URL: http://cxc.harvard.edu/ciao3.4/why/slang.html
Last modified: 12 March 2007

S−Lang as the CIAO Scripting Language

Return to: Why Index

The following is an outline on how it was decided that S−Lang should become the CIAO scripting language.

Introduction

On top of the usual benefits one expects from the use of a scripting language (e.g. extensibility, rapid prototyping, dynamic/loose typing, no compilation, no malloc/free worries, etc), there were several additional requirements:

- powerful mathematical capability (with transparent support for multi−dimensional arrays) that would serve as a basis for rapid scientific algorithmic development
- concise syntax that will be as natural as possible for scientists to adopt
- embeddability into existing applications
- as small a footprint as possible (CIAO is already fairly large)

Other Languages Considered

We evaluated several major languages in this regard (Tcl, Python, Perl) and several lesser−known languages/environments (glish, the CINT/Root system from Integral, and S−Lang). Here is a short summary of the review, the details of which can be found in the memo "Evaluation of Alternatives for adding Arrays, Arithmetic, and Programmability to CIAO2.0".

Tcl was rejected primarily because it is string−based. Bulk numerical operations are inefficient and imprecise in Tcl because internally conversions must be done between string and numeric representations.

Perl, despite its power and the zealous love of its adopters, was rejected primarily on syntactical grounds. Most astronomers would not find it an intuitive scientific language. This is especially true given that at the time of the review CIAO already had interactive applications with their own simple command languages (*ChIPS* for plotting and imaging, *Sherpa* for modeling/fitting) with which Perl would not have meshed cleanly at all.

Python was a considerably stronger contender, both because of the general power of the language and the wide array of Python modules available. Ultimately the choice to reject it came down to several factors:

- it is a much larger platform/language than is S−Lang
- the powerful numerical and array−arithmetic capability we sought was not native to the language, but rather provided in additional modules (NumericalPy), which adds complexity to building/distributing CIAO
- at the time of our review, Python was not positioned as an embeddable language; In fact, per Python FAQ 3.30:

```
In practice, few applications embed Python -- it's
much more common to have Python extensions...
```

All of these mean it's much harder to incorporate Python seamlessly/invisibly within CIAO. Putting Python "into" the CIAO *Sherpa* and *ChIPS* applications would be much like "having the tail wag the dog."

Additionally:

- writing non−trivial Python scripts would eventually require familiarity with object−oriented programming techniques
- SPACE/TAB/indenting issues: Python scripts traditionally require strict formatting, since class/module scoping is inferred from the indentation level of the code. From Dive into Python:

```
Python functions have no explicit begin or end, and
no curly braces that would mark where the function
code starts and stops. The only delimiter is a colon
(":") and the indentation of the code itself.
```

  While full−time programmers may be willing to adjust to this as the cost of using a powerful object−oriented scripting language (e.g., by using custom editor features to help with formatting), the idea of quickly prototyping an algorithm does not include the notion of paying strict attention to how many spaces you have on the line. It was felt that this would be a continual source of problems for scientists writing even simple scripts. Cut/paste can become rather troublesome, and so on.

The Benefits of S−Lang

In contrast, ***S−Lang*** is a procedural language with a C−like syntax which should be very familiar to a large majority of practicing scientists. While S−Lang admittedly has idiosyncrasies of its own, it does not require mastery of a distinctly new syntactic idiom (Perl), nor does it mandate strict editing format or the adoption of an object−oriented perspective (Python). Several CXC scientists, for example, have converted their IDL scripts to S−Lang with remarkably little effort (verging on the order of only minutes in some cases). S−Lang was designed from the outset to be embeddable, takes all of about 5 minutes to compile on an average system, and generates a shared object library that is less than 500K in size on Solaris. S−Lang was written by a professional physicist (Dr. John Davis, at the MIT CSR) and as a result provides intrinsic scientific/numerical capability that is superior to that of most scripting languages, and at the primitive level is nearly on par with that of commercial systems like IDL and MatLab.

While lesser−known than other scripting languages, S−Lang does have an international following, and a strong reputation for economy, quality, and speed. It is embedded within numerous extremely popular open−source applications (e.g., the SLRN newsreader, the Lynx web browser, the JED editor/emacs clone), and along with those applications is bundled with most Linux distributions (e.g., Redhat).

The chief disadvantages of adopting S−Lang are that the number of importable S−Lang modules available to the community is well below that of the other major scripting languages, and that the documentation corpus is likewise not as large. Given our niche, though, these were deemed less significant factors than those described above, and we are working on developing new modules as needs arise.

References

- S−Lang Homepage
- S−Lang Manual
- S−Lang Threads for CIAO
- Presentations from the CXC's internal S−Lang classes

The Chandra X−Ray Center (CXC) is operated for NASA by the Smithsonian Astrophysical Observatory.
60 Garden Street, Cambridge, MA 02138 USA.

URL:
http://cxc.harvard.edu/ciao3.4/why/slang.html
Last modified: 12 March 2007