

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 led 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.

Early 2016 (post code camp IV - Delft) - Release 4.0

The major task for this release will be the separation of the model calculation code from the GUI. This work is currently underway with the model package in a separate repository (*sasmodels*) and models being gradually migrated to the new framework. This project will significantly clean up the code base and start disentangling the computational code from the GUI code which has crept in over the years. This will enable the use of sasmodels with BUMPS in scripts to fit SESANS data and prepare for the integration of SESANS fitting into the SasView GUI. Finally, subject to the availability of sufficient resources, all the model documentation will be reviewed and the tutorial documentation will begin to be redone.

Additionally, it will enable transparent access to multiprocessing and GPU support for many users. On most macs this could provide native speedsups of 10 to 100 for complex fitting, while windows machines should see a modest speedsups but will require the installation of OpenCL by most users to take full advantage of the GPU speedup. This process will be simplified as much as possible for the user. The sasmodels package will incorporate calculation of SESANS curves from SANS models as well as native SESANS models. This will enable the use of sasmodels with BUMPS in scripts to fit SESANS data and prepare for the integration of SESANS fitting into the SasView GUI. Finally, subject to the availability of sufficient resources, all the model documentation will be reviewed and the tutorial documentation will begin to be redone.

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

- Move models to new independent sasmodels package
- Review all model documentation for accuracy
- Begin refactoring tutorial documentation to "getting started" tutorial
- Redesign model framework to use the new sasmodels package
- Enable OpenCL GPU utilization for models and work on simplifying access for all users
- Begin work on integrating SESANS into the SasView GUI
- Usual bug fixes and other minor improvements as time and interest permit

Mid 2016 - 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 2016 (after code camp V - SNS) - Release 4.w

Release 4.w will focus on addressing a growing list of requests for smaller feature enhancements and improvements to the interface and workflow rather than any major structural changes. Subject to the availability of sufficient resources, two major efforts will be to begin tackling work on refactoring the Save Project code and the Report Results code. In particular, the saving functionality will be refactored to "do what it says" including providing a true saving of the state of the entire SasView project being worked on. Again, subject to the availability of sufficient resources, other tasks will likely be smaller and prioritized prior to the code camp from the list of tickets. These would include things like fixing linearized fits, ensuring that all range settings (including zooming) can be set easily both manually (typing in numbers) and graphically, allowing models to plot beyond the fit range, allowing the general scattering calculator output to combine with a data set for comparison. We will try to also start placing a focus on improving the Mac look and feel. Currently development is 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. Finally we will continue to work on the "getting started" tutorial that replaces the current tutorial and on ensuring better coverage of doc strings for developer documentation.

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

- TBA from list of tickets at the time
- Begin Refactor of Report Results
- Begin Refactor of Save Project
- Work on Mac usability issues
- Work on new "getting started" tutorial
- Work on coverage and correctly formatted docstrings (developer documentation)

Early 2017 (after code camp VI - TBA) - Release 4.x

Subject to the availability of sufficient resources, release 4.x will focus more on infrastructure needs: Full separation of UI from computational code; Full unit testing, adding adaptive integration; Integrating SESANS into SasView GUI; and Refactoring configuration/preferences settings. Additionally, remaining work on Reporting Results and Save Project will be completed. Addition of a dispatcher will begin during code camp VI but is not expected to be completed in time for this release. As part of this effort a refactoring of the Plotting module will be considered. Adaptive integration should reduce help requests due to numerical integration problems. Also will begin studying the further need for refactoring to allow users to chose the integration method. Work will continue on improving the Mac usability. Finally, help documentation will be updated and the "getting started" tutorial will be completed.

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

- Finish UI and computation code separation
- Finish ensuring all computational code has proper unit tests and that they are all being run
- Finish "getting started" tutorial
- Revisit help documentation and update as needed.
- Integrate SESANS into GUI.
- Add adaptive integration to help users optimize fitting and explore need for refactoring to allow user choice of integration methods
- Refactor startup configuration/preferences setting to give access to configuration file more cleanly
- Begin work on dispatcher
- Consider the need to refactor plotting modules and begin work as appropriate
- Finish work on refactor of Saving Project and Reporting Results
- Continue focus on mac platform
- Usual bug fixes and other minor improvements as time and interest permit

Late 2017 (after code camp VII - TBA) - Release 4.y

Subject to the availability of sufficient resources, release 4.y will provide a dispatcher module in the UI which will be used to enhance the user experience by better integrating the fit panels with their respective plots and with the batch mode. Otherwise this release will focus again on addressing the number of smaller feature requests and GUI and workflow enhancements. Documentation review and creation of new documentation will continue. In particular we will try to develop a "how to add a C 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
- Dispatcher module added to allow tighter integration between related panels
- Finish refactoring of Plot modules
- Develop a "how to add a C model and submit using a pull request" tutorial
- Work on developing a new SasView architecture manual
- Enhancing of plot functionality
 - 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?
- Usual bug fixes and other minor improvements as time and interest permit

Early 2018 (after code camp VIII - TBA) - Release 4.z

Subject to the availability of sufficient resources, release 4.z will see the completion and deployment of the report refactoring while the remaining models which do not support magnetic contrast will have that added. 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, outputting P and S separately in such a fit, fitting oriented model to 1D cut etc. Work will begin on refactoring the simultaneous/constrained fitting workflow interface and refactoring/cleaning-up the batch fitting workflow to include batch processing of for example ROI (box sum or slices) etc. Work will also begin on possible specialized workflows for magnetic scattering. 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):

- Add support for magnetic contrast to remaining models
- Complete architecture manual
- Begin work on refactoring constrained/simultaneous fits.
- Begin work refactoring/clean-up of Batch fitting (to include batch operations on roi such as box sum and slices).
- Begin model fitting refactoring work to allow custom re-parameterization of models, allow reading in an array representing either PQ or SQ for *PS fits, allow print out of P and S separately during a PS fit*, fitting oriented model to 1D cuts including revisiting orientation definitions etc.
- Begin work on adding possible custom workflows for magnetic scattering – NIST and ESS, possible ISIS and ILL needs? talk to Albrecht Wiedenmann
- 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 5.0

Subject to the availability of sufficient resources, release 5.0 will provide new fitting functionality such as custom re-parameterization of models, allow reading in an array representing either PQ or SQ for *PS fits, allow print out of P and S separately during a PS 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 *PS fits, allow print out of P and S separately during a PS fit*, fitting oriented model to 1D cut etc.
- Refactor simultaneous/constrained workflow interface
- Refactor/clean-up batch workflow/interface (should also works with other tools such as roi slices/box sums etc)
- Continue development of advanced fitting tutorial
- Start new workflow/interfaces (e.g. contrast variation etc)
- Start adding new workflows for magnetic scattering as appropriate
- Usual bug fixes and other minor improvements as time and interest permit

Early 2019 (after code camp X) - Release 5.x

Subject to the availability of sufficient resources, release 5.x 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. Finally , refactoring of the user interface will begin with an examination of new technologies and a new toolbox may be chosen. Current contenders seem to include: wx, qt, django, D3 and genap. New interface should probably support web application as well as local operations. Further a smartphone web app tool to interface with web will be considered. Finish advanced fitting tutorial and other unfinished documentation projects. 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 new workflow/interfaces (e.g. magnetic scattering, contrast variation, etc)
- Begin UI refactor work - agree on technology and designs
 - Work on Web interface - initial version can have minimal features but would be useful for demos?
 - Begin work on smartphone app web interface
- 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. UI refactoring 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 UI Refactoring including web 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 likely 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.

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> →