How SAGE Works: Interfaces

William Stein (University of Washington)

May 11, 2006: UW SAGE Seminar

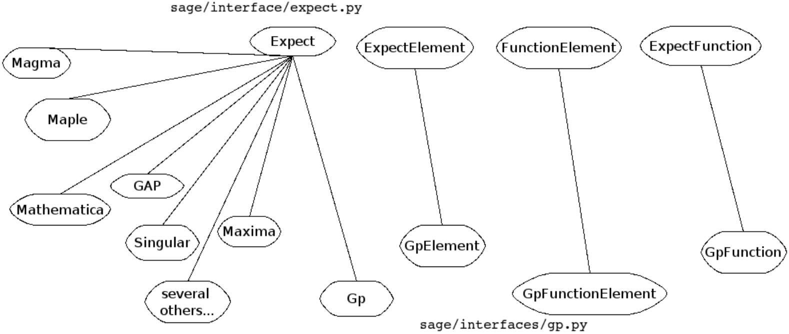
Abstract

This talk will be aimed at people who want to understand the architecture of SAGE. Today I will explain some of the details about how SAGE interfaces with computer algebra systems and how people interface with SAGE.

1. How SAGE **interfaces** with other computer algebra systems.
2. How SAGE **interfaces** with users via their web-browsers.

Part I: How SAGE interfaces with other computer algebra systems.

Class Hierarchy



(examples)

Pexpect and Pseudotty's

- **Pseudotty:** A device which appears to an application program as an ordinary terminal but which is *in fact* connected to a different process. Pseudo-ttys have a slave half and a control half.
- **Pexpect:** *makes Python a better tool for controlling other applications.* (`pexpect.sourceforge.net`)
  Pexpect is a pure Python module for spawning child applications; controlling them; and responding to expected patterns in their output. Pexpect works like Don Libes' Expect. Pexpect allows your script to spawn a child application and control it as if a human were typing commands.

Using pexpect: The Basic Idea

```
sage: import pexpect
sage: p = pexpect.spawn(’gp --emacs --fast --quiet’)
sage: p.maxread = 100000   # crucial for speed!!!
sage: p.expect(’\\? ’)
0
sage: p.sendline(’factor(2006)’)
13
sage: p.expect(’\\? ’)
0
sage: p.before
’factor(2006)\r\n\r\n[2 1]\r\n\r\n[17 1]\r\n\r\n
 [59 1]\r\n\r\n’
sage: p.close()
```

Difficulties

1. Getting subprocesses (and their children!) to quit when you quit SAGE.
2. I/O Prompts: make very obscure or embed control codes.
3. Control-C: How to break out of a computation.
4. Large I/O: use files.
5. Remote servers: Use the network – will greatly improve this summer (Yi Qiang).
6. Readline (weird formatting in server); often there are debug modes, e.g., in Maple; or emacs modes.
7. Support for direct interaction – coding sprint at Sage Days 1.
8. Unknown number of return values – Magma.

How to Create a New Interface

1. Decide which of the following programs is most similar to your program: gap, gnuplot, magma, maxima, genus2reduction, gp, mwrank, ecm, kash, maple, octave, singular, gfan, macaulay2, mathematica
2. Copy `sage/interfaces/similarsys.py` and modify as appropriate. Use the `log=` option to interface constructor for debuging.
3. Figure out what the true, false, equality, etc., symbols are in your system and code those into the interface.
4. Provide special functionality for tab completion (listing all defined functions), help on functions, etc.

How to Improve an Existing Interface

Let C be a computer algebra system, e.g., Maple, Macaulay2, etc.

1. Systematically work through a standard tutorial for C but using the SAGE interface; record anything that is difficult, impossible, or *unnatural* to do. Add functionality to interface to remedy these problems.
2. Try the interface I and if I.[tab] doesn't give all functions in the system, figure out how to get them and tell me!
3. Write conversion functions between SAGE objects and objects in C, e.g., matrices. We need far more of these!

Part II: Web Browser Interfaces to SAGE

1. Simple cgi-bin script interface
2. SAJAX – an AJAX interface to SAGE
3. SAGE Web Notebook – brand new local web-server interface

1. Simple cgi-bin script interface

This is at `http://modular.math.washington.edu/calc/`, but people have copied it for Kash, MAGMA, etc.

1. Started so Keith Conrad's student at UCONN could use PARI easily for their number theory class.
2. Uses apache2 and some simple Python cgi-bin scripts.
3. Provided way to "try out MAGMA" (and variant is now available at MAGMA website.)

Date: Mon, 19 Dec 2005 16:54:09 -0800
From: "John Cannon" <john@maths.usyd.edu.au>
William,

This is to formally advise you that your permission to run a general-purpose calculator based on Magma ends on Dec 31, 2005. [...]

Note that this does NOT affect the use of Magma in your modular forms site. Nor is it likely that we would withdraw permission for this use of Magma provided that Magma is properly acknowledged (as it has been in the past). In fact we are encouraging the use of Magma as part of the backend in similar servers.

Please confirm receipt of this letter.
Wishing you a happy Christmas,
John

2. SAJAX: AJAX interface to SAGE

Aaron Klem found an AJAX terminal application at `http://antony.lesuisse.org/qweb/trac/wiki/AjaxTerm`. It's a stand-alone webserver written with Python combined with a javascript page.
I modified it to make
`http://modular.math.washington.edu:8389/`

1. No authentication – need chroot. The AJAX term web server runs as a normal process. When a user connects, a chroot'd SAGE session starts set to timeout after an hour.
2. Multiple connections at once are allowed.
3. *Rumored* to be unusably slow, though I've not experienced this (i.e., it's great if the server is in "on campus").
4. No graphics. No scrollback. Cut and paste is very difficult.
5. Tab completion, history, etc., work well.
6. Hard to setup (because of chroot jail).

3. The SAGE Web Notebook

Alex Clemesha and I wrote it from scratch as part of SAGE (on Monday). It uses that Python has (as part of the standard library!) a web server built in.

- Completely standalone.
- Python for webserver; some Javascript in the web page.
- Not done! But I already like using it more than command line!
- Saves all I/O to a disk file whenever possible (if you gave optional name='file' argument).
- Multiple people connect to a single session at once.
- May become the standard SAGE interface; you start a local server on your computer (e.g., via Cygwin or VMware under Windows), connect to a local web address, and use SAGE. Or connect to a university server and use SAGE; same interface!
- Completely avoids need to create GUI for SAGE.

SAGE Web Notebook: TODO List

```
    [] Add authentication
    [] The "move to the current input box" javascript *only* works
       with firefox (not opera, not konqueror); also this should
       just keep the page position where it is rather than move it.
       Moving to a more AJAX-ish model would alternatively fix this, maybe.
    [] A. Clemesha: shrink/expand input/output blocks
    [] A. Clemesha: When hit shift-enter the next text box should be made
       into focus.
    [] Add plain text annotation that is not evaluated
       between blocks (maybe in html?)
       E.g., just make ctrl-enter on a block by HTML-it.
    [] Ability to interrupt running calculations directly
       from the web interface (no console access)
    [] Nice animation while a computation is proceeding.
    [] Some way to show output as it is computed.
    [] Option to delete blocks
    [] Make block expand if enter a lot of text into it.
    [] Evaluate the entire worksheet
    [] Theme-able / skin-able
    [] Embedded graphics from plots;
        also embed png from latexing of math objects (so they look nice).
    [] Downloading and access to exact log of IO to client SAGE process
    [] Save session objects as to log objects so don't have to re-eval?
    [] The entire page is resent/updated every time you hit shift-enter;
       using 'AJAX' this flicker/lag could be completely eliminated.
    [] When pressing shift-enter a line feed is inserted temporarily
       into the inbox, which is unnerving.

Try it and send me some ideas!
```