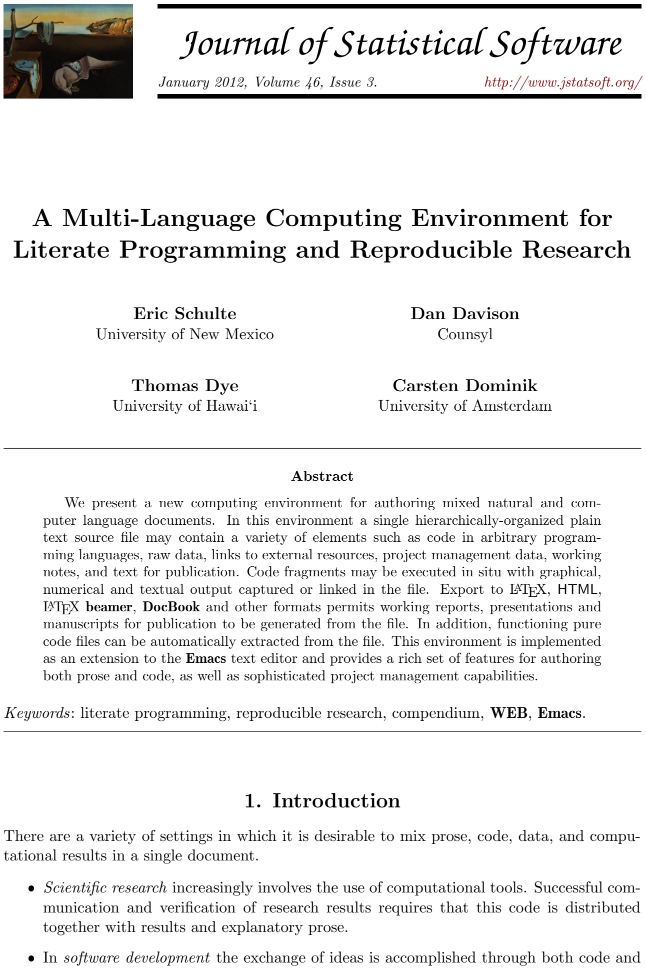
Literate programming as a tool for scientists

Heikki Lehvaslaiho, CBRC, KAUST

March 28, 2012

Contents

(This document is saved with extension allowed by Mediawiki, literate.txt. The original was called literate.org to prompt emacs to open it in org-mode. In emacs, you can type ‘M-x org-mode’ start org mode on any document.)



Journal of Statistical Software
January 2012, Volume 46, Issue 3. http://www.jstatsoft.org/

A Multi-Language Computing Environment for Literate Programming and Reproducible Research

Eric Schulte
University of New Mexico

Dan Davison
Counsyl

Thomas Dye
University of Hawai‘i

Carsten Dominik
University of Amsterdam

Abstract

We present a new computing environment for authoring mixed natural and computer language documents. In this environment a single hierarchically-organized plain text source file may contain a variety of elements such as code in arbitrary programming languages, raw data, links to external resources, project management data, working notes, and text for publication. Code fragments may be executed in situ with graphical, numerical and textual output captured or linked in the file. Export to LaTeX, HTML, LaTeX **beamer**, **DocBook** and other formats permits working reports, presentations and manuscripts for publication to be generated from the file. In addition, functioning pure code files can be automatically extracted from the file. This environment is implemented as an extension to the **Emacs** text editor and provides a rich set of features for authoring both prose and code, as well as sophisticated project management capabilities.

Keywords: literate programming, reproducible research, compendium, **WEB**, **Emacs**.

1. Introduction

There are a variety of settings in which it is desirable to mix prose, code, data, and computational results in a single document.

- *Scientific research* increasingly involves the use of computational tools. Successful communication and verification of research results requires that this code is distributed together with results and explanatory prose.
- In *software development* the exchange of ideas is accomplished through both code and

Based mainly on article :

Schulte, E, Davison D, Dye T, Dominik C: A Multi-Language Computing Environment for Literate Programming and Reproducible Research”. 46(3):1–24, 2012.

1 What is literate programming?

The basic idea is that [1]:

- Programs are useless without descriptions.
- Descriptions should be literate, not comments in code or typical reference manuals.
- The code in the descriptions should work. Thus it is necessary to extract the real working code from the literary description.

"Let us change our traditional attitude to the construction of programs: Instead of imagining that our main task is to instruct a computer what to do, let us concentrate rather on explaining to human beings what we want a computer to do." [2]

2 What is reproducible research?

The term **reproducible research** was first proposed by Jon Claerbout at Stanford University (2009) and refers to the idea that the ultimate product of research is the paper along with the full computational environment used to produce the results in the paper such as the code, data, etc. necessary for reproduction of the results and building upon the research.[3]

3 Examples of literate programming environments

3.1 General purpose

- Haskell - support for literate programming
- Sweave - R extension
- POD - Perl Plain Old Documentation
- Scribble - documentation language in scheme
- Org-mode - Emacs major mode for code in any language

[1] http://users.stat.umn.edu/˜geyer/Sweave/

[2] Knuth DE (1984). Literate Programming. The Computer Journal, 27, 97111.

[3] http://en.wikipedia.org/wiki/Reproducibility#Reproducible$_{\text{research}}$

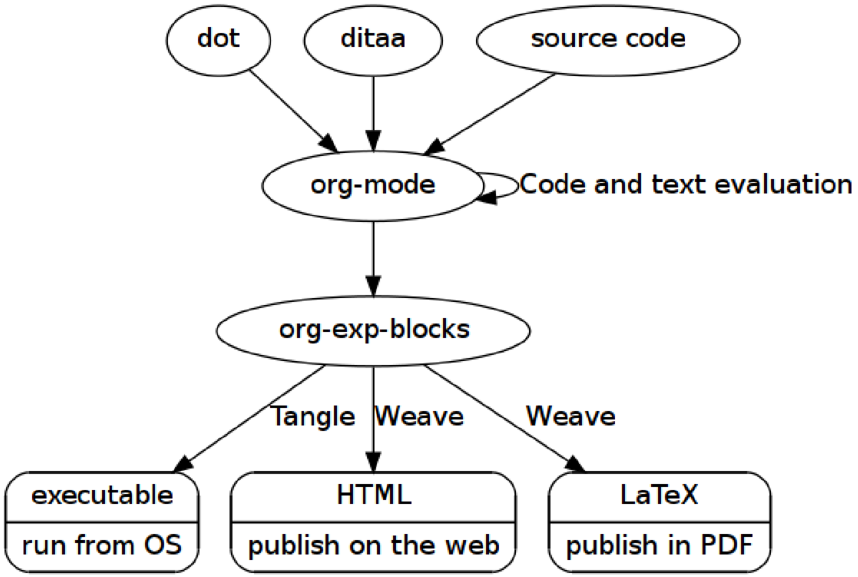
3.2 Scientific

- Sage - web based mathematician's tool (python)
- Galaxy - web based bioinformatics tool
- Taverna - work flow engine

With the data, code and text of a project stored in the same document, which can be exported to a variety of formats, the future reproducibility of the work is enhanced without placing a great burden on the author.

4 Key terms

- **Weaving**
  - Export of code and results into an other media (e.g. HTML, PDF)
- **Tangling**
  - Code blocks can extracted from the document for interactive execution in different order



5 Org-babel

- Originally additional module building on the emacs org-mode, powerful plaint text management environment:
  - http://orgmode.org/
  - http://orgmode.org/worg/org-contrib/babel/
- Now part of org-mode

6 Use cases

Scientific research

A research project should document:

- The data being studied;
- Details of calculations and code used in data analysis;
- Methodological conventions and assumptions;
- Decisions among alternate analytic paths.

Software development

- Code and prose mixed with graphics (UML) provide the justification

Education

- This talk.
  - emacs configuration is in the CBRC git repository:

    ```
    git clone ssh://git@git.cbrc.kaust.edu.sa/data/git/emacsd_heikki.gi
    ```

- Programming basics with Perl http://baloo-dev.cbrc.kaust.edu.sa/live/

7 Languages currently supported

- Python
- Ruby
- Perl
- R
- Shell
- Latex
- emacs-lisp
- Ditaa
- Dot
- Plantuml

7.1 Python

```
def hello(str):
    return "Hello,␣" + str + "!"
return hello ("Seb")
```

7.2 Ruby

```
require 'date'
"This␣was␣last␣evaluated␣on␣#{Date.today}"
```

7.3 Perl

Example perl code that produces different output every time ran.

Note how the block can be “tangled” from this document to be a standalone executable script.

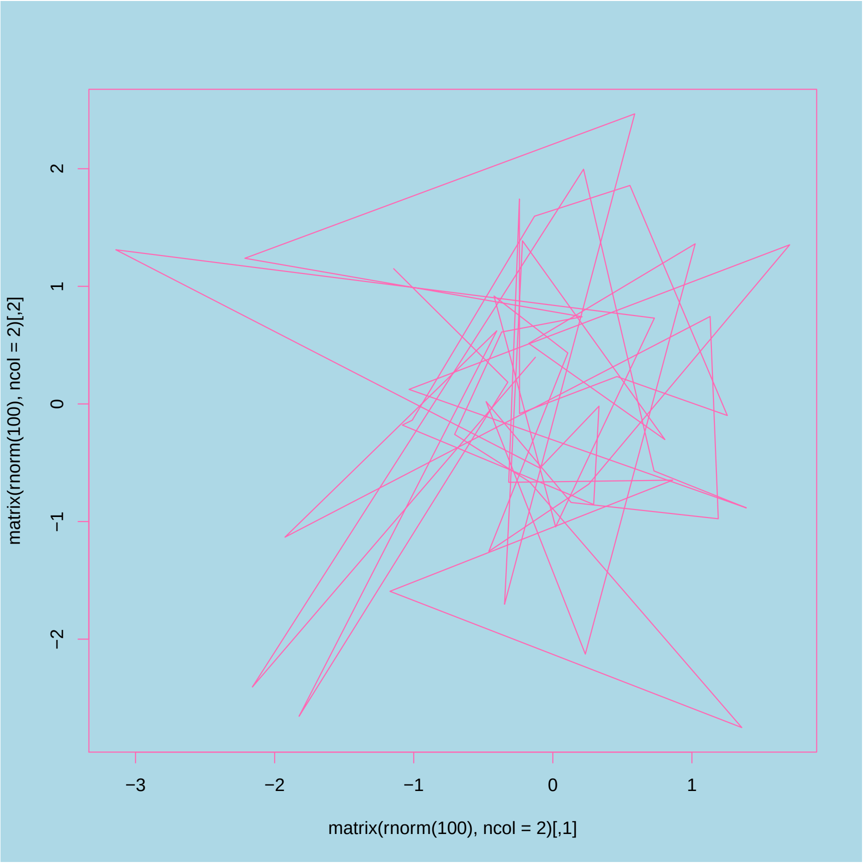
```
use v5.10;
my @a = 'a' .. 'g';
while (@a) {
    my $rand = int(rand(10));
    if ($rand <5) {
        say "$rand␣", (shift @a);
    } else {
        say "$rand␣", (pop @a);
    }
}
```

```
8 g
7 f
0 a
2 b
1 c
0 d
1 e
```

7.4 R

```
plot(matrix(rnorm(100), ncol=2), type="l")
```

More in http://orgmode.org/worg/org-contrib/babel/languages/ob-doc-R.html



7.5 Shell

```
ls -1
```

7.6 Latex

[[file:simpleEQ.png]]

7.7 emacs-lisp

Code block in very basic elisp:

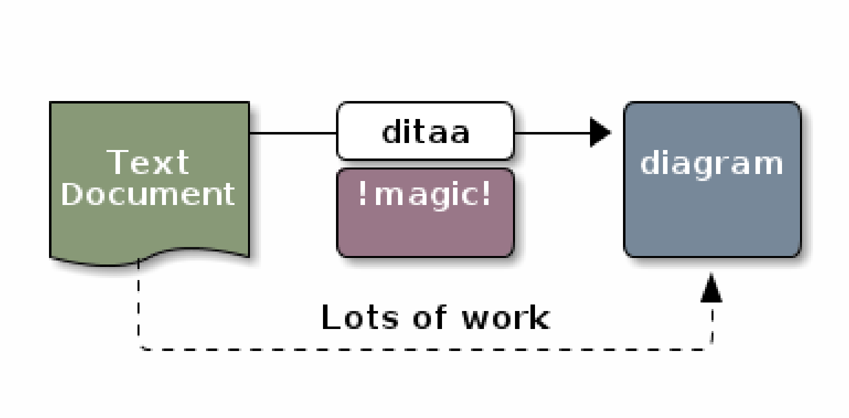
```
(* 2 n)
```

Call basic-elisp code with value 4:

```
8
```

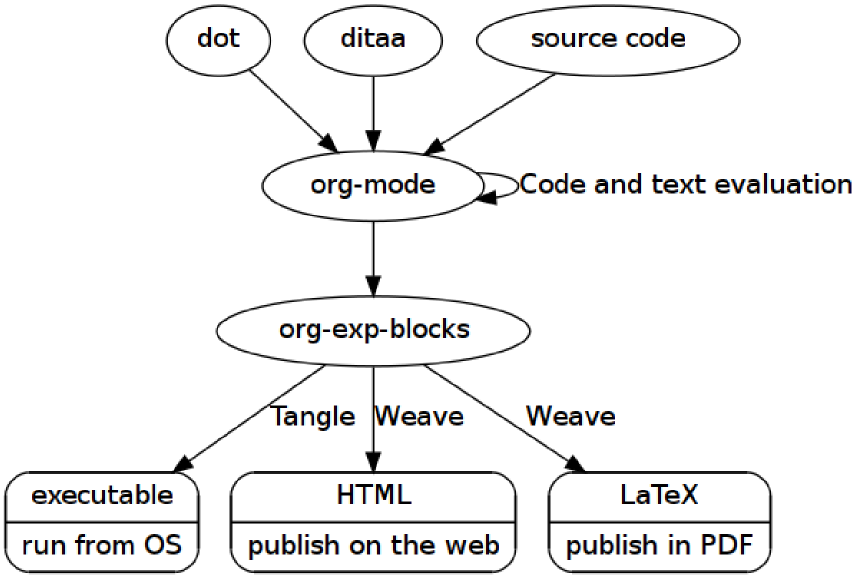
7.8 Ditaa

- http://ditaa.org/
- http://ditaa.sourceforge.net/
- Simple ASCII drawings get converted to PNG images.
- DitaaEps, a separate rogramme can convert to EPS and PDF



7.9 GraphViz

- http://www.graphviz.org/
- Graphing tool using dot language
- Plenty of extensions in various languages, e.g. GraphViz for Perl

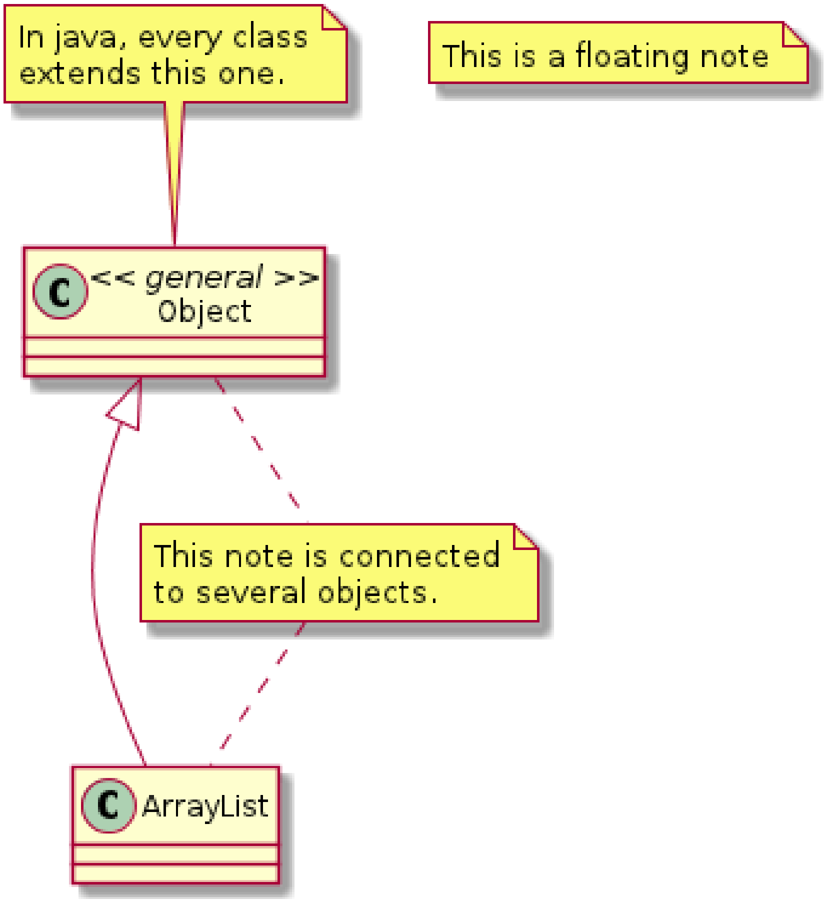


7.10 PlantUML

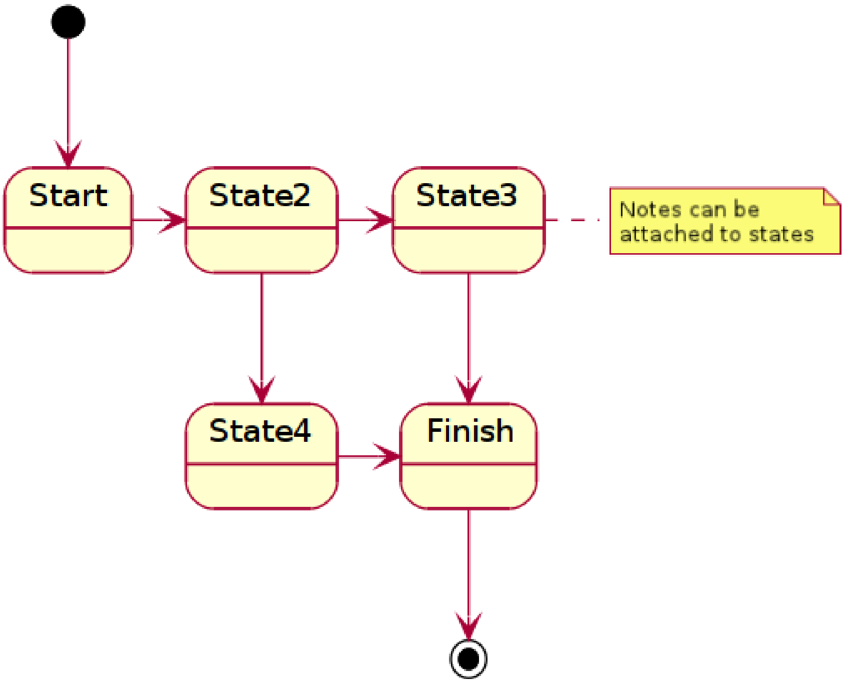
- http://plantuml.sourceforge.net/
- Implementation of **U*nified *M*odeling *L*anguage - UML is the standard representation *any** models
- Simple text gets converted to

Types of diagrams supported:

- sequence
- use case
- class


In java, every class extends this one.
This is a floating note
C
<< general >>
Object
This note is connected to several objects.
C
ArrayList

- activity
- component
- state



- object

8 The buzz word?

Open source. The Bioconductor

Open source. The Bioconductor project has a commitment to full open source discipline, with distribution via a SourceForge.net-like platform. All contributions are expected to exist under an open source license such as Artistic 2.0, GPL2, or BSD. There are many different reasons why open-source software is beneficial to the analysis of microarray data and to computational biology in general. The reasons include:

- To provide full access to algorithms and their implementation
- To facilitate software improvements through bug fixing and plug-ins
- To encourage good scientific computing and statistical practice by providing appropriate tools and instruction
- To provide a workbench of tools that allow researchers to explore and expand the methods used to analyze biological data

- To ensure that the international scientific community is the owner of the software tools needed to carry out research
- To lead and encourage commercial support and development of those tools that are successful
- To promote reproducible research by providing open and accessible tools with which to carry out that research (reproducible research is distinct from independent verification)

R note:

Reproducible research and automated report generation can be accomplished with packages that support execution of R code embedded within LaTeX, OpenDocument format and other markups.[4]

The goal of reproducible research is to tie specific instructions to data analysis and experimental data so that scholarship can be recreated, better understood and verified.

R largely facilitates reproducible research using literate programming; a document that is a combination of content and data analysis code. The Sweave function (in the base R utils package) can be used to blend the subject matter and R code so that a single document defines the content and the algorithms.

9 Galaxy

Galaxy is scientific workflow, data integration, and data and analysis persistence and publishing platform for computational biology.

- http://galaxyproject.org/

Full analysis of sequence data can be performed with galaxy. Each step is stored and the whole sequence can be replayed by anyone.

[4] http://cran.r-project.org/web/views/ReproducibleResearch.html