done a pro or con evaluation about IDL (from RSI) verses other languages like tcl/tk?

1. I will have a real-time connection to another embedded machine.
2. There will be one data spigot (be it parallel port, USB, serial port, or network) to the other embedded machine. Does IDL support interfaces such as those or do I have to write my own drivers?
3. Has anyone used IDL in a real-time environment successfully (or unsuccessfully)?
3.1 The display of data in real-time may not be a strong issue if you can throw a lot of horsepower at the problem. However, since IDL is an interpertative language, you have to wonder about the impact of performance over straight compiled code? Is this a problem? Is Windows a problem?
4. How hard is it to incorporate C/C++ code into IDL? Literature suggests a thorough knowledge of IDL before attemping this and I don't have that yet.
5. IDL will probably be executed on a Laptop (166 MHz or better) under Win95 or NT (the OS may have a Real-Time exec handler, although there are people that would question a Real-Time exec handler can be done correctly). Other real time OSs might be considered like VxWorks, QNX, etc.
6. IDL learning curve steep?
7. What is the downside of IDL? (bloat code, etc).
8. IDL cost (both for the software & maintenance of code)? How easy is it for someone to come in and pick up where you have left off?

I would appreciate your comments. If this is not the place for this discussion, I apologize for the intrusion.

Stuart

Stuart Murray
Sandia National Labs.

Testers & Experimental Ground Stations Dept. 5715, M/S 0965
Bldg 890 Room 1048
Email: semurra@sandia.gov or snmurray@worldnet.att.net (home)
Phone: (505) 844-0160 (Office) or (505) 898-2795 (Home)
Fax:: (505) 844-5993 (Office) or Pager 845-0142-5066 (SNL/A)

Subject: Re: IDL verses other interpertative languages (tcl/tk, khouros, pv_wave, etc).
Posted by Martin Schultz on Thu, 18 Dec 1997 08:00:00 GMT
View Forum Message <> Reply to Message

Peter Mason wrote:
>
> On Tue, 16 Dec 1997, Stuart E. Murray wrote:
> <...about writing a Satellite Sensor Simulator>
>
> As much as I like IDL, I don't think that it is the right tool for this kind
> of task.
>
[...]

> ..."thing" called LabView. I say "thing" because it's quite unlike any
> other application I've seen. It's a "graphical programming environment
> for instrumentation" - one can generate very sophisticated programs with it,
> without having to type a line of code. Apparently it's very popular for this
> kind of work. My colleague can't seem to say enough good things about it.
> Indeed he has quite an evangelical attitude towards it, and wouldn't give up
> on catching me at teatime and remarking about "LabView" until I had spent a
> morning with him being (genuinely) impressed by its power and ease-of-use.
>

just a comment on LabView:
I once tried to estimate the time needed and the difficulties to overcome for using LabView to program the complete data assimilation of an atmospheric (trace gas) measurement station - judged against doing the job in a classical language like Pascal or C. So I took a 1 week course in LabView including much time for free play, and I had access to a full version for about another two weeks. My impression was: YES, it's doable, and probably very nicely so, but there would be several problems as soon as you leave the standard path of LabView modules, and I feared that maintenance could be a real issue (1: who likes to follow 100s of lines on a computer monitor [the connections between the so-called "virtual devices"], and 2: because LabView is so special you definitively need trained people to maintain the code, and training certainly takes MUCH longer than the 3 weeks I had). If you see some LabView demos or applications, that can be quiet impressive, but if you

have some pre-determined ideas in what you want to achieve and how you want to do it (e.g. automatic calibration and flagging of the data when this is done, synchronizing of a suite of about 30 measurements and fail-proof because the station had to be run un-manned), you may well find limitations in LabView as well.

My word on IDL is this:
IDL is really powerful in terms of graphic capabilities and it is fast
(although I have heard several comments that object oriented graphics is sensed rather slow). I have never tried to link IDL to C or C++ programs, but since IDL is a script language it bears some resemblance to C (actually a bit more like FORTRAN in some ways), so it is probably much easier to learn. I guess, with IDL you can start laying out a concept of your application relatively fast, and if you are a good prgrammer, chances are that you don't have to rewrite everything afterwards (only all the details of course ;-) Because IDL is interpretive, it is very convenient to test and debug code, and you can develop things in a modular fashion, adding them to the application when they are ready and tested. One of the greatest strenghts of IDL is that you almost never have to worry about array dimensions, this is very often done automatically (e.g. where you have to write
a loop in other languages, you can use something like A(*,3)=5. or A([1,3,7])=2. or A(where(B gt 3)) = 1). I am often surprised how easy it is to visualize data in the form I want with IDL, although I am always struggling if I want to improve my plots to convert them into a "publishable form" (controlling axis appearance, line thickness, annotating etc. - these things are doable, but sometimes rather counter-intuitive).

Hope this helps,
Martin.

---------- -------
Dr. Martin Schultz
Department for Earth&Planetary Sciences, Harvard University
186 Pierce Hall, 29 Oxford St., Cambridge, MA-02138, USA

phone: (617)-496-8318
fax : (617)-495-4551

e-mail: mgs@io.harvard.edu
IDL-homepage: http://www-as.harvard.edu/people/staff/mgs/idl/
---------- -------