Automating Scientific Documents

Ana Nelson

April 9, 2011

Scenario

The day before you are due to hand in your thesis,
a serious problem is discovered with your dataset.

Question

How do we work so that this is a minor inconvenience, rather than a major crisis?

What if...

What if you could regenerate your thesis incorporating all the new data?

How?

1. Write complete scripts for everything*.
2. Create a master script which runs all the scripts in the correct order.
3. Integrate the results of scripts into documents (like your thesis).

Everything?

- generating/obtaining data
  - running simulation
  - interacting with an API
- cleaning data
- analyzing data
- graphing/presenting data
- installing/configuring software
- launching EC2 instances

(anything which is not scripted is a manual step which will need to be maintained manually, and whose relationship to the overall 'build' will need to be maintained manually)

Scripts

Scripts can be in R, Matlab, Python, Bash. Anything really (probably a combination of several languages).

Scenario

If we did this, how would we deal with our thesis situation?

1. Rerun our master script.
2. Check to make sure the claims we make in our thesis are still correct.
   -> add assertions/tests to our scripts which represent the claims we make in our thesis

Does it make sense to work this way?

Costs

- More Effort?
- Learn New Skills?

Benefits

- Error Reduction
- Productivity
- Standardization
- Iterability
- Reproducibility
- Auditability

What next?

So let's say we have automated all our scripts for data generation/cleaning, data analysis, graphing data, software configuration. And we have tests/assertions about our results. Now what? We have the raw ingredients (our scripts) for some automated scientific documents.

Raw Ingredients

What should we do with these raw ingredients?

- show them off as-is or with syntax highlighting
- run them and save the output they generate
- run them and save the data/image files/binary artifacts they generate
- extract metadata (class diagrams, sloccount, state charts)

Cooked Ingredients

What should we do with all this stuff?

- write process documentation describing the steps we took, backed up by sections of scripts demonstrating how those steps looked in code
- write blog posts about any aspect of our work that we want to share, incorporating source code and/or artifacts
- write PDF journal articles featuring your results
- write your thesis
- ?

Improvements

What should go into a system to do all this scripting stuff?

- dependency management (so I don't have to write the master script myself)
- caching (so I don't always have to run everything when I want to update just 1 part)
- data sharing (so it's easy to pass data between scripts and have it available when writing documents)

And...

And while we're at it let's not each reinvent the wheel many of these automation scripts have common elements why not make it really easy to

- do syntax highlighting
- compile code
- execute code
- run tests
- interact with APIs

and share scripts with each other

If you've come this far...

If you have all these elements
you have a powerful tool for

- reproducible (computational) research
- software documentation
- technical blogging
- document automation...
  - books
  - websites
  - tutorials

Dexy

`http://dexy.it`
`http://blog.dexy.it`
`http://bitbucket.org/ananelson/dexy-templates`
`http://bitbucket.org/ananelson/dexy`