Literate Research versus Reproducible Research

Li-Thiao-Té Sébastien
LAGA, CNRS UMR 7539
Université Paris 13
lithiao@math.univ-paris13.fr

December 5, 2012

Dear colleagues,

Like many I have struggled with research and source code that I have now forgotten how to use. Or not quite. When I started doing research, I was wise enough to make it simple to run my programs and I can still get results out of them. What I fail to perform are the tiny adjustments to the code that are required to re-use the programs in my current line of research, for writing papers, slides, posters or basically anything else.

My previous research is still reproducible, but no longer usable.

Of course I did not write enough documentation. Of course I did not write enough comments inside the source code. I fully take the blame. Unfortunately, I believe that this is the majority case in software engineering, in scientific research but also in general.

So last year, I decided that I needed to upgrade my methods and procedures and I wrote the Lepton software[2] to do just that. Let me stress the following point. Lepton is a tool to make it easy to create work — scientific research, teaching activities, software engineering or whatever — that hopefully I can re-use and modify in the long term. I called this "literate papers" in [3]. Lepton documents are reproducible research papers merely as a side effect of the way they are designed.

Do not get me wrong. Reproducible research is a must-have feature. Being able to reproduce exactly the results that were obtained by fellow researchers is a guarantee of the scientific quality and the reality of the phenomenon under scrutiny. For instance, the laws of physics are completely reproducible. (I am referring to apples falling from a tree). And we do research to unearth explanations for the exceptions to the well-established rules.

However, I get the feeling that we miss the point when we attempt to craft systems to provide perfect reproducibility to research results.

First of all, software are the tools by which we enslave computers to do our bidding. Let us not forget that tools should be practical. Reproducible research frameworks should not hinder researchers. On the contrary, such systems should provide additional functionality with as little a cost as possible. My opinion is that the current frameworks are not designed for the researcher. Some are designed for archiving research results. Some systems are designed for datamining with semantic tags. Some systems are designed for interactive web demonstrations. Some systems are designed for a specific architecture or software environment. In some sense, current proposals are computer-readable by design which is often the opposite of human-readable.

In fact, this is certainly the consequence of taking reproducibility mainly for the purpose of publishing research results. When submitting a manuscript, it is natural to follow guidelines, and these may include recommendations about how to provide input data, source code and

intermediate results. Frankly, this is often a hassle. Can we not provide the properties of a reproducible research paper beyond the limited scope of "publish or perish" ?

I designed Lepton for me to work with. In particular, I designed it to be

- trivial to learn and use,
- compatible with existing research methods, including current paper submission pipelines,
- universal, i.e. not dependent on a particular environment, programming language, publication format, etc.
- fit for collaborative work with colleagues without Lepton.

So far I have written two conference papers with this software as well as a few manuscripts in preparation. What I am most pleased with is that I use Lepton for almost every computer related task, from implementing research methods and writing test subjects down to the miscellaneous software projects that occupy my spare time. I will gladly demonstrate Lepton during the ICERM workshop.

But Lepton is no more than a means to an end. We need no tools for our research to be re-usable in the long term. What we need is to do research that humans can understand. There will come a time in the future when our work becomes obsolete, because no one bothers to maintain backward compatibility, emulators for legacy hardware, or when it is just faster and more computationaly efficient to reimplement than to use the source code from the archives. At that point in time, what is critical is not the source code but rather its documentation.

Literate papers are to research papers what literate programs are to source code. I believe that Knuth's ideas[1] did not get the long term adoption that they deserved. In literate programming, "source code" is a work of literature; consequently we should write a report on the implementation details, and embed the source code inside it. This provides a full-fledged publishing system such as LaTeX to writing code documentation. Likewise, Lepton makes it easy to embed all the elements of a research paper in the same literate paper, and makes it easy to document everything: the input data, the source code, the methods but also the scripts and commands that glue all these elements together and contain the nifty parameters used to produce the results, i.e. the figures and tables.

In the end, it all comes down to this: I hope that my research results in mathematics will be useful to others and that they can re-use my work. With or without Lepton.

Sincerely,

References

[1] D.E. Knuth and Stanford University. Computer Science Dept. *Literate programming*. Center for the Study of Language and Information, 1992.

[2] S. Li-Thiao-Té. *Lepton User Manual*.

[3] Sébastien Li-Thiao-Té. Literate program execution for reproducible research and executable papers. *Procedia Computer Science*, 9(0):439 – 448, 2012. Proceedings of the International Conference on Computational Science, ICCS 2012.

[4] Sébastien Li-Thiao-Té. Literate program execution for teaching computational science. *Procedia Computer Science*, 9(0):1723 – 1732, 2012. Proceedings of the International Conference on Computational Science, ICCS 2012.