Literate Scientific Software

Dan Szymczak

Computing and Software Department
Faculty of Engineering
McMaster University

Ernie Mileta Visit, Jan. 12, 2016

Overview

1. Literate Scientific Software
2. LSS Today: Building on our Previous Example.
3. The Current Framework Design
4. Next Steps.

Literate Scientific Software

- Motivation
  - Improve verifiability, maintainability and reusability.
  - Save money and time
- One “source,” multiple views
  - Requirements
  - Design
  - Test Cases
  - Build instructions
  - ...

Recap

Last time:

- Took a look at a simple example from a project involving a fuel pin.
- Discussed the challenges of managing change throughout the software documentation.
- Proposed encapsulating all of the requisite knowledge in one source composed of “chunks”.

Example: h_g and h_c

A simple example taken from the SRS for FP

Let's take a quick look at the hand-written Software Requirements Specification (SRS) for h_g and h_c and compare it to the one generated by the LSS framework.

Example: h_g and h_c

The source

The current source for this example is broken up into three pieces.

1. The recipe
2. Global/often reused chunks
3. New chunks specific to this example

Example: h_g and h_c

The Recipe

The recipe is how we specify exactly what we want our view to look like.

It tells the generator what information is needed and how that information should be displayed.

Example: h_g and h_c

Global Chunks

Global chunks are chunks that will often be reused across many different projects. As such they are grouped with relevant related information and kept in specific source files.

This example contains one set of global chunks: SI Units.

Example: h_g and h_c

New Chunks

This example contains two obvious chunks that need to be declared: h_g and h_c.

There are also a few less obvious chunks that need to be declared: the values which h_g and h_c depend on in their equations. For example τ_c, k_c, etc.

Framework Design

Chunks

Chunks come in many flavours: The simplest type has only one field (name). Each more complex chunk builds off this base and adds (at least) one new field.

All of the more complex chunks are built up from simpler chunks.

Currently there are a few different types of chunks including (but not limited to) those for describing concepts and variables.

Framework Design

Recipes

Recipes are specified using a domain-specific language.

They tell the generator both what information needs to be laid out and how that information should be laid out for the current document type.

There will exist many types of Recipes, however currently only the SRS recipe is working.

Framework Design

The Generator

The generator is fairly straightforward. It:

- Takes a given recipe and looks up all the requisite information.
- Creates an internal generalized representation of the document being generated.
- Translates that internal representation to one specific to the format of the desired output.
- Prints the document to the desired output format.

Currently the generator only prints to a TeX representation (the same as the example SRS source), which can then be typeset.

Next Steps

What next?

- Generate the rest of example 1.
- Finish implementing options for different document ”views”.
    - Ex. SRS with or without derivations.
- Implement additional document types.
- Generate the source code from the equations.
- Implement more examples using the framework.

Thank You!