
Subject: help with future application

Posted by [Peter Brooker](#) on Wed, 26 Jul 2000 07:00:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

We are presently considering how to handle simulation requirements for future technologies. (Present technologies are handled well by commercial codes). As such, we have access to various internal and university codes of which we have the source code. These codes are in C and FORTRAN. The problem is that we have to link these codes together and add GUIs to every thing to make them more user friendly. Also, once a new model appears in the literature, we want to code this up as fast as possible to try this out.

I see that IDL could be used to do this. For example, IDL could handle all the input and output with GUIs as well as using CALL_EXTERNAL to execute subroutine forms of all the codes. If IDL is used, then new models could be programmed up very quickly.

The question is though, Is IDL the best choice for this linking application? What about MATLAB? How do the Numerics of matrix inversion of MATLAB compare to IDL?

thanks-Peter Brooker

Subject: Re: help with future application

Posted by [Craig Markwardt](#) on Wed, 26 Jul 2000 07:00:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

"Mark Hadfield" <m.hadfield@niwa.cri.nz> writes:

>

> Another possibility in the same vein is Python:

>

In addition to Mark's great comments recommending Python, I believe there is a separate piece of software called SWIG, which enables easy (and automatic) interfacing between compiled libraries and Python (perhaps Perl too). It is essentially a code generator which creates the interface stub code for you.

Craig

--

Craig B. Markwardt, Ph.D. EMAIL: craigmnet@cow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response

Subject: Re: help with future application

Posted by [Mark Hadfield](#) on Thu, 27 Jul 2000 07:00:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

"Phillip David" <phillip_david@xontech.com> wrote in message
news:397F4604.B0F7025E@xontech.com...

> Peter Brooker wrote:

>>

>> We are presently considering how to handle simulation requirements for
>> future technologies. (Present technologies are handled well by
>> commercial codes). As such, we have access to various internal and
>> university codes of which we have the source code. These codes are in
>> C and FORTRAN. The problem is that we have to link these codes together
>> and add GUIs to every thing to make them more user friendly. Also, once
>> a new model appears in the literature, we want to code this up as fast
>> as possible to try this out.

> Peter;

>

> In fact, if the primary thing you're looking to do is to code up a GUI,
> you might want to check out TCL (Tool Command Language). It's a
> powerful language if that's all you're trying to do. I haven't used it
> myself, but know of a number of successful projects that use this
> approach, and can work quickly to get the GUI up and running.

Another possibility in the same vein is Python:

<http://www.python.org/>

Like TCL and Perl, it is oriented towards scripting and rapid application development but (as far as I can tell) the base language is more elegant & scalable. There is a package called Numeric Python that makes numeric calculation in Python feasible:

<http://numpy.sourceforge.net/>

I have used Python for a few small text-processing utilities & have considered it as an alternative to IDL (as a language it's *much* nicer) but it doesn't have the history of application to numerics and graphics that IDL has. However I think it would be very well suited to tying together modules written in C and Fortran, and I believe it is being used for this at Lawrence Livermore. The current maintainer for Numeric Python is Paul Dubois. He writes a regular column in Computers in Physics and has covered Python in the past.

Mark Hadfield

m.hadfield@niwa.cri.nz <http://katipo.niwa.cri.nz/~hadfield/>

National Institute for Water and Atmospheric Research

PO Box 14-901, Wellington, New Zealand
