# Dynamic Documentation Generation and Crowdsourcing in SasView


Brayden Miller
*Albert Einstein High School*

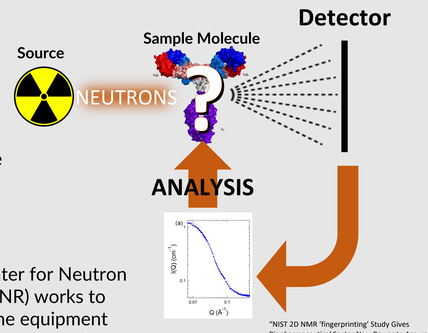
Dr. Paul Butler, Mentor


## Background

***Small angle scattering (SAS): What is it?***

Scientists ask a question:
***WHAT ARE CHARACTARISITICS OF THIS MOLECULE?***

SAS is used to determine these characteristics, giving us data that must be analyzed. This can be done manually but is greatly enhanced using analysis software.




"NCNR East Guide Hall." *NIST*, April 17, 2017. https://www.nist.gov/image/20170417016guidehalljpg.

The NIST Center for Neutron Research (NCNR) works to supply both the equipment (shown left) and the analysis software necessary for modern SAS techniques.

"NIST 2D NMR 'fingerprinting' Study Gives Biopharmaceutical Sector New Power to Assure High Quality Monoclonal Antibody Therapeutics." NIST, December 21, 2018. https://www.nist.gov/news-events/news/2018/12/nist-2d-nmr-fingerprinting-study-gives-biopharmaceutical-sector-new-power.

### *SasView*

SasView is a SAS software analysis suite.

- Capable of a wide variety of analysis operations
- **Extensive documentation *~140 Pages, some ~6,000 words***
- Maintained by community

*SasView is an open-source, collaborative, and international project*

Open source – ***READABLE AND CONTRIBUTABLE BY ALL***

Collaborative – ***PROTECTED FROM THE "BUS FACTOR"***

This means SasView is *developed sustainably*, with the goal that it can continue to be developed in the future. Unlike many alternative tools for SAS analysis, it has a consistent release cycle.

*If a bus were to hit members of the developer team (or if they were to retire), the project will live on!*


Ahmed, Adnan. "The Bus Factor." *Tajawal* (blog), May 4, 2018. https://medium.com/tech-tajawal/the-bus-factor-6ea1a3ede6bd.

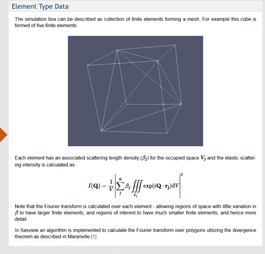
## The Documentation Problem

***WHAT DOES DOCUMENTATION LOOK LIKE?***

```
97 Element Type Data
98 ^^^^^^^^^^^^^^^^
99
100 The simulation box can be described as collection of finite elements forming a mesh.
101 For example this cube is formed of five finite elements:
102
103 .. figure:: vtk_mesh_example.png
104    :align: center
105
106 Each element has an associated scattering length
107 density ($\beta_j$) for the occupied space $V_j$ and the elastic scattering
108 intensity is calculated as
109
110 .. math::
111    I(\mathbf{Q}) = \frac{1}{V}\left|\vert\sum_j^N\beta_j\iiint\limits_{V_j}\exp(i\mathbf
112
113
114 Note that the Fourier transform is calculated over each element - allowing
115 regions of space with little variation in $\beta$ to have larger finite
116 elements, and regions of interest to have much smaller finite elements, and
117 hence more detail.
118
119 In SasView an algorithm is implemented to calculate the fourier transform over
120 polygons utilizing the divergence theorem as described in Maranville
121 [#MARANVILLE1]_
```

***What we write***
*(not user-friendly)*

*Compiling + Formatting*

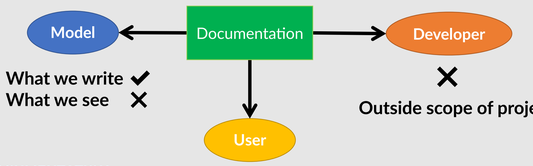



***What we see***
*(user-friendly)*

Files are written in a mix of Restructured Text (ReST) and LaTeX for embedded mathematics. The viewable files are structured using HTML.

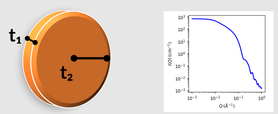
## *The Documentation Problem (cont'd)*

SasView has three types of documentation, each with its own problems:




***MODEL DOCUMENTATION***

**Fitting** is a core analysis in SasView. If users know the general shape of their molecule, they can write an equation, called a model, to represent its shape. Variables in this model represent unknown parameters, like radius and length below.




$I = f(t_1\ t_2)$

Researchers frequently write their own models and share them with the community. Writing documentation is possible, but it is **not visible to users when 'Help' is pressed for their models.**

***USER DOCUMENTATION***

User documentation assists users with the functionality of SasView and some of the theory involved in its calculations.

***SASVIEW IS COLLABORATIVE*** ***DOCUMENTATION MUST BE CONSTANTLY EVOLVING***

*We need to have an easy solution for letting the community edit documentation!*

Users may see errors in documentation or be knowledgeable in an area of documentation and want to contribute. The current process involves:

- Using a GitHub account
- Cloning a repository
- Installing dependencies
- Running setup and config files via command line
- Edit raw RST documentation files by navigating source tree
- Running Makefile via command line to view HTML version

***THIS IS TOO DIFFICULT OR TIME CONSUMING FOR A LARGE SET OF USERS!***

## *Objective:*

**Design Considerations:**

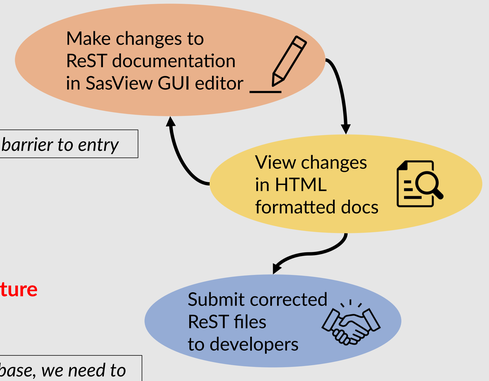
*Purpose is to lower barrier to entry*

**Functional**

**Easy to use**

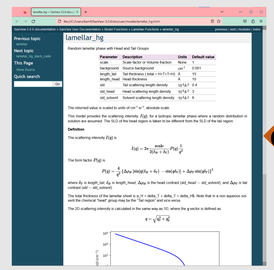
**Minimal changes to code structure**

*SasView is a large codebase, we need to develop sustainably*



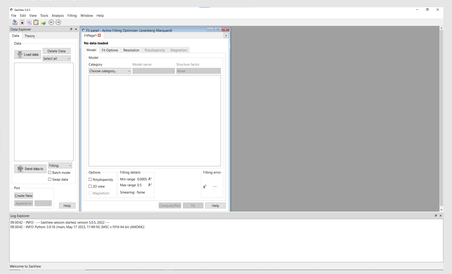

## *Before*

*Web Browser*



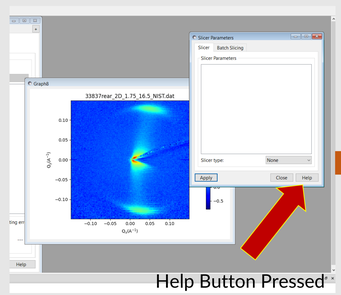
*OPENS*

*SasView*



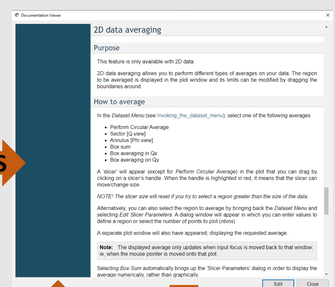
## *After Project*

*SasView*




*OPENS*

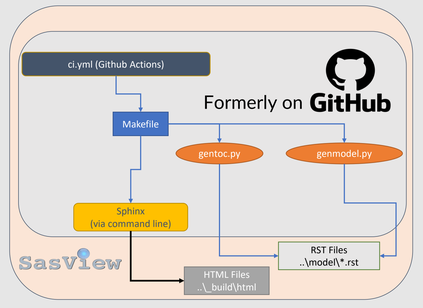
*Documentation Viewer* ***inside SasView***

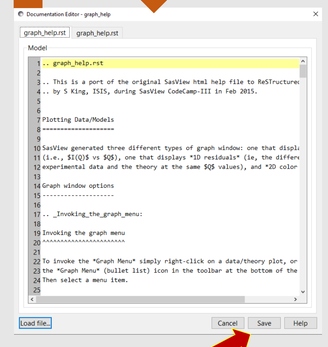



*REFRESHES* *OPENS*

Edit Button Pressed

## *Functionality Included*




Save Button Pressed

## *Summary*

Accomplishments:

- Allow users to edit documentation locally
- Create opportunities for crowdsourcing SasView
- Documentation for community models is now visible

Bonus:

- Math will display regardless of browser settings

Future work:

- Automatic submission of edits to documentation
- Documentation regeneration scripts can be optimized
- Math still needs internet connection to display correctly