

SINE2020 and SasView Roadmap

Work on SasView modernization

DMSC

www.europeanspallationsource.se

March 2016

- Two positions working full time on SINE2020 between Jan 1, 2016 until Dec 31, 2017:
Wojtek and Piotr

SasView Roadmap vs. SINE2020 goals

- Roadmap

Post-CCIV, 4.0

- Move models to new independent Sasmodels package
- Enable OpenCL GPU utilization for models
- Separation of the model calculation code from the GUI
- Begin work on integrating SESANS into the SasView GUI

Post-CCV, 4.w

- Results reporting refactoring
- Project save refactoring

Post-CCVI, 4.x

- Finish UI and code separation + testing
- Preferences/startup config refactoring

Post-CCX, 5.x

- UI refactor work

- SINE2020 (2016-2017)

- Code modularization
- New API
- New GUI
- Optimization of algorithms for real time analysis
- Extension with SASFit models

SasView Roadmap vs. SINE2020 goals

- Roadmap

Post-CCIV, 4.0

- Move models to new independent Sasmodels package
- Enable OpenCL GPU utilization for models
- Separation of the model calculation code from the GUI
- Begin work on integrating SESANS into the SasView GUI

Post-CCV, 4.w

- Results reporting refactoring
- Project save refactoring

Post-CCVI, 4.x

- Finish UI and code separation + testing
- Preferences/startup config refactoring

Post-CCX, 5.x

- UI refactor work

- SINE2020 (2016-2017)

- Code modularization
- New API
- New GUI
- Optimization of algorithms for real time analysis
- Extension with SASFit models



SasView Roadmap vs. SINE2020 goals

- Roadmap

Post-CCIV, 4.0

- Move models to new independent Sasmodels package
- Enable OpenCL GPU utilization for models
- Separation of the model calculation code from the GUI
- Begin work on integrating SESANS into the SasView GUI

Post-CCV, 4.w

- Results reporting refactoring
- Project save refactoring

Post-CCVI, 4.x

- Finish UI and code separation + testing
- Preferences/startup config refactoring

Post-CCX, 5.x

- UI remake work

- SINE2020 (2016-2017)

- Code modularization

- **New API**

- New GUI

- Optimization of algorithms for real time analysis

- Extension with SASFit models

SasView Roadmap vs. SINE2020 goals

- Roadmap

Post-CCIV, 4.0

- Move models to new independent Sasmodels package
- Enable OpenCL GPU utilization for models
- Separation of the model calculation code from the GUI
- Begin work on integrating SESANS into the SasView GUI

Post-CCV, 4.w

- Results reporting refactoring
- **Project save refactoring**

Post-CCVI, 4.x

- Finish UI and code separation + testing
- **Preferences/startup config refactoring**

Post-CCX, 5.x

- **UI refactor work**

- SINE2020 (2016-2017)

- Code modularization

- New API

- **New GUI**

- Optimization of algorithms for real time analysis

- Extension with SASFit models

SasView Roadmap vs. SINE2020 goals

- Roadmap

Post-CCIV, 4.0

- Move models to new independent Sasmodels package
- **Enable OpenCL GPU utilization for models**
- Separation of the model calculation code from the GUI
- Begin work on integrating SESANS into the SasView GUI

Post-CCV, 4.w

- Results reporting refactoring
- Project save refactoring

Post-CCVI, 4.x

- Finish UI and code separation + testing
- Preferences/startup config refactoring

Post-CCX, 5.x

- UI refactor work

- SINE2020 (2016-2017)

- Code modularization

- New API

- New GUI

- **Optimization of algorithms for real time analysis**

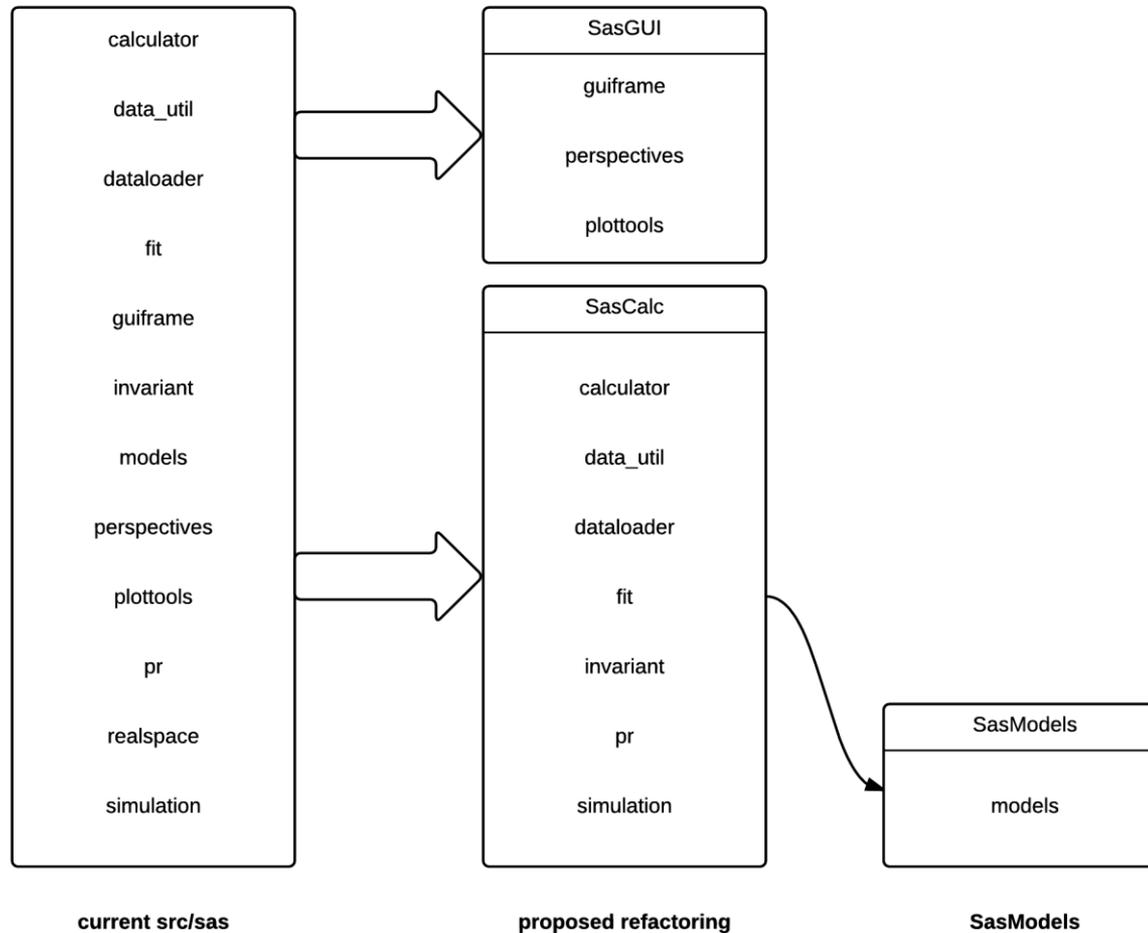
- Extension with SASFit models

Code refactoring

- Separate calculation from GUI
 - SasCalc can be used as a standalone module
 - No SasGUI dependencies in SasCalc
 - No SasModels dependencies in SasGUI
- Need to define how to use SasCalc modules (agree on API)
- Write example scripts and API documentation

Proposed refactoring

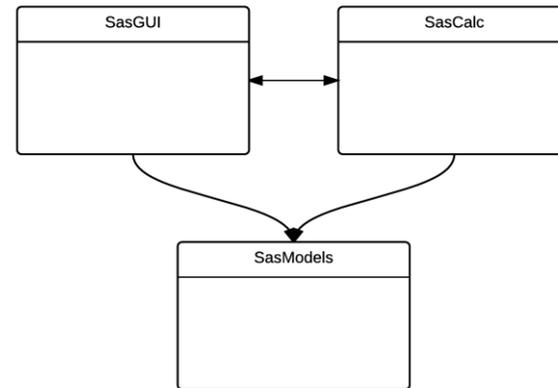
geographical changes



Module dependencies

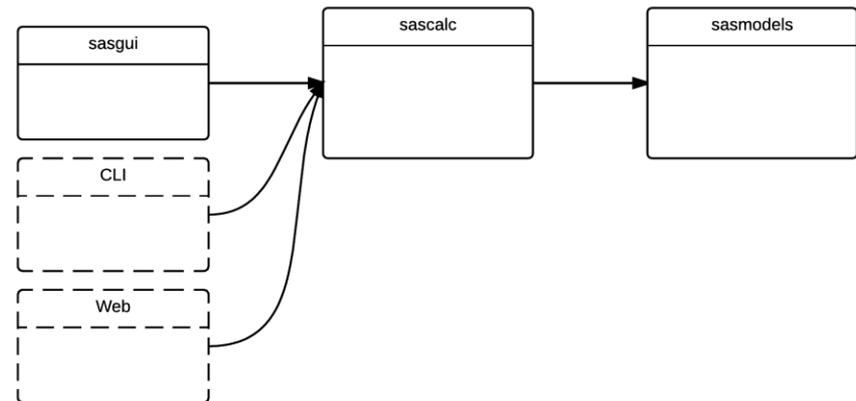
- **Currently :**

- SasCalc depends on some methods in SasGUI
- SasModels is used by SasGUI

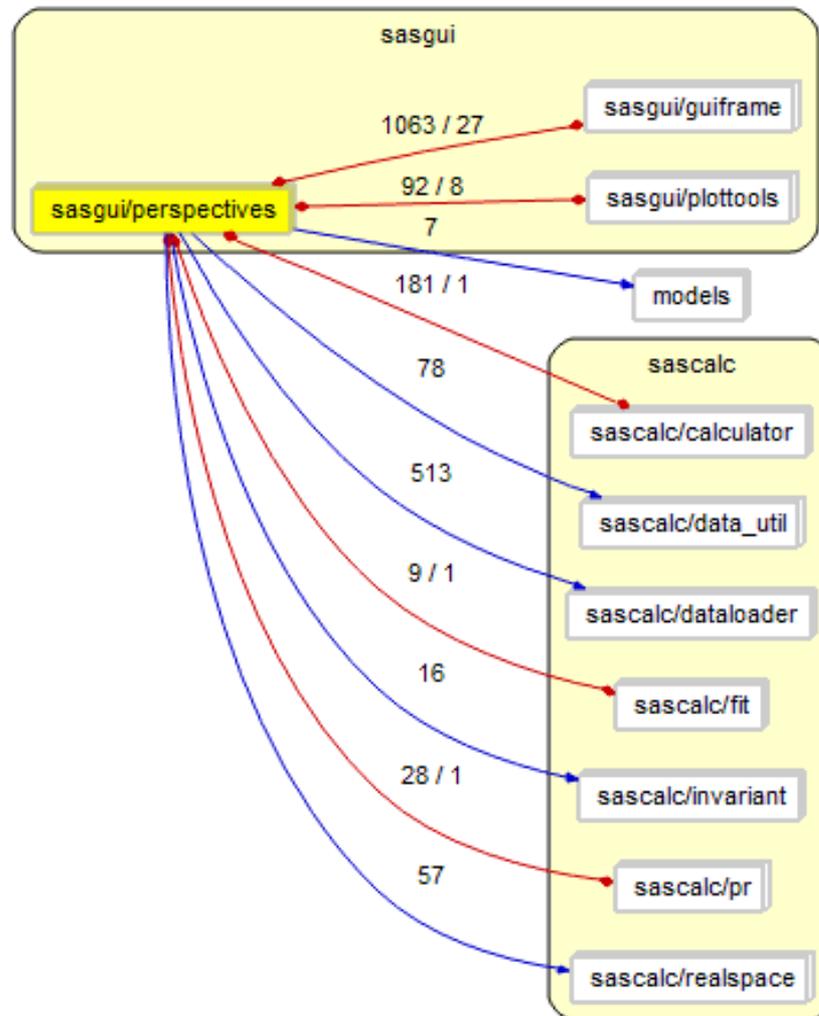


- **Proposed:**

- SasCalc dependent only on SasModels
- SasView using SasCalc methods exclusively
- CLI access to all the SasCalc functionality and models



Module dependencies



Using SasCalc as a module

```
from sas.sascal.dataloader.loader import Loader
from sas.sascal.pr.invertor import Invertor
```

```
loader = Loader()
test_data = loader.load("sphere_80.txt")
```

```
pr = Invertor()
```

```
# Set data
```

```
pr.x = test_data.x
```

```
pr.y = test_data.y
```

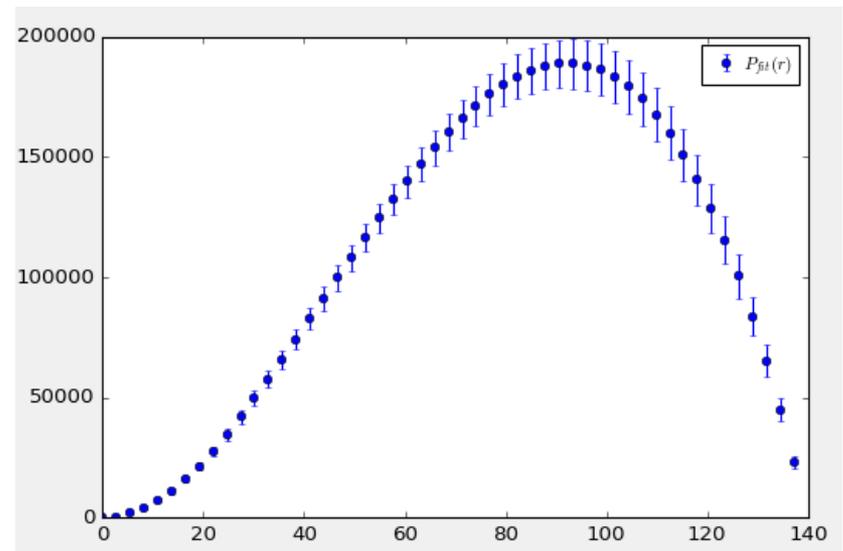
```
# Perform inversion and show graph
```

```
x, y = pr.invert()
```

```
import matplotlib.pyplot as plt
```

```
plt.plot(x, y)
```

```
plt.show()
```



Need to (re)define how the calculators are to be used – agree on API for them.

GUI modernization

- With SasGUI refactored it is possible to work exclusively on the GUI part of SasView
- Proposed rewrite using PyQt
 - Platform consistency - dialogs look and behave the same across all platforms
 - Professional (more mature) technology
 - Long term maintainability
 - Ease of development (Qt designer)
 - Clean separation of UI and code
 - Signals connected to slots automatically
 - Inherent MVC pattern in Qt simplifying data management
 - Native thread support
 - SINE2020 requirement for interoperability with other Qt based codes (Mantid, BornAgain)

Quick dialog prototyping

Dialog - [Preview]

Selection Options
Select all

Data
Load
Delete

Send to Fitting Batch mode

Theory
Freeze

Plot
New
Append to Graph1

Dialog - [Preview]

I(q) data source
Name:

Total Q range
Min: Max: \AA^{-1}

Customized input
Background: cm^{-1} Scale:
Contrast: \AA^{-2} Porod constant: $(\text{cm } \text{\AA}^3)^{-1}$

Extrapolation
Q Range: Min Max \AA^{-1}

Low Q
 Enable Low-Q extrapolation
Npts:
 Guinier Fit
 Power law Fix
Power:

High Q
 Enable High-Q extrapolation
Npts:
 Fit
 Fix
Power:

Output
Volume fraction: +/-
Specific Surface: +/- \AA^{-1}
Invariant Total [Q]: +/- $(\text{cm } \text{\AA}^3)^{-1}$

Calculate Status Help

Dialog - [Preview]

I(q) data source
Name:

Estimate background level

Parameters
Slit parameters
Height Height \AA^{-1}

GroupBox
Q min Qmax \AA^{-1}

Suggested value
Number of terms 10
Regularization constant 1e-07
Max distance (A) Explore

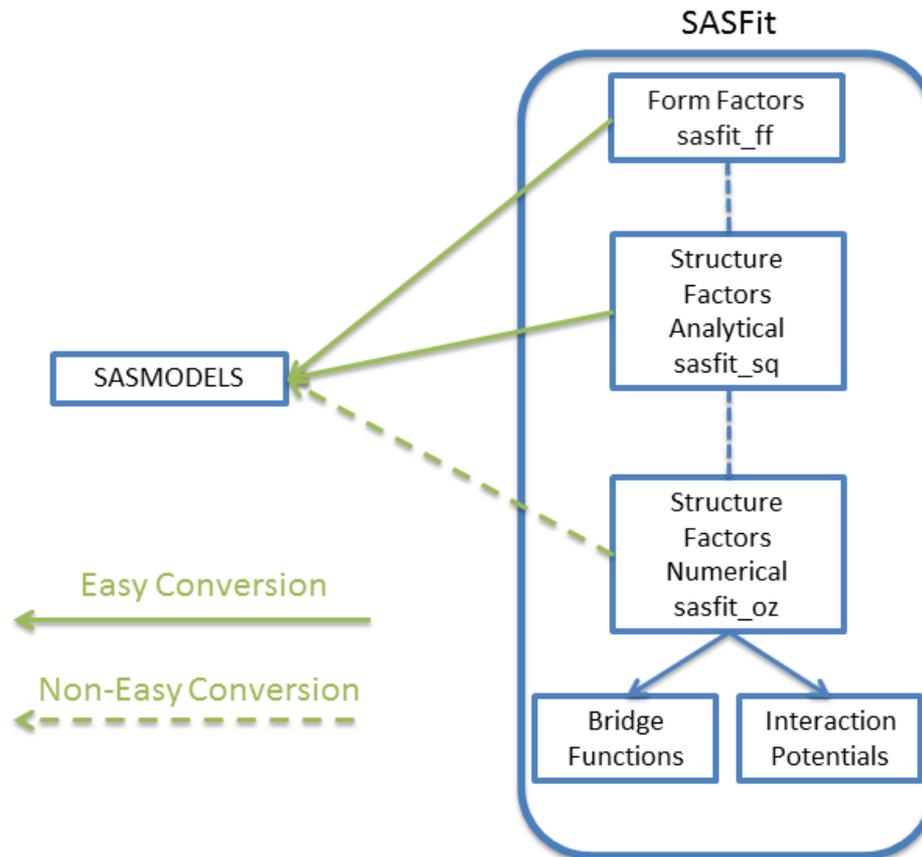
Output
 R_g \AA
I(Q=0) \AA^{-1}
Background \AA^{-1}
Calculation time sec
 χ^2/dof
Oscillations
Positive fraction
1- σ positive fraction

Calculate Help

SasFit integration

- Idea – reuse a large set of SasFit fitting functions in SasModels
- Create setup similar for SasModel – conversion Wiki, compare.sh script, etc.
- Investigate possible automation of the conversion or its parts
- Examine SasFit structure factor methods with the OZ equation
- In collaboration with PSI

SasFit integration



Proposed timeframe

