SasView 5 Year Roadmap

The purpose of building and operating large scattering facilities is to provide unique tools to answer new scientific questions with the final presentation of results (usually in the form of a paper) as the output. The biggest obstacle to that output is often the analysis of the acquired data. Data analysis software has been variously viewed as being in the domain of the scientist using the facility, a service to be provided by scattering facilities, or as the individual responsibility of the scientists running the facility beamlines. The result has been a proliferation of packages and libraries, many written and supported by one key person, often not as their primary responsibility [1].

Over the past decade several trends have contributed to exacerbate the analysis bottleneck: 1) As the techniques have matured the user pool has broadened. This combined with an apparent decrease in the overall level of programming taught to scientists, means that fewer users are capable of building their own analysis tools. 2) With the increasing maturity of the field, a large amount of basic modeling is well understood and developed. Even those capable of coding their own should not be wasting their time re-inventing the wheel but focus on new science and perhaps new analysis developments to enable that new science. 3) The quantity of data being produced by instruments and the complexity of the experiments being performed have increased. 4) Finally, as the general software landscape has moved towards increased quality of usability and support, users of scattering facilities, in particular new users, have similar expectations of the software they use to operate the instrument and process and analyse their data.

To enable the production and maintenance of software that meets users expectations, a greater level of resources needs to be applied to the problem. This has been recognised in the neutron scattering community through projects such as DANSE [2], Mantid [3], CCPSAS[4] and projects within the European Union Horizon 2020 programme such as SINE2020[5]. While each facility or scientist may not be able to commit the necessary resources, pooling of expertise between facilities and with the user community into a single project can amplify the effort and provide gains in quality, functionality, longevity and supportability with obvious benefits to both the facilities and the users.

The aim of the SasView project is to provide open source, *collaboratively developed software for the analysis of small angle scattering data*

The collaborative development model is designed to encourage and enable contribution from a range of experts in small angle scattering, computer science and software development. Users of the software are encouraged to participate through bug reports, submitting code and by integrating their own scattering models.

Development Model

The SasView project is co-ordinated by a small management team who are also members of the broader development team. The development team currently consists of scientists and software engineers from scattering facilities around the world. Progress is monitored and project direction discussed at bi-weekly video conferences. The work is divided into various work packages that bundle together related tasks and issues. The project aims for a major release each year, with minor bug fix releases as required in between.

The source code for SasView is hosted on GitHub, allowing anyone to fork the code base, add functionality or fix bugs, and easily request that the code be merged into the project. The use of unit tests and code quality metrics is employed to help manage the contribution of code from a varied and dispersed developer base.

As of 2015, few if any of the developers have SasView as a major component of their job assignments. Given the nature of software development this currently means most development occurs during annual code camps. While this has proven to be a highly successful approach, it limits the rate of progress. The Roadmap presented below is intended to give guidance to the development team and to our stakeholders as to the direction we aim to take over the next 5 years. It is an optimistic one predicated on more resources becoming available, either through new developers joining the project or work on SasView becoming a greater part of the day-to-day assignment for the current developers. It also should be used as a source of ideas for projects that could be undertaken with short-term resources or to meet specific requirements of stakeholders if they have the necessary resources available.

Roadmap

Every release should include bug fixes and new models as requests come in. These are assumed with each release and not included specifically below. Likewise general robustness and ease of use issues will be addressed in each release cycle.

May 2014 - release 3.0

Release 3.0 was released in the spring of 2014 after code camp II. Between code camp II and code camp III, quite a few features and bug fixes were added however no new versions were released. Version 3.0 contains over 70 models, including several customizable spherical models. Magnetic contrast with polarization analyzed data are also available for 6 models (sphere, core shell sphere, core multishell sphere, core 2nd moment, cylinder, Triaxial Ellipsoid, Parallelepiped). It supports batch mode analysis, simultaneous and constrained fitting, pair distance distribution analysis, invariant analysis, a number of linearized plots and fits, a simple model editor to create a model, a number of tools such as an SLD calculator, scattering profile from a PDB file or an image file a resolution estimator among others.

Early 2015 - Infrastructure changes and organizational work

In preparation for code camp III the repository was moved from SVN hosted on sourceforge to git on github. The website hosting was also moved to github so that updates to the master repo automatically update the site. The build version of python and wx was moved from 2.6 and 2.8 respectively to 2.7 and 3.0.2 respectively, with the wx change being the big change. As part of that change the official build machines are now being hosted at ESS in Denmark along with the official Jenkins server.

June 2015 - Release 3.1

Wrapping up the code camp III work lead to release 3.1 which includes several bug fixes including a thread problem that could cause crashing. The documentation was completely updated, and the optimization engine was replaced with the BUMPS package which provides several new options besides the default levenberg-Marquardt, including a monte carlo method, DREAM, which provides a more robust error estimation and a number of graphical outputs of parameter correlation and convergence.

March-September 2016 (post code camp IV - Delft) - Release 4.0

The major task for this release was the separation of the model calculation code from the GUI. This was done by creating a model package (*sasmodels*) in a separate repository and migating all the old models into the new framework. This project helped to significantly clean up the code base and start disentangling the computational code from the GUI code which has crept in over the years. Most importantly, it also hugely simplifies the process of implementing new models, provides the ability for users to drop in either a C or python SasView discoverable model and provides access to the built-in polydispersity functions. **This had been identified as the biggest stumbling block for further uptake by the community: it was both a frustration to current users and preventing many power users from embracing SasView more fully.**

In the process, the model code was completely refactored to both simplify and standize the writing of models, and to automaticaly enable transparent access to multiprocessing and GPU support for many users by using OpenCL. Depending on the model this could provide speedups of 10 to 100 for complex fitting finally making 2D fitting start to be practical. Macs come natively with openCL installed as do many pcs. Most PCs can be however be so enabled if they don't come that way out of the box and there are some speed gains even without GPU. SESANS infrastructure did not make it into release 4.0 but is ongoing at this point, however sasmodels can be run via scripts to fit SESANS data using BUMPS. Finally, the a plan for improving the user documentation was developed. The model documentation was moved into the model file itself thus keeping the model calculation and its documentation together, and all the model documentation was reviewed at some level and a standardized format developed though not completely implemented by release 4.0. Some cleanup to the menu bar and organization of the online documentation was begun with a plan to release it also as a PDF which would serve as a manual and a discussiong types of tutorials should be considered. Finally, with the aid of a summer student funding from ISIS, a web based "markeplace" of models was deployed which allows for uploading and sharing models.

Task Summary (Subject to the availability of sufficient resources):

- Move models to new independent sasmodels package
- Review all model documentation for accuracy
- Restructure non-model documentation and develop more holistic documentation plan
- Refactor model framework to use the new sasmodels package (keeping as much backward compatibility as possible for now)
- Enable OpenCl GPU utilization for models.
- Develop SESANS modules and scripts that demonstrate the use of sasmodels to fit SESANS data and prepare for integration into the SasView GUI
- Begin work on GUI refactoring
- Build and deploy a marketplace of models
- Usual bug fixes and other minor improvements as time and interest permit

October-December 2016 (post code camp V - SNS) - Release 4.1

The main aim for release 4.1 is to address the growing list of requests for smaller feature enhancements and improvements to the interface and workflow rather than any major structural changes. However, due to the availability of SINE2020 resources, code camp work will minimize major GUI enhancements in favor of making progress on the new GUI interface. Cleanup of projects left over from the 4.0 release such as finishing the model documentation standardization and review are also a focus of this code camp. While a complete overhaul of saving state is being built into the new GUI project, the ability to save constrained fit pages will be addressed in this release. Currently development is heavily focused on ensuring everything works on windows platforms which constitutes 60–70% of the user base. The Mac versions covering 25–30% of the user base, have not been nearly as polished something that will hopefully start to improve with this release. The GUI will provide the ability to fit SESANS data. Finally, with funding for a summer student at ISIS the old corfunc functionality from CCP13 will be integrated as a new perspective.

Task Summary (Subject to the availability of sufficient resources):

- TBA from list of tickets
- Add a new corfunc perspective
- Added file converter to support multifile data
- Integrate SESANS into the SasView GUI
- Fix Save Project when constrained fits are used
- Continue work on GUI refactoring (and the clean separation of UI from computational code.
- Work on increasing model coverage looking at non-overlap with SASfit models.
- Begin work refactoring/clean-up of Batch fitting (to include batch operations on roi such as box sum and slices).
- Finish model documentation review and formating
- Add missing documentation and documentation of new functionality

- Work on improving infrastructure (build systems, 64 bit/Anaconda on all platforms of build machines, trac, licensing, etc)

Early 2017 (post code camp VI - Grenoble) - Release 4.2

Subject to the availability of sufficient resources, release 4.2 will again focus on feature enhancements and bug fixes while continuing to work on the GUI refactoring in preparation for a release 5 at the end of the year. As part of that effort plotting requriments and design work should begin. Model work will finish pulling the SASfit models and add computation of the amplitude in preparation for implementing the beta approximation; improve code optimization in particularl adding suport for multiple GPUs; begin to transition to using adaptive integration in order to improve speed and accuracy of model fitting; and streamline the API to make scripting and pipelining more straightforward. Also will consider the need to allow users to chose the integration method as well as start working on building the beta approximation into sasmodels. On the documentation front, besides the continual updating, work should begin on a tutorial series. Finally, an effort to reach to full unit test coverage should begin as will as verifying code usage and weeding out redundant code.

Task Summary (Subject to the availability of sufficient resources):

- TBA from list of tickets
- Continue work on new GUI and code separation
- Enhancing of plot functionality - Collect requirements and begin design.
    - Identify requirments and improvements desired from current design such as tighten space, better fonts, provide both graphical and text entry of controls, put residuals on same panel, provide option to turn auto-plotting of residuals on or off etc)
    - decide on technology to use matplotlib, qtplot, pyqtgraph - currently use matplotlib which is most used but slow at times. ESRF uses it also but has added a layer to make it faster?
- Begin work to ensure all computational code has proper unit tests and that they are all being run
- Work on verifying code redundancy and weed out old/unused/obsolete code
- Begin work on tutorial series
- Update documentation
- Finish SASfit model integration
- Add adaptive integration to help users optimize fitting and explore need for refactoring to allow user choice of integration methods and begin work on sasmodels infrastructure to provid beta approximation and coherent summing of models
- Restructure sasmodels API to streamline scriptability
- Work on sasmodels code optimization for GPU and add support for multi GPU.

Mid 2017 - Organizational work

Subject to the availability of sufficient resources, a paper describing SasView will be submitted to J. Appl. Cryst. The first webinar on adding a user model (aimed at power users and instrument scientists) will be organized, while a couple of tutorials on fitting data with SasView will also be planned. Finally a first effort at identifying and documenting papers that have used SasView for their analysis will be undertaken.

Late 2017 (after code camp VII - TBA) - Release 5.0

Subject to the availability of sufficient resources, release 5.0 will move to the new QT based GUI interface. This will provide a clean API so that future GUI efforts such as web interfaces etc will be simpler to introduce while significanty improving the user experience by providing better intergration between the various parts of the GUI that have evolved and grown organically over time in response to requests. This will hopefully also address the mac vs PC usability, save state, reporting, and preferences setting which have been raised as high priority useability issues. Documentation review and creation of new documentation will continue. `In particular we will try to develop a “how to add a model and submit using pull request” tutorial as well as a code architecture manual.

Task Summary (Subject to the availability of sufficient resources):

- TBA from list of tickets at the time

- Finish refactoring new Qt based GUI and deploy including new plotting interface
- Release sasmodels 1.0 and load to PyPI
- Develop a “how to add a model and submit using a pull request” tutorial
- Begin developing use cases and design for custom workflows such as magnetic scattering, polarized beam scattering, contrast series scattering, etc.
- Work on developing a new SasView architecture manual
- Usual bug fixes and other minor improvements as time and interest permit

Early 2018 (after code camp VIII - TBA) - Release 5.1

Subject to the availability of sufficient resources, release 5.1. Work will start on refactoring fitting to allow, for example custom re-parameterization of models (e.g. replace SLD with fraction of solvent in layer), using an input array for P or S in a P*S model, fitting oriented model to 1D cut etc. Work will begin on refactoring the simultaneous/constrained fitting workflow interface and on custom workflows identified as highest priority and having a well developed design. User documentation/tutorials will be reviewed, an advanced “how to fit my data” tutorial will be started, and the architecture manual completed.

Task Summary (Subject to the availability of sufficient resources):

- Begin model fitting refactoring work to allow custom re-parameterization of models, allow reading in an array representing either PQ or SQ for P*S fits, fitting oriented model to 1D cuts including revisiting orientation definitions etc.
- Complete architecture manual
- Begin work on refactoring constrained/simultaneous fits.
- Begin work on adding custom workflows identified as highest priority and
- Begin work on advanced model fitting tutorial
- Usual bug fixes and other minor improvements as time and interest permit

Late 2018 (after code camp IX - TBA) - Release 5.2

Subject to the availability of sufficient resources, release 5.2 will provide new fitting functionality such as custom re-parameterization of models, allow reading in an array representing either PQ or SQ for P*S fits, allow print out of P and S separately during a P*S fit, fitting oriented model to 1D cut etc. The refactored workflow interfaces for constrained/simultaneous fits and batch fitting and plotting module will be deployed in this release. Work will continue on an advanced data fitting with SasView tutorial. Work on new workflow/interfaces for contrast variation for example and new magnetic scattering workflows will begin. These workflows are not expected to be in the release however.

Task Summary (Subject to the availability of sufficient resources):

- Finish fitting refactoring work to allow custom re-parameterization of models, allow reading in an array representing either PQ or SQ for P*S fits, fitting oriented model to 1D cut etc.
- Refactor simultaneous/constrained workflow interface
- Continue development of advanced fitting tutorial
- Start new workflow/interfaces
- Usual bug fixes and other minor improvements as time and interest permit

Early 2019 (after code camp X) - Release 5.3

Subject to the availability of sufficient resources, release 5.3 will again try place an emphasis on addressing requests for smaller feature enhancements and improvements to the interface and workflow. It will also include some new workflow interfaces. Use cases and design development will commence on a web interface (possibly including smartphone app capabilities). Advanced fitting tutorial and other unfinished documentation projects will be completed. Review all documentation and prioritize needs for next release.

Task Summary (Subject to the availability of sufficient resources):

- TBA from list of tickets at the time
- Finish advanced model fitting tutorial
- Include more workflow/interfaces
- Begin use case and design on Web interface (with possible smartphone app feature) - initial version can have minimal features but would be useful for demos?
- Finish outstanding documentation projects
- Prioritize new documentation tasks
- Usual bug fixes and other minor improvements as time and interest permit

Late 2019 (after code camp XI - TBA) - Release 5.y

Subject to the availability of sufficient resources, release 5.y will start providing intelligent feedback on unreasonable choices. Support for ASAXS will be added and other SAXS specific tools/workflows will be added as needed. web UI work will continue but is not expected to be ready for this release. Finally work will begin to allow computational code to run on a cluster and an intelligent launcher/scheduler design started for the GUI frontend which will make the use of the a cluster backend transparent to the user. Include documentation tasks prioritized in previous round.

Task Summary (Subject to the availability of sufficient resources):

- Start including intelligent limits/help (possibly include switch between enforcement and warning only) and explore the use of wizards in some cases
- Continue work on web UI and smartphone app
- ASAXS support added
- Add extra SAXS specific needs as appropriate
- Begin Work on getting computational code running on clusters and refactoring GUI to add an intelligent launcher/scheduler that makes the use of a cluster back end transparent to the user
- Documentation tasks as determined during previous code camp or fortnightly meetings
- Usual bug fixes and other minor improvements as time and interest permit

Early 2020 (after code camp XII) - Release 6.0

Subject to the availability of sufficient resources, release 6.0 will allow running compute intensive portions of SasView computation on a cluster back end with a transparent access from the user GUI. It will also allow deployment as a webservice with a web based front end which will have limited functionality in this first instance. Work on a smartphone app interface to the webservice will continue but lilkely will not be ready for this release. The use of wizards and intelligent user guidance will be expanded and new workflows/interfaces may be added as appropriate. Include documentation tasks prioritized in previous round.

Task Summary (Subject to the availability of sufficient resources):

- UI refactoring complete
- Deploy computational code on clusters
- GUI includes intelligent launcher/scheduler that makes the use of a cluster back end transparent to the user
- Deploy Web application
- Expand use of wizards and intelligent user guidance
- Add new workflow interfaces as appropriate
- Continue work on smartphone UI
- Documentation tasks as determined during previous code camp or fortnightly meetings
- Usual bug fixes and other minor improvements as time and interest permit

Late 2020 (after code camp XIII - TBA) - Release 6.x

Subject to the availability of sufficient resources, release 6.x will again try place an emphasis on addressing requests for smaller feature enhancements and improvements to the interface and workflow. It will also continue to expand on intelligent guidance and include more functionality on web app and see the deployment of a smartphone app. Include documentation tasks prioritized in previous round.

- TBA from list of tickets at the time
- Continue to expand use of wizards and intelligent user guidance
- Deploy smartphone app
- Expanded functionality of web app
- Documentation tasks as determined during previous code camp or fortnightly meetings
- Usual bug fixes and other minor improvements as time and interest permit

2021 - Release 6.y-z

- A 20% contingency is built into this 5 year roadmap to achieve the goals laid out.

Revision History

- 2015–11–24 : First release
- 2016–10–11 : Updated after Code Camp V discussions

---

1. http://smallangle.org/content/Software ↩
2. http://www.danse.us/ ↩
3. http://www.mantidproject.org ↩
4. http://www.ccpsas.org ↩
5. http://sine2020.eu ↩