SNS will have significant impact in the area of structure determination as it will open up neutron diffraction to a large community used to dealing with x-rays. High flux will allow x-ray sized samples to be run and high throughput will allow better access to the beam. To realize these gains, scientists need to be able to do science in real, or almost real-time. The Dream Desktop concept is designed to allow scientists to refine and visualize structures in real-time as data are being collected. This changes the neutron beamtime from a measurement to an experiment where experimental strategy is modified in real-time based on the fit results.

This slide addresses a broader impact of the Diffraction sub-project of DANSE.

The x-ray diffraction user community is one to two orders of magnitude larger than the neutron user community, largely due to the easy accessibility of x-rays with the ubiquity of in-house sources. Many structural problems are too complex for these sources however, and would benefit from neutron experiments, but never make it to a neutron source because of various entry barriers such as the need to make large samples, and getting access to the beam. SNS presents an opportunity to significantly reduce these barriers and therefore to significantly broaden participation in neutron scattering. Software can play a key role in this. Friendly, science oriented, user interfaces will reduce the learning curve for neutron diffraction. Distributed software that can be run from the researcher’s local institution will facilitate remote access. Real-time analysis and feedback from the modelling and data acquisition systems would improve the efficiency of beam use allowing more samples to be run. Good software and user interfaces also decreases the learning curve of new users such as beginning graduate students, enabling education.

The graphic shown here is a screen-shot of a Dream Desktop application showing how an experimenter could interact with the data quickly and effectively to do structure science (rather than “neutron scattering”) in quasi-real time. The name “Dream Desktop” came from the original concept that was presented at the first DANSE reverse site visit at the NSF in May of 2004: this is the desktop that an experimenter might dream about. However, what is shown here is a real, working implementation that was put together in the space of a few months as part of the DANSE Design grant. It illustrates a few aspects of the DANSE project: (a) rapid application development by reusing legacy software (b) interoperability of diverse software components through the use of Python and the Pyre interface. The software running underneath the GUI includes a home-written c++ PDF modelling program (PDFFIT2) an open source graphics library (Matplotlib), and an open source molecular visualizer (PyMol). The Dream Desktop project incorporates other aspects of DANSE, and these are described, along with the code design, in more detail on the next slide.

We developed the Dream Desktop application to prototype and develop the software construction process that we will use in the construction phase of the project, and as a demonstration project to help us get to grips with application development in a modular software environment and in a geographically distributed collaboration. However, the resulting application is so successful that students and post-docs in the Billinge group are anxious to use it for their current modelling tasks. As well as providing a useful tool for the students, this will allow us to complete the development cycle with user testing and feedback and release management. We expect a few interesting scientific articles to appear in the process too!
DANSE Diffraction sub-project: 
Dream Desktop for Neutron Diffraction 
Simon J. L. Billinge, Michigan State University 
DMR-0412074 

The object oriented design of the Dream Desktop package with the development team superimposed.

This slide shows the a UML component diagram of the Dream Desktop design, with pictures of the developers superimposed and their responsibilities within the group.

In addition to the factors discussed on the previous viewgraph, the following aspects of the DANSE code design philosophy are highlighted in this application:

- The modular design ensures that the application is extendable and maintainable.
- The program is distributed: the modelling code is run on a different computer so it doesn’t compete for system resources with the visualization and plotting programs.
- The user interface (GUI) is separated from the other programs so the GUI can be redesigned, or there can be different GUIs, on top of the same underlying engine.
- Design: The modular design allows different programs to interoperate effectively. It also allows for maintainability and extensibility. For example, we could replace Pymol with another structure visualization program at a later date. This would be fairly straightforwardly accomplished by wrapping the new program with the same generic interface, or by updating PymolControl, the python program that controls Pymol. The rest of the program should not have to be modified.
- The program was implemented with PDFFIT running on a remote computer but updating the plots on the local computer in real-time. This allows the large-scale modelling runs to be run without competing for system resources with the 2-D plotting and 3-D visualization programs that are running at the same time.

Personnel: 
The developers include people from physical and mathematical science backgrounds and with varying levels of programming expertise. Modular development allows effective teamwork with strong interactions between the group members that leads to creative problem solving and excellent opportunities for education and personal development in the team.

The dynamics within the group turned out to be excellent with individuals working on their distinct modules and frequent meetings to discuss how to consolidate the modules and what the functionality and timeline should be. There was significant discussion and exchange of problem solving strategies within the group leading to high morale; much better than my past experience where individual grad students worked on their own codes to do specific tasks. Interactions between the DANSE diffraction sub-group, DANSE central services and the other subgroups also went well. Breakout teleconference sessions where developments were discussed between the subset of people in the wider project who were interested in those developments, were particularly valuable.