IMR-MIP: DANSE

Distributed Data Analysis for Neutron Scattering Experiments CNST

Brent Fultz*, Michael A.G. Aivazis*, Ian S. Anderson‡

* California Institute of Technology, Pasadena CA 91125 USA
‡ Spallation Neutron Source, Oak Ridge TN 37831-6474 USA
btf@caltech.edu aivazis@caltech.edu andersonian@sns.gov

with

S.J.L. Billinge, R.M. Briber, P.D. Butler,
P.A. Kienzle, T. Swain, F.R. Trouw, E. Üstündag

Keywords:
DANSE, neutron scattering, software, data analysis, Spallation Neutron Source, pyre, distributed computing, grid, experiments, modeling, simulations
§ Software and the Spallation Neutron Source

The Spallation Neutron Source (SNS), under construction in Oak Ridge, Tennessee with a budget of B$ 1.411, is the world’s largest science construction project. In 2006 it will begin operations, and in 2009 it will produce intense beams of neutrons for studies of materials and condensed matter. The instruments that control these beams, and detect neutrons scattered from the specimens, are state-of-the-art. Neutron scattering experiments performed at the SNS will produce data of unprecedented detail on the positions and motions of atoms and spins in materials.

To date, however, there is no project to develop new software to analyze the data from the SNS. There exist today many individual software programs for data analysis, and the SNS is engaged in an international effort to provide a runtime platform for these codes, and access to them by remote login. Unfortunately, these existing codes are not mutually compatible, and few were written professionally. More importantly, there is a need for software that enables the new types of science that are possible with modern neutron instruments. We propose to develop this software in a system called DANSE – distributed data analysis for neutron scattering experiments. Through prior software development work, including an IMR-MIP CED design proposal, much of the framework for DANSE has been built and some of its scientific capabilities have been tested. Nevertheless DANSE is a large effort, and we propose a five-year construction project under the NSF IMR-MIP Program to build a new system for computational neutron science.

The DANSE project would include two parts. The first is a software engineering effort to complete a framework to support the interoperability of modular software components, and distributed grid-style computing. The second is an effort by scientists to develop software modules for the different subfields of neutron scattering research. The DANSE system would provide a complete, open-source solution to the data analysis needs of the SNS because it includes a common framework for all data analyses, and covers all major subfields of neutron scattering science.

§ Software Engineering

Data analysis will be performed with reusable software components that are integrated into a coherent software framework that manages the life cycles of components and their data exchanges. The DANSE framework would allow the same application to operate on an isolated laptop without access to the internet, or on high-end facilities in a computational grid. The framework uses the open source interpretive environment provided by the Python language, which allows virtually unlimited flexibility and extensibility. As a result, the framework clears the path for the forward migration of legacy codes and their mutation into modern, object-oriented, maintainable computational engines. The framework enables high-performance computing on distributed resources in a nearly transparent manner. It provides an unprecedented opportunity to merge data analysis, theory and simulation into a uniform computing environment.

The diversity of the user community makes it imperative that both novice and expert users are comfortable using the same system of software. A greater challenge, but one considered in detail, is to enable users of all levels of programming sophistication to contribute new capabilities to the system. Accessibility to the data analysis software is provided at four levels of increasing complexity:

− With the click of the mouse, users can select from a menu of debugged and tested scripts that perform specific data transformations.
− New scripts can be composed from existing ones in a visual programming environment by dragging components on a canvas and wiring them together (e.g. Fig. 2), as is done with the commercial applications Labview or IRIS Explorer, for example.
− Alternatively, new scripts and analysis codes can be written in Python, with an entry barrier as low as possible with any programming language.
− For optimal performance of computationally-demanding analyses, code written in low level languages such as C++ or Fortran can be integrated fairly easily into the framework as the cores of Python components. This process will be automated further.

The development of this framework would be centered at Caltech in an activity led by Michael A.G. Aivazis.
§ Science Applications

It is necessary for data analysis software to be developed by scientists who work with real data. Neutron scattering research is organized into subfields specializing in different types of science, approximately clustered around instruments that perform particular types of measurements. To best accommodate the needs of these subfields, the development of scientific software would be organized with a similar structure. Science subprojects are proposed in these subfields:

1. Diffraction. This method has the largest user community. Experiments include studies of crystal structure and microstructure, both on liquids and amorphous materials, polycrystalline materials, and single crystals. S.J.L. Billinge, leader.

2. Engineering Diffraction. Research in this field includes measurements and interpretations of internal strains in materials, and studies of crystalline textures in polycrystalline materials. E. Üstündag, leader.

3. Small-Angle Neutron Scattering (SANS). Users in this field have interests that span from biochemistry to solid-state magnetism. SANS research includes a large activity in polymer structure, and the structural evolution of polymers under temperature and flow. P.D. Butler, leader.

4. Reflectometry. This method measures the depth profile of neutron scattering near a surface. The science includes structures of large molecules at surfaces and interfaces, and surface magnetism probed with polarized neutrons. P.A. Kienzle, leader.

5. Inelastic scattering. Dynamical processes such as the elementary excitations of phonons and magnons in solids, and vibrations and motions of molecules are studied by inelastic scattering. B. Fultz, leader.

The leaders of these five subprojects are specialists in these fields of neutron science who have had experience with software development. They will organize the data analysis procedures of today, making neutron scattering research easier for new users. A more exciting goal is to develop software that enables new types of science. These opportunities are different in each science subfield, but often involve leveraging the remarkable developments in computational materials science that have occurred over the past decade. With the help of a project manager, Brent Fultz will coordinate these subprojects, and ensure that the software engineering at Caltech is compatible with their needs.

§ NSF Review Criteria

1. Intellectual merit. The intellectual merit would differ from other NSF-funded university-based projects because DANSE would be a construction project. Primarily, the DANSE software system will facilitate scientific interpretations of data acquired from the instruments at the Spallation Neutron Source, scheduled for operations in 2006 with an active user program in 2008. The DANSE system, or its parts, would be useful for other neutron sources too. DANSE will offer ease of use to scientists who are new to neutron scattering research. More significantly, data analysis will be possible with much higher sophistication than has been possible to date. DANSE will elevate the science of neutron scattering research, increasing its potential for impact.

2. Broader impact. The DANSE project is helping to organize the neutron scattering science community in the U.S., and has generated worldwide interest. DANSE is a natural application for grid-based computing, and the layered design of the DANSE framework was planned for migration to the TeraGrid, or a similar future cyberinfrastructure. The DANSE framework could be adapted for data analysis in other fields of science. An outreach effort has been planned as a collaboration with education professionals at Iowa State University.
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Summary</td>
<td>i</td>
</tr>
<tr>
<td>§ Software and the Spallation Neutron Source</td>
<td>i</td>
</tr>
<tr>
<td>§ Software Engineering</td>
<td>i</td>
</tr>
<tr>
<td>§ Science Applications</td>
<td>ii</td>
</tr>
<tr>
<td>§ NSF Review Criteria</td>
<td>ii</td>
</tr>
<tr>
<td>1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>2 Vision and Goals</td>
<td>2</td>
</tr>
<tr>
<td>§ Goals</td>
<td>2</td>
</tr>
<tr>
<td>§ A Software Component Framework</td>
<td>2</td>
</tr>
<tr>
<td>§ Enabling Neutron Science with DANSE</td>
<td>4</td>
</tr>
<tr>
<td>§ Outreach</td>
<td>5</td>
</tr>
<tr>
<td>3 Project Infrastructure</td>
<td>5</td>
</tr>
<tr>
<td>§ Structure of This Proposal</td>
<td>5</td>
</tr>
<tr>
<td>3.1 Computing Resources</td>
<td>5</td>
</tr>
<tr>
<td>3.2 Configuration and Release Management</td>
<td>6</td>
</tr>
<tr>
<td>3.3 Personnel and Other Resources</td>
<td>8</td>
</tr>
<tr>
<td>4 Central Services: The Software Engineering Effort</td>
<td>8</td>
</tr>
<tr>
<td>4.1 Distributed Component Framework</td>
<td>8</td>
</tr>
<tr>
<td>4.2 Visualization</td>
<td>12</td>
</tr>
<tr>
<td>4.3 User Interfaces</td>
<td>14</td>
</tr>
<tr>
<td>4.4 User Layer Data Services</td>
<td>16</td>
</tr>
<tr>
<td>4.5 Personnel and Other Resources</td>
<td>17</td>
</tr>
<tr>
<td>5 Common Tasks for Scientific Subprojects</td>
<td>17</td>
</tr>
<tr>
<td>§ Science Subfields</td>
<td>17</td>
</tr>
<tr>
<td>§ Open Source and Commercial Software</td>
<td>18</td>
</tr>
<tr>
<td>5.1 Data Reduction</td>
<td>18</td>
</tr>
<tr>
<td>5.2 Modeling of Data</td>
<td>18</td>
</tr>
<tr>
<td>5.3 Full Simulations of Data</td>
<td>19</td>
</tr>
<tr>
<td>5.4 Common Algorithms and Data Types</td>
<td>21</td>
</tr>
<tr>
<td>5.5 Personnel and Other Resources</td>
<td>22</td>
</tr>
<tr>
<td>6 Diffraction</td>
<td>22</td>
</tr>
<tr>
<td>§ Scope of Science for Diffraction</td>
<td>22</td>
</tr>
<tr>
<td>§ Neutron Diffraction Capabilities at the SNS</td>
<td>23</td>
</tr>
<tr>
<td>6.1 Data Reduction</td>
<td>23</td>
</tr>
<tr>
<td>6.2 Modeling and Co-Refinement</td>
<td>23</td>
</tr>
<tr>
<td>6.3 Direct Experiment Simulation</td>
<td>25</td>
</tr>
<tr>
<td>6.4 Personnel and Other Resources</td>
<td>26</td>
</tr>
<tr>
<td>7 Engineering Diffraction</td>
<td>26</td>
</tr>
<tr>
<td>Scope of Research</td>
<td>26</td>
</tr>
<tr>
<td>7.1 Experimental Setup and Data Reduction</td>
<td>27</td>
</tr>
<tr>
<td>7.2 Mechanics Modeling</td>
<td>27</td>
</tr>
<tr>
<td>7.3 Experiment Simulation</td>
<td>28</td>
</tr>
<tr>
<td>7.4 Subproject Personnel</td>
<td>29</td>
</tr>
<tr>
<td>8 Small-Angle Neutron Scattering</td>
<td>29</td>
</tr>
<tr>
<td>§ Scope of Science for SANS</td>
<td>29</td>
</tr>
<tr>
<td>8.1 Data Reduction</td>
<td>29</td>
</tr>
<tr>
<td>8.2 Analysis</td>
<td>30</td>
</tr>
<tr>
<td>8.3 Simulations and Other Advanced Analysis Tools</td>
<td>31</td>
</tr>
<tr>
<td>8.4 Integrated User Interface</td>
<td>32</td>
</tr>
<tr>
<td>8.5 Personnel and Other Resources</td>
<td>32</td>
</tr>
<tr>
<td>9 Neutron Reflectometry</td>
<td>33</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>Overview</td>
<td>33</td>
</tr>
<tr>
<td>9.1 Data Reduction</td>
<td>33</td>
</tr>
<tr>
<td>9.2 Analysis</td>
<td>33</td>
</tr>
<tr>
<td>9.3 Direct Experiment Simulation</td>
<td>34</td>
</tr>
<tr>
<td>9.4 Personnel and Other Resources</td>
<td>35</td>
</tr>
<tr>
<td>Inelastic Neutron Scattering</td>
<td>35</td>
</tr>
<tr>
<td>10.1 Data Reduction</td>
<td>36</td>
</tr>
<tr>
<td>10.2 Modeling of Dynamical Processes in Solids</td>
<td>36</td>
</tr>
<tr>
<td>10.3 Direct Experiment Simulation</td>
<td>37</td>
</tr>
<tr>
<td>10.4 Personnel and Other Resources</td>
<td>38</td>
</tr>
<tr>
<td>Education, Outreach, and Knowledge Transfer</td>
<td>38</td>
</tr>
<tr>
<td>11.1 Textbook Content</td>
<td>38</td>
</tr>
<tr>
<td>11.2 K-12 Education</td>
<td>39</td>
</tr>
<tr>
<td>11.3 Undergraduate Education</td>
<td>39</td>
</tr>
<tr>
<td>11.4 Graduate Education</td>
<td>40</td>
</tr>
<tr>
<td>11.5 Meetings, Workshops, and Participant Support</td>
<td>40</td>
</tr>
<tr>
<td>11.6 Collaboration with other Software Efforts</td>
<td>40</td>
</tr>
<tr>
<td>11.7 International Outreach</td>
<td>41</td>
</tr>
<tr>
<td>Project Plan</td>
<td>41</td>
</tr>
<tr>
<td>12.1 Operational Goals</td>
<td>41</td>
</tr>
<tr>
<td>12.2 Planning Process</td>
<td>41</td>
</tr>
<tr>
<td>12.3 Summary of Project Plan</td>
<td>42</td>
</tr>
<tr>
<td>Project Management</td>
<td>44</td>
</tr>
<tr>
<td>13.1 Management Organization</td>
<td>44</td>
</tr>
<tr>
<td>13.2 Project Controls</td>
<td>46</td>
</tr>
<tr>
<td>13.3 Transition to SNS Operations</td>
<td>48</td>
</tr>
<tr>
<td>Summary</td>
<td>50</td>
</tr>
<tr>
<td>References</td>
<td>51</td>
</tr>
</tbody>
</table>
1. Introduction

Neutron scattering is used to study the structure and dynamics of materials, molecules, and condensed matter. Most of what we know about the positions and motions of atoms in materials has been learned from scattering experiments. In diffraction experiments, the wavefunction of a neutron is scattered by atoms to produce constructive wave interferences that give information about the positions of atoms [1, 2]. Neutron diffraction is an important method for studying the spatial arrangements of both atoms and electron spins. Neutron scattering plays an even larger role in research on the dynamical processes in materials, such as the vibrations of atoms or spin structures.

Neutron scattering research in the United States is poised for a spectacular leap after the Spallation Neutron Source (SNS) is commissioned in 2006. With the power of its source and the performance of its instruments, the SNS will improve measurement efficiencies by typically two orders of magnitude. The SNS will elevate the interest in neutron scattering in the U.S, and many scientists will make a tentative start with this method to assess its value for their own research. Analyzing data is an essential, but non-trivial, step in neutron scattering research, and for these new users the analysis should be as simple and as robust as possible. Nevertheless, the system for data analysis must also be conveniently configurable by more sophisticated users, who should leverage new capabilities of scientific computing to advance the science of neutron scattering.

Starting in early 2002, Brent Fultz and Michael Aivazis had numerous discussions with staff at the SNS and other neutron scattering scientists concerning software for data analysis. The potential of distributed computing was demonstrated at nine workshops and reviews\(^1\), and a software component framework that supports the interoperation of a diverse set of software packages was released. The results of seven email polls\(^2\) and surveys were also analyzed. From these workshops, meetings, and polls, we learned that inadequate software, inconsistent in its user interface, incomplete in its function, and worse yet inconsistent in performance, is the major source of frustration for users of neutron scattering instruments. Some of these inconsistencies and impediments are not large individually, but their cumulative effect can cause neutron scattering experiments to be inefficient or even unproductive. These frustrations originate from the fragmented and incomplete state of today’s software. Data analysis is performed by specialized programs because each was developed to solve a specific problem of immediate interest to its author. For subsequent users, this usually means that even if appropriate programs and data are available readily, the programs do not interoperate, do not run under the same operating system, and have entirely different user interfaces. Users are forced to perform tedious translations and transportations of data, often before knowing if the different programs are based on compatible data structures or scientific assumptions. Finally, data visualization is generally left to the users, so many gravitate towards commercial data analysis packages. These commercial packages, especially IDL or Matlab, have separate communities of ardent supporters because they provide integrated environments for data manipulations and visualization. Unfortunately, software developed within one of these environments cannot be reused in the other, and these codes are often confined to personal computers without access to high-performance computing.

The proposed DANSE system would overcome many frustrations of computational neutron science by providing users a seamless access to high-performance computing resources, configurable procedures for data analysis, and a software framework that does not take away a user’s favorite tools. The specific tasks in this proposal were selected to satisfy the requests from a broad range of neutron scientists in workshops, surveys and polls. The proposed DANSE system would meet the standards, and provide much of the functionality, described in the document “SNS Data Analysis Systems Functional Requirements and Desired Capabilities” (SNS-IS-107020000-TD0001-R00).


\(^2\) A large (73 respondents) poll was conducted by the Neutron Scattering Society of America in late 2002. Many other polls are described in Section 12 of this proposal.
2. Vision and Goals

§ Goals
We propose to build a software system for distributed data analysis for neutron scattering experiments (DANSE). In essence, DANSE will be a software system for doing the computations needed in experimental neutron science, integrated with software for doing computational neutron science. Users could run DANSE locally on their personal computers, or use DANSE to build distributed networks with high-performance remote resources.

The DANSE system will provide to all subfields of neutron scattering research the data analysis tools of today, and a core set of components for new types of analysis based on recent developments in materials theory. At its application layer, the software components for scientific computation are modular, and can be arranged flexibly through different user interfaces to accommodate different needs and personal preferences. Below the application layer, DANSE is readily extensible so that working scientists can efficiently transform their existing computer programs into components of the system. At the framework layer, the distributed capability of DANSE will make it possible to use high-performance computing resources, enabling new types of neutron science.

The construction of DANSE involves two efforts, an effort to adapt a software component framework to the needs of computational neutron scattering research, and an effort to build or adapt scientific software components to run on this framework. During its construction phase, DANSE will engage a diverse group of personnel mixing graduate students with software professionals, staff from national laboratories and universities, and collaborators from around the world. An academic emphasis is placed on documentation, and outreach will be coordinated professionally. SNS personnel will be involved in the construction project, all the way through the level of Co-Principal Investigator, ensuring that the DANSE system will be used successfully during the operations phase of the SNS.

§ A Software Component Framework
The value of a software component framework can be understood by analogy to an operating system, but at a higher level of abstraction. Without an operating system, it is possible to boot a computer into the entry point in memory for an application program. The application program would need to include input/output services so that output could be saved to a disk file. It would then be possible to reboot the computer to run a second application that uses the saved results. Today nobody uses a computer this way. All users demand a robust operating system that allows application programs to run concurrently, facilitates interchanges of data through memory or pointers, manages the memory, and hides many machine-specific details of the input/output routines beneath the application layer. An analogous set of basic services is provided by a component framework, conceptually located in a layer above the operating system. The framework allows a scientific programmer to focus on scientific cores of components, with less attention to specifics of operating systems, data structures, error handling, or even the physical location of the computation.

Technically, DANSE is a “component-based runtime environment.” This means that the components are pre-compiled, and interconnected by the user at runtime. The user directs the interconnections of these components, using either a menu, a graphical programming interface, or a command-line interface, depending on need or preference. The user interface is a layer independent of the components, and can be replaced or modified without affecting the core functionalities of DANSE. For either distributed or local computing, the user could select a favorite interface for all neutron instruments, shortening the learning curve for new users, and encouraging expert users to experiment with new types of data analysis and computational science.

DANSE will be organized with the data flow paradigm as the basic abstraction, with a layer for execution control as shown in Fig. 1. Typical scientific analysis codes, written in languages such as Python, Fortran and C, will be the cores of software components. A component mediates in several ways between the core and its environment. The components inherit methods from the framework, including methods for passing data and handling errors. The component is responsible for the initialization of its core, which may require user-supplied information (depicted in Fig. 1 as information above the component boxes). After a component is instantiated, it provides information
Figure 1. (a) Schematic of the conversion of raw data into an energy spectrum. The component **NexusReader** is responsible for reading a file, and converting it into a data object. After further analysis and user-supplied information, the component **Energy** produces a histogram of intensity as a function of energy. Components such as **Bckgnd** and **Energy** are themselves composed of lower-level components. (b) Schematic of a software component that supports a scientific code at its core.

To the framework that could be passed to the user interface. Components will negotiate their data exchanges with the help of XML-based data exchange protocols when they are first connected.

The dark lines between components in Fig. 1a depict data streams between the input and output ports of different components. The conceptual decoupling of components from each other through data streams facilitates their physical decoupling. With the serialization of data streams,\(^3\) it becomes possible to distribute the computation among multiple computers. Alternatively, the user may prefer to organize a computation with multiple threads, or as separate processes on the same computer. The control of component execution is encapsulated in the executive layer of the framework. The executive layer manages the life cycles of components, and handles error conditions. With the assistance of the executive layer, components will have access to a centralized mechanism for the logging of status, errors, and for preserving a record of the computation.

The DANSE framework evolved from work at the California Institute of Technology in the Center for Simulating Dynamic Response of Materials, a DOE ASCI Center of Excellence.\(^4\) The software framework used for the ASCI project, **pyre** [3], was originally developed to support the upper end of high-performance computing, and could be adapted to high-performance networks such as the NSF TeraGrid [4]. The **pyre** framework offers maturity and stability – **pyre** has run over a million CPU hours, and is today in its seventh version. It has supported scientific computations on the largest computer systems, but has run on laptops. The **pyre** framework has run on Linux and most flavors of UNIX, and some versions of Windows, facilitating the work of developers. It is entirely open source code with a nonrestrictive BSD license.

For its use in DANSE, **pyre** requires new capabilities that are not yet available on any software framework for scientific computing. In particular, there is today no working software framework that can manage the execution order of multiple processes running simultaneously on different computers. The executive layer of the DANSE framework must control distributed executions when concurrency

---

\(^3\) For the domain of computational neutron scattering research, the main types of data to be exchanged between components are histograms and tables. Other lightweight information that describes the data will be passed as metadata described by XML.

\(^4\) This large program, now in its eighth year, has constructed a virtual shock physics facility [5]. Adaptation of the **pyre** framework for DANSE started in 2001 in a collaboration between Brent Fultz and Michael Aivazis to develop software for the ARCS spectrometer, one of the first instruments scheduled for operation at the SNS.
cannot be ensured. This capability for distributed computing is designed, and many of its components have been tested, but more work remains. Section 4 describes framework services for execution control and monitoring, recording the history of the computation, auto-installing the framework on remote computers, supporting a rich set of proprietary and open source visualization tools, and obtaining documentation. A sophisticated capability, scheduled for development later in the project, is concurrent computing with data as it streams between components in a distributed computation. The development of this capability and others in DANSE are of interest for grid-based computing.

§ Enabling Neutron Science with DANSE

The DANSE system will enhance the science of neutron scattering research in several ways, such as:

- The proposed DANSE system enables the design of more efficient experiments by use of prior simulations, and the opportunity to alter an ongoing experiment with information from modeling or simulations that can be performed as experimental trends begin to emerge.

- New types of experiments are made possible with the high-performance instruments at the SNS, but many new experiments require the development of new software. For example, software is not yet available for obtaining \( S(\vec{Q}, E) \) from most three-dimensional single crystals measured on inelastic chopper spectrometers. Beam time optimization for stress tensor measurements with an engineering diffractometer will benefit from near real-time computations that are not performed today. Today’s software for the analysis of SANS data from anisotropic samples restricts the information that can be extracted from experimental data, and better software is needed.

- Detailed simulations of the structure and dynamics of materials and molecules have become more generally accessible with the \textit{ab initio} codes developed over the last decade [6, 7, 8, 9, 10]. Sections 5-10 describe opportunities for new science by bringing these calculations and neutron scattering data analysis into the same software environment.

Additionally, the DANSE system will free many neutron scattering scientists from today’s drudgery of compiler options, linkers, and system-specific data conversion protocols, allowing scientists to focus on creating new types of data analysis procedures by interconnecting powerful components for neutron science computations. The developers of software for neutron scattering research are usually scientists who undertake this work by necessity. Many gravitate towards commercial data analysis environments such as Matlab [11], IDL [12] or Igor [13] to escape the lower levels of computing. The challenge is to offer them a powerful environment that is easy to use. An important design decision for the DANSE system was to build software components and their interconnections with a high-level programming language. In DANSE, program execution is directed by a Python interpreter [14, 15], using XML [16, 17, 18] for describing the data and control functions.

A software system that shields end users from many programming details will encourage more scientists to participate in what is today state-of-the-art scientific computing. Python gives scientists the freedom to experiment, giving them a way to rapidly interconnect software components into analysis networks, a consistent way to write small software components for control of the interconnections, and a way to control large components for materials simulations. It lets scientists build higher-level structures that connect components conditionally, contingent on the state of the computation. Different types of user interfaces will be available to access the framework, so many capabilities will be accessible by scientists with limited expertise in software development. A more diverse group of scientists will be able to customize their data analyses.

For advanced scientific programmers, a convenient interoperability with a diverse set of well-characterized components will encourage them to contribute new components to the DANSE system. Python is readily extensible to incorporate programs written in Fortran, C++, C, and offers interoperability to programs that run under the environments of Matlab and IDL. All software components in the DANSE system are Python objects, and on a single computer these objects interoperate through the Python interpreter [19, 20]. Computationally-intensive tasks will be coded in Fortran, C, and C++, and compiled into dynamic libraries. When components have well-planned granularity, the Python interpreter incurs negligible overhead when linking to these libraries at runtime, giving to the end user the flexibility of Python with the speed of optimized, compiled languages.
The DANSE project will be organized with a central services subproject at Caltech focused on software engineering, and five subprojects focused on software for the major science subfields of neutron scattering research, led by scientists at academic institutions in the U.S. The central services subproject will develop software and hardware that are needed by the five science-based subprojects, including the component framework, development tools, data structures, user interfaces, visualization components, and some simulation components. The subprojects will develop traditional methods for data reduction, using data from working instruments at the neutron sources at Argonne, Los Alamos, Oak Ridge, and Gaithersburg. All science subprojects have a focus on state-of-the-art methods for materials simulation and scattering calculations.

The productivity of the SNS for cutting-edge science will be enhanced considerably by a coordinated software development effort. A relatively small investment in new software can leverage the much larger investment in hardware at the SNS. There is urgency to start the DANSE construction project now, so that data analysis for the SNS can be managed after operations begin in 2006, and high-power operations begin in 2009. It is in the national interest for the SNS to deliver quickly on its promise of scientific advances in our understanding of the structure and dynamics of materials and condensed matter.

§ OUTREACH

After the white paper describing DANSE was circulated in early 2003 [21], DANSE received strong support from the neutron community in the U.S. and beyond. A software project that engages a diverse spectrum of scientists will help strengthen scientific interactions in the international neutron community, building on the Neutron Scattering Software Initiative (NeSSI), for example.

As a project to construct scientific software, DANSE offers educational opportunities at many levels. Fundamentally, many of the scientific concepts of neutron scattering are organized in hierarchies that can be reproduced in the inheritance relationships of object-oriented designs. Software designs that follow scientific principles can be robust in practice [22], and can offer a clarity and documentation that makes it natural for graduate students to be involved in the DANSE project. The K-12 outreach effort of the DANSE project will include a teacher training effort led by educational professionals who will work with the DANSE subproject teams.

3. Project Infrastructure

§ Structure of This Proposal

Most of this proposal is organized as a Work Breakdown Structure (WBS) description of the tasks for a DANSE construction project. This proposal text is accompanied by a Microsoft Excel spreadsheet that details the resources, effort, cost and schedule for each of the level 4 tasks. It was submitted into the NSF Fastlane system as a pdf document, but it should be available in Excel by request to the NSF. Heading numbers of Sections 3-11 of this proposal follow exactly the task numbering in the project plan. The 3-digit headings in this proposal text correspond to level 4 tasks in the project plan. A DANSE construction project would be similar in scale to an instrument construction project, but software projects bring their own challenges [23, 24].

3.1. Computing Resources

3.1.1. Hardware

High performance computing hardware is required for the development of software for high performance computing. A computer purchase is proposed for the first year of the DANSE project. The hardware includes 20 nodes of dual 2.0 GHz Opteron processors in 1U rack units, each with 4 GB of RAM and a 250 GB disk. This system is similar to, and would interoperate with, the 12 node

5 The exceptions are the first two WBS categories: 1. Milestones and External Drivers, and 2. Project Integration. More detail on these are provided in Section 12. The DANSE WBS has 11 categories, extending through Education and Outreach, whereas this proposal has extra sections. Unlabeled sections such as the present subsection are not part of the WBS hierarchy, but offer context, clarifications, and justifications of the proposed work.
Opteron system we have built for the ARCS spectrometer. The theoretical peak performance of the proposed 40 processor configuration exceeds 150 GFlop/s. For economy we have selected a GigE switch for internode communication, although this is not ideal for molecular dynamics calculations, for example. This hardware configuration was developed using a complete set of vendor quotations and an estimate for installation, but a future reassessment of the technology may provide a somewhat different hardware configuration at the same price point. A second large hardware purchase will begin early in the fourth year of the project. This system will emphasize high performance internode communication and high bandwidth external communications, capabilities that approximately double the price per node today.

3.1.2. System Administration
Half-time support is requested for a system administrator to set up and maintain the computing cluster and help with the computing infrastructure of the DANSE project.

3.2. Configuration and Release Management
We anticipate that the DANSE project will include dozens of developers, even in the early stages of the project. The developers will be distributed geographically, and will have different levels of experience with large-scale software development. Further complexity is added by requiring that the software operate reliably on a wide variety of platforms over many years. It is therefore crucial to establish proper configuration management procedures from the outset to ensure quality control as the software evolves.

One of the advantages that the DANSE software will enjoy is that it is built from scratch using a very powerful interpretive environment. This obviates the need for implementing policies and procedures with external tools that are typically designed to interact with monolithic applications. The same framework that manages the data analysis applications may be used for many aspects of configuration management. This approach becomes even more compelling when one considers that the software will be extensible by the end users, who need the same tools to create data analysis modules, for example. This subsection outlines essential practices that serve both the developers and the end users.

3.2.1. Automatic Configuration of Build Platform
We expect the DANSE software to be available on many platforms over many years, and users will write extensions to the software system using a variety of programming languages with an evolving set of compilers and run-time environments. We propose to manage this complexity by taking advantage of the framework to produce a tool suite that overcomes the disadvantages of the available open source solutions, and is freely available to our end users.

The system should be intelligent enough to detect the platform, operating system and compilers available to the developer, and allow her to specify the full tool set to be used for building the software. There is also a need for flexibility, allowing her to take advantage of installed third party tools and libraries that are supported by the DANSE software. The plan is to produce a system that utilizes a broad database of information about the various supported configurations, and is tunable at installation time to take advantage of the end user’s actual environment. The database will be constructed by the DANSE developers and will be distributed as part of the software. The configuration tool will be able to compare the set of supported facilities with what is accessible at installation time, and guide the user through the discovery and selection process.

3.2.2. Build System
Integrated support for building the software is one of the key factors that will determine the stability and longevity of DANSE. We propose to construct a suite of tools to perform all the steps necessary to transform source code into actual running applications. The goal here is to overcome the limitations of the existing tools by taking advantage of the power of the framework. Specifically,

- We will construct an XML-based specification language to describe the various aspects of supported platforms, operating systems, compilers and other tools that are required for a complete and consistent build of the DANSE software. These descriptions will form a knowledge database for the mechanics of building the software. The same specification language will describe how
third-party tools and libraries are to be used during the build process. Finally, each project
directory will contain a specification of the set of local files that are required for the build, and
their precise role in the build process.

- We will construct a tool to access the XML project description files and construct the full graph
  of dependencies of the products on the project sources. Using object-oriented techniques, the
tool will navigate the dependency graph, executing instructions for each step as prescribed by the
XML database.
- We will produce an object model and associated support so that the production scripts will let
users modify or extend the steps of the build, using Python and framework facilities.
- We will construct a full logging facility to record the various steps in the build process in an
XML-based hierarchical log file.
- We will design and implement a powerful, integrated debugging facility to help users diagnose
problems with the build.
- We will take advantage of the framework to produce a facility for performing unattended builds
of the system on a regular basis (perhaps nightly), so that developers can have regular feedback
on how the code they are producing behaves on systems to which they have no direct access.
- We will construct a reporting facility that can consolidate the log files from multiple builds into
coherent reports of the state of the software.

3.2.3. Test Harness
Automated, regular testing of software is absolutely essential. Once again, our strategy will be to
build in support for automated testing by taking advantage of the fact that the software is so easily
accessible through Python. This approach should reduce to manageable levels the complexity of any
realistic testing of distributed components. The plan consists of the following steps:

- We will design an XML-based language for describing the expected output of tests. This lan-
guage will be flexible enough to describe both output that is suitable for byte-by-byte string
comparisons, or other forms more suitable for the verification of numerical routines.
- We will construct specialized harnesses for the DANSE components that utilize the integrated
debugging facilities. The goal is to be able to perform extensive testing on isolated components,
e.g. unit testing, and a full analysis of distributed networks.
- We will build an extensible set of tools to detect deviations of computed results from expected
results. Flexibility and extensibility are essential because many analysis codes will require
specialized error detection algorithms.
- The testing system will be able to keep the results of test suites in a database for future retrieval.
This feature will enable us to track the state of the system over time.
- We plan to implement analysis tools that utilize various proven metrics to assess the state of
the software. Of course, the construction of test plans for distributed systems is not a mature
subject. This issue is addressed separately in Sec. 3.2.4.
- We will construct a tool to help visualize the results of tests, and generate web-based test
reports.
- We will build a reporting tool that will be able to construct a comprehensive view of the quality
of the software on a regular basis. Further we plan to take advantage of the test result database
to monitor system behavior as the software evolves. Such reports are valuable for deciding when
the number of defects is low enough so that a new release is possible.

Using the same framework for the construction of the configuration management tools and for the
construction of the software system itself will allow us to construct a thorough and flexible test plan.

3.2.4. Quality Assurance
The Software Quality Research Laboratory (SQRL) associated with the Department of Computer
Science at the University of Tennessee, Knoxville will receive a subcontract to develop testing
procedures for the distributed computations of the DANSE system. The SQRL will help the science subproject teams develop codes according to consistent software quality standards, and within a common test plan. The SQRL role in testing will facilitate independent verification and certification of the DANSE software quality, and its usability by the SNS.

3.3. Personnel and Other Resources
The effort on project infrastructure will be directed by Michael A.G. Aivazis, who will be assisted by staff members of the Center for Advanced Computing Research at Caltech. A subcontract will be issued to the Univ. of Tennessee, Knoxville, for support of personnel at the Software Quality Research Laboratory.

4. Central Services: The Software Engineering Effort
This section presents in more detail the tasks needed to realize the software vision articulated in Section 2. What we propose can be achieved within the scope of this project only because it represents a natural extension of pyre, a pre-existing software architecture that has been in development for many years and has been shown to satisfy the requirements of high performance scientific computing while facilitating the construction of usable applications.

The most challenging extension involves augmenting pyre so that it can handle components that are distributed over a network. Retrofitting distributed computing in existing software architectures is notoriously hard, but pyre was originally designed to help compose and control massively parallel scientific applications remotely, so its design already anticipates many of the required changes.

The other large source of risk for this project is the difficulty in providing access to a diverse user community with many backgrounds, interests, levels of expertise, and investments in existing software. In Sec. 4.2 we describe how we plan to integrate many of the existing visualization tools into the DANSE toolkit. In Sec. 4.3, we describe the strategy for mitigating the risk associated with graphical user interfaces by lowering the cost of constructing them and grafting them on top of existing DANSE applications.

4.1. Distributed Component Framework
The framework encourages the construction of data analysis applications that employ the streaming computational paradigm, in which components receive data in input streams, modify them, and produce output data streams. Even applications that at first sight are not good candidates for this architectural model, such as heavily interactive ones, can be designed well with this model.

One of the most compelling reasons for choosing the streaming paradigm is the possibility of enabling distributed computing in a manner that is entirely transparent to scientific programmers, i.e. the entire burden of managing the complexity of a distributed application can be borne by the underlying framework. This transparency also benefits the end user, who can select at run time the resources that are employed in a given computation. It is our goal to bring the power of distributed computing to users who lack the resources to effectively participate in the large national grids. Neutron scattering scientists would contemplate distributed computing only if it is possible entirely in user space without overly-complex software. We are proposing a system that gives users full support for creating ad hoc grids using computer systems for which they are authorized users. The following sections describe the tasks needed to enhance the pyre framework with the ability to manage distributed applications.

4.1.1. Distributed Component Control
An application composed of cooperating components can be transformed into a distributed application by providing alternative implementations of its framework services. This transformation can occur dynamically, after remote computational resources are discovered. The following tasks will enhance the component management layer of the pyre framework:

---

6 As mentioned frequently in this proposal, we anticipate that many of the underlying required services will be provided by standard Grid software eventually. We will ease this migration by constructing an abstraction layer to encapsulate all the required mechanisms.
− Runtime bootstrapping. To utilize remote computational resources, the framework will be able to deploy a remote process that launches, manages and monitors the computations that are performed on the remote machine. We refer to this process as the service factory. The service factory acts as a proxy of the framework, and acts as a registry that enables the application to discover what computations can be launched on the remote resource.

− Service deployment. The framework will be extended so components can be launched in a specialized harness that enables their remote management, monitoring and invocation. We refer to such components as services. One of the responsibilities of the service harness is to allow components to respond asynchronously to outside events, as required by the fact that the actual application is running remotely.

− RPC protocol implementation. Remote Procedure Calling (RPC) is an area of active research, especially by the Grid community in the context of web services. Our early implementations of remote control of components will continue to be based on XMLRPC, a very simple XML-based protocol with excellent support provided by Python. We are confident that as the community settles on a standard mechanism, Python bindings will be forthcoming and we need only replace the communication layer.

− Component life cycle management. We will extend the framework so that components that have been launched as services can be initialized, configured with parameters provided by the remote application, and finalized when they are no longer needed. Further enhancements will allow applications to monitor services and redeploy them if they develop faults.

− Static and dynamic data connections. The data sources and sinks of services are very likely to be other remote services. We will provide mechanisms for notifying services about their data sources and sinks, establish the required connections between them, and add handlers for data streams to their event loops.

− Data and meta-data transport. We will construct efficient data encoders and decoders to allow the serialization and transportation of data and meta-data. These codecs will use information about the computational proximity and compatibility of the target services to guarantee accurate and efficient delivery of the data stream. For example, two components that live in the same process space can exchange data using pointers, obviating the need for serialization, while components on different hardware platforms must stream data in a way that accounts for different representations of numeric data. We plan to use XML-based documents to transport meta-data because they are well suited for low volume, semantically-rich content, whereas bulky data streams, such as detector counts, may be transported in compressed binary streams.

− Error handling. Finally, we will extend the error handling capabilities of the framework to accommodate the detection, classification, propagation and delivery of error conditions from remote services to the application. Our goal is to deliver exceptions generated by a remote component as exceptions in the application, making error handling as transparent as possible. Furthermore, we will provide exception handler support so that the offending services can be shut down gracefully. Certain applications may even be able to restart services without disturbing the logical coherence of the calculation, and we plan to provide mechanisms to support this.

4.1.2. Scripting Access
The capabilities of the distributed control layer of the framework must be accessible directly from Python, so complete distributed applications can be constructed as Python scripts. Some users will find this mode of operation preferable to graphical user interfaces, which necessarily provides a subset of these scripting commands. This strategy enforces a separation between the business logic of an application and its user interface. Furthermore, it makes it possible to satisfy the design requirement of enabling unattended, automatic testing of the application without specialized harnesses for the user interface. The following tasks will enable scripting access to the distributed control layer:

− Job specification. We will provide mechanisms for the end user to describe the set of machines to be used for the computational grid for a given application. The user will describe to the application the expected state of the initial environment in which the service factory is to be
deployed, and specify where the database of available services is located. Further mechanisms will be provided to automatically verify that all the services required by an application are launched somewhere, and all deployed services receive the set of services they need.

- **Observers.** We will provide support for third parties to observe running computations. We will provide mechanism to allow the service factory to authenticate and determine the access privileges of observers and guarantee the coherence of the view. Observers will be able to modify dynamically the amount of monitoring information they receive.

- **Active monitoring.** Certain privileged users will be able to deploy and activate application probes that extract monitoring data from a running application, as well as actuators that enable normally dormant execution paths, such as those that perform extensive tests of the internal consistency of the calculation. These features will be invaluable during the debugging of distributed applications, and may be of significant value to end users.

- **Steering.** We will provide the ability to dynamically modify the execution context of a service. This will enable a certain amount of interactivity that is necessary for some data analyses.

- **Graceful finalization.** We will provide a set of error handlers that assist in the detection and diagnosis of faults, and provide a graceful way of shutting down computations.

- **Error recovery.** In certain cases, it may possible to recover from faults. We will explore the usage patterns of our codes, and attempt to classify the recoverable faults. This may pave the way for a reliability facility.

- **Synchronous execution.** We anticipate that the synchronous execution model will be very well suited for command line applications. In this model, computation proceeds by executing a script, one line at a time, with the script author in complete control over the sequence of computational steps. The script author will be able to control transparently the creation of local proxies for remote services, the necessary data connections between services, and have remote faults delivered as exceptions that can caught locally. This will be accomplished by organizing the mechanisms provide by the framework in a coherent object model.

- **Asynchronous execution.** Asynchronous execution takes place by composing applications as a set of handlers for a well-defined set of events, and then letting the state of the application be determined by the sequence in which the various events take place. While somewhat more advanced, this model is required for applications with graphical user interfaces. It is also better suited for writing distributed applications with good performance and scalability. We will extend the synchronous capabilities of the framework with the ability to explicitly control the flow of services, based on the application context, and provide computation actuators that initiate the control cascade as a response to application events.

These tasks will provide the fundamental mechanisms necessary to compose robust data analysis applications that are easy to verify and validate.

4.1.3. **Standard Services**

Many of the services required by conventional applications are provided by the operating system and the user’s shell. We will construct distributed analogs of many of these services. The following tasks will provide a basic set of required services.

- **Authentication.** A user must present valid credentials to a service factory with each request for the creation of new service handler. The service factory must present these credentials to an authentication service, which verifies that the credentials are valid and have not expired. In a similar manner, observers of a given calculation must also be authenticated. We will construct both password and secure-shell-based authentication services so users can install and use our applications with minimal administrative support.

- **Access control and privileges.** We will construct access control services that will determine the full set of privileges that have been granted to a requester by the application’s owner.

- **Accounting.** Similarly, we will build accounting services that will be able to keep track of the resources used by each user.
− **Global instance identification.** We will construct a service that will assign a globally unique identifier to each instance of a service. Applications that require multiple instances of services will therefore have a way of identifying uniquely. The same service will be able to resolve the identifies back to services by providing the host name and port number of the associated service.

− **Journaling.** We will enhance the existing monitoring and logging facilities provided by the pyre framework by constructing a journaling service. This service will be used by all enabled codes to act as an accumulator of all diagnostic messages generated during the course of the calculation.

− **Job persistence.** Finally, we will construct a service capable of contacting the full set of components in a distributed application to collect enough information so that the calculation can be restarted from an intermediate state.

As mentioned elsewhere, we are mindful of the developments in the Grid community and we will ensure that our implementations can accept the standard services produced by the Grid community as drop-in replacements.

4.1.4. **Integration of GGF Services**

Many of the services described above are forthcoming from the Grid community. It is possible that many of them will have standard interfaces with reference implementations by the time funding for this proposal arrives. Furthermore, as the Grid matures and an increasing number of scientists realize the enormous potential of distributed computing, it will become critical for the DANSE software to support the national grids. We will continue to monitor the development of the grid software. We are committed to ensuring that DANSE runs on the Grid as it will be understood by the time of the first public release of DANSE.

4.1.5. **Grid Services Light (GSL)**

The TeraGrid, a research-oriented hardware and software infrastructure, is a natural platform on which to test and deploy DANSE. Both Caltech and Oak Ridge National Lab are nodes on the NSF-funded TeraGrid, facilitating access to its resources. However, the existing middleware in the TeraGrid standard software stack has interfaces that are far too complex for use by high level applications. This has motivated us to construct an encapsulating layer whose interface is a better fit for the needs of DANSE. We call this layer Grid Services Lite (GSL). The design and implementation of this layer will facilitate the eventual transition to Grid-based distributed computing in a manner entirely transparent to DANSE components. DANSE, however, aims to deliver functioning software to real users in a rather short timeframe, so initial implementations of all these facilities will take advantage of existing technologies, such as ssh, the secure shell. Today GSL uses the internet, but it does not require the existence of the Grid itself.

Distributed computing with DANSE makes use of the pyre framework, which must be present on the different computers running a DANSE analysis procedure. Two steps are required for this service. First, the remote computer must be configured properly for the computation, and second, computations on the remote computer must be staged, launched, managed, and terminated. The GSL extensions of the pyre framework perform these steps with conventional technologies. The GSL client on the user’s laptop coordinates these steps in a way that appears to a remote computer as a very fast and highly organized user.

The first parts of GSL are built and tested with UNIX-flavored systems. They involve:

− Detection of the remote filesystem, with automatic generation of a report to the client software.

− Modification of the remote shell environment files if necessary.

− Checkout of missing framework components and packages from the DANSE cvs repository, and building them on the remote system.

These steps are driven by a single controlling script that can query the remote system and formulate the optimal path for installation of the pyre framework with its build procedure. These tools will also be used for loading and updating the pyre framework on a user computer.
The second steps of GSL involve launching a process from Python via ssh/scp/mpi\texttt{run}. These steps are not yet complete, but will involve at least three levels of execution, which can be completely independent of one another:\footnote{\textit{\footnotesize For example, the part of the program that understands how to form an mpi\texttt{run} command line should be independent of the component that understands how to use os.popen. This is delicate if one wants to interact with a child process. When output comes from a child process, a manager will examine the output and decide if a component needs to respond.}}

- The immediate Python execution strategy, examples include \texttt{popen}, \texttt{os.system}, \texttt{fork/exec};
- an intermediate execution strategy: \texttt{ssh}, \texttt{mpirun}, and others;
- the launching of the executable.

4.1.6. \textit{Implementation of GSL Using the TeraGrid Standard Software Stack}

In the course of the DANSE project we propose to develop a path forward to migrate GSL and the higher-level DANSE science applications to the TeraGrid. This involves two steps. First we must identify the Grid services required by DANSE. To do so, we propose to develop realistic use cases for data analysis software, and use these cases to derive the requirements for distributed data analysis. We will:

- Identify services in the TeraGrid standard software stack that are missing or incomplete.
- Build a suite of unit tests to exercise the various distributed aspects of the system.

The second step is to extend the \texttt{pyre} framework to take advantage of the existing Grid services. This task is expected to consist mostly of technology transfer because Keith Jackson’s group at the Lawrence Berkeley National Laboratory is building bindings for the relevant services that are easily utilized. Specifically, we will:

- Identify implementation strategies for the missing services.
- Construct a realistic full-application test case. Our candidate test case is the on-demand computation of space-time correlation functions from a molecular dynamics code. This will allow us to test our ability to compare predictions from a molecular dynamics model and experimental data in near real time.

4.1.7. \textit{Reliability}

As mentioned above, the framework will provide excellent support for fault detection even in the case of distributed deployment. Later in the project, we propose to build upon these facilities and explore the issue of reliability more fully. This is a topic of current research, so it is difficult to construct a list of deliverables today. We intend for DANSE to be a candidate platform for large-scale experimentation on reliability issues. Furthermore, we are committed to perform all the necessary modifications to the component management and data transport layers to enable transaction-based computing, and process migration among computational resources.

4.1.8. \textit{Deployment Optimization}

Similarly, we envision DANSE being a natural platform for testing deployment optimization strategies in distributed computing. DANSE computations will already be componentized. We intend to decorate the computation with meta-data that outlines the computational needs of each component, both in terms of processor cycles and network bandwidth, so that a deployment optimizer can make intelligent recommendations about the correct deployment location for each component.

4.2. \textit{Visualization}

4.2.1. \textit{Standard Graphical/Plotting Environments}

The most passionate discussions in any workshop, survey or poll concern interactive data graphics and analysis software packages. We have made no attempt to select “the” graphics package for DANSE. For two reasons, porting several viable graphics packages to the DANSE framework for the $\alpha$-release is less effort and risk than selecting “the” graphics package \textit{a-priori}: 1) There are vocal supporters of most of the viable graphics packages. There is no compelling reason to deny
them the tools they like. 2) The usefulness of the different packages is completely different on a component framework than as stand-alone software packages. For example, as standalone packages, Grace has far less functionality than Igor. On the other hand, a Grace component interoperates with all components of the DANSE system, so it is not obvious which package best empowers a user of DANSE. The ultimate utility of the different data graphics packages can be assessed only after a working release of the DANSE system is available. Evaluation of the graphics components requires user testing and Darwinian evolution, which will ultimately determine the best set of interactive data graphics packages for the DANSE system.

Open Source Interactive Graphics Packages
The following open source packages have been, or will be, ported to the DANSE framework:
- Grace, a 2D plotting system, has already been ported to the pyre framework. Grace provides both a convenient point-and-click GUI and a scripting interface, and can produce true publication quality graphics.
- The old PGPLOT graphics subroutine library has Python bindings, and is used by other software packages for making simple scientific graphs.
- Gnuplot is a portable, interactive, command-line driven graphical program that was originally developed for visualization of mathematical functions and data. Gnuplot also serves as a non-interactive plotting engine for miscellaneous portable third-party applications, like Octave.
- ParaView is funded by the US Department of Energy ASCI Views program for display of 3D data. It is based on the 3D visualization library VTK.
- Matplotlib is a Python 2D plotting library that produces publication quality figures, and works with many interactive GUI environments. Matplotlib can be used in Python scripts, interactively from the Python shell. Its internal language is mostly compatible with Matlab.

Commercial Integrated Graphics and Programming Packages
The following proprietary packages have been, or will be, ported to the DANSE framework:
- Matlab has already been ported to the framework. It is a commercial quantitative programming environment distributed by MathWorks.
- IDL now has Python bindings, and IDL has been partially ported to the framework. IDL is designed for visualizing large and complex datasets.
- Mathematica is a numeric and symbolic computational engine with a programming language and a graphics system.
- Unfortunately, Igor can be launched only as a standalone package because there is no external access to the Igor command parser.

Some applications require tighter integration with the graphics package than simple bindings allow. An example would be selecting a point, dataset, range or region of the graph with the mouse and using this to control further processing of the data. Developers of integrated applications will want to embed the graphs into forms, and add context-sensitive menu choices for data operations. The task of selecting an existing plotting engine, integrating it with the DANSE standard widget set (Section 4.3.3), and adapting it to the needs of the subprojects is the responsibility of this Central Services activity.

4.2.2. Advanced Graphical Environments
Many of the advanced analyses that we hope to enable through DANSE involve the specification and interaction with complicated 3D geometries. For example, specifications of 3D shapes are necessary for the calculation of shape factors for scattering intensities, $S(Q)$, for multiple scattering calculations for real samples, and for the specification of sample shapes for engineering diffraction. Other analyses involve geometrical queries on complicated surfaces specified either explicitly or implicitly, such as obtaining the $S(Q)$ appropriate for SANS by Fourier transforming the scattering length distribution.

The pyre framework already includes excellent support for solid modeling by providing an object model for the specification and persistence of complicated geometries, as well as a complete set of bindings for ACIS, the commercial solid modeling engine behind CAD tools such as AutoCAD. To
augment these capabilities with a graphical user interface, we will provide access to environments such as Blender3d and ParaView.

4.3. User Interfaces

The intended audience of the DANSE software is a community of users with diverse needs and backgrounds. No single user interface approach can satisfy the needs of a majority of users, and this view is confirmed by the user polls we have conducted in recent years, which further point to the user interface as the largest source of risk for the entire project. Fortunately, this problem is rather well understood by the software development community and it provides yet another opportunity for DANSE to bring state-of-the-art software practices to the neutron scattering community.

Our approach consists of constructing an object model for constructing and controlling DANSE applications as a set of cooperating components. This object model will enable access at the Python level to the various framework services, such as those described in Sec. 4.1.3. This will be a user interface layer for users who are comfortable with programming in Python, allowing them to write batch scripts or develop new functionality. This layer is described in Sec. 4.3.1. In Secs. 4.3.2 and 4.3.3 we describe the support for constructing more traditional graphical applications, suitable for specialized analyses by users who do not want to deal with programming. Finally, in Secs. 4.3.4 and 4.3.5 we propose two visual programming user interfaces that allow the end user to compose applications dynamically by dragging and dropping components on a canvas, as shown in Fig. 2.

4.3.1. Scripting

Our plan is to allow users to write entire applications in Python scripts, so the full capabilities of the DANSE system must be accessible to Python. The pyre framework already provides extensive support for this, and will be augmented by components that perform specialized computational tasks, as well as interface with the administrative services necessary to stage DANSE applications, such as those described in Sec. 4.1.3. In general, the exposed component interface will be too detailed to be usable by end users, so we propose to construct a framework layer that facilitates the composition of applications in terms of a simpler set of control objects that stage the application.

These simplified framework services will address application issues such as whether to run in a distributed or standalone mode, which distributed resources to allocate, how to perform host specific runtime environment initializations, and how to deploy the required subset of standard services. Further, this layer will facilitate the selection of the relevant computational components, initialize them in a consistent way, and allow the user to specify the data flow path. Application control will then be handed back to the user’s script, which will be responsible for guiding the various components through the necessary data transformations. The scripting layer will also be responsible for monitoring the status of the calculation, delivering messages from the various components, trapping errors so that they can be processed by the user’s script, and saving the state of the application so that calculations can be restarted at a later time.

4.3.2. Graphical User Interface Infrastructure

In addition to providing a powerful means for composing neutron scattering applications, the scripting layer described above forms the basis for building powerful user interfaces. The decoupling of the application logic from the interface allows the latter to be specialized to the needs of science subfields with relatively little effort. In fact, we propose to make the construction of traditional graphical applications as simple as it can be by constructing an XML-based language for describing the interface definition. This will allow non-programmers to build specialized graphical user interfaces that harness the full power of the DANSE software without the need for mastering the complexities of the various GUI toolkits. There are already efforts in the software community to make such things possible. There are a variety of interface definition languages, such as XUL from the Mozilla group or, more recently, UIML from OASIS. GUI generators abound, but they tend to be intimately associated with a particular toolkit. A prototype facility in the context of the DANSE architecture has already been constructed and demonstrated. We propose to bring together the best solutions available and build a portable, toolkit-independent description language for our user interfaces, along with tools that can render these descriptions as functioning applications by relying on the interpretive nature
Figure 2. Screen shot of a demonstration version of ViPEr client software for distributed computing. Boxes on upper right represent libraries of data analysis components, local or remote, on five file systems. The workspace for arranging data analysis procedures is on the lower right, with components in place for reducing a data set from the Pharos instrument. As the cursor is placed over input or output ports, a description of data types is presented in a dialog box. Upper left shows the Matlab window launched by the analysis network, in which the data files were corrected and normalized. The output from Matlab was sent into the graphical display in the lower left window for comparison with a lattice dynamics simulation on the right of the window. (This display was generated by ISAW, an integrated spectral analysis workbench [25].) Components that ran remotely are bordered in purple, local components are bordered in yellow.

of Python to effect the associations between the layout described in the interface definition file and the behavior provided by the DANSE components.

4.3.3. Standard Widget Set
We will provide a set of user interface widgets for the DANSE framework. These include input widgets such as labels, buttons, text, tables, trees, lists, toggles and dials, and layout widgets such as grids, frames, notebooks, panes and dialogs. There will also be specialized widgets such as a value-uncertainty-units widget, and a file selection dialog which uses the metadata associated with the measurements to help the user select the appropriate datasets, and an interactive graph widget. Context-sensitive help will be available for all widgets. Interface layout will be device independent, so packers will determine the amount of space available to individual widgets, scroll bars will appear where needed, and fonts will be relative to a user-selected base size. End user scientists will be able to create powerful applications using simple forms and associate visual control elements to with inputs, outputs and behavior of component networks.

4.3.4. Vision Extensions
For the first years of the DANSE project, we will adapt a GUI developed by Michel Sanner’s group at the Scripps Oceanographic Institute, the “Visual Python Programming Environment” (ViPEr) [26], which is based on open source software. This user interface is shown in Fig. 2, and described in the caption. For purposes of a public demonstration, we have modified the execution engine of
ViPEr to support distributed computing. The new version of this package is called Vision. We will work with Sanner’s group to augment Vision so that it can be a viable interface for the DANSE software (see Sanner’s letter of commitment). Some of this is already underway since Sanner’s group has been funded at the level of M$ 5 for porting Vision to the TeraGrid.

4.3.5. **Visual Component Map Editor**

In the longer term, we will construct a visual programming environment that takes advantage of our component framework from the ground up. This environment will be built using OpenGL for all aspects of its interface and therefore will not suffer the usual portability problems that plague user interfaces that are built using a particular toolkit. We have already produced various visual prototypes of the look and feel, and the response from users has been promising. We expect other projects that rely on the pyre framework to contribute to this task.

4.4. **User Layer Data Services**

At the user layer, DANSE data services will give convenient access to scattering data and metadata. These services will help users locate data in a distributed world, couple that data to DANSE streams, and manage the new data generated from analyses and computations performed within DANSE. In addition, the DANSE executive layer will simplify the management of software licenses and intellectual property.

4.4.1. **Access to the Neutron Science Portal and Facility Databases**

DANSE components will receive data from many sources, with the data repository at the SNS expected to be a primary one. Remote login to a data portal is a priority for the SNS software group, and extending the SNS portal login to DANSE users will be one of the first capabilities implemented by DANSE. More generally, users’ data will be stored in heterogeneous databases at facilities around the world. DANSE accessor components that manage user authentication and session tokens will negotiate access with neutron facility database servers. High-level repository proxies will manage users’ views of what data is available to them, while low-level proxies will manage the exchange of data between files and streams. In both cases, interfaces that are independent of the underlying source will offer uniform access to the user.

4.4.2. **NeXus and XML Tools**

XML (eXtensible Markup Language) applications have become the ubiquitous solution for structuring networked communications. DANSE will provide XML parsers and renderers for neutron scattering data and metadata, as well as generic tools that can be specialized to applications that emerge in the future. Generic XML tools include components to render and parse information to and from XML documents. Parsers are built on Python’s standard library tools. Renderer components will use the standard visitor pattern [27], decoupling the data source from the rendered product.

The NeXus International Advisory Committee (NIAC) is engaged in specifying XML document type definitions (DTD) for a variety of neutron and X-ray metadata, including instrument, sample, and experiment configuration [28]. DANSE will provide a standard component to render the structure of NeXus files into the XML application defined by these DTD’s. NeXus XML documents become a serialization of data structure, independent of the data source. Components will render this information to display it to users in a coherent fashion. Other components will link user selections back into stream selectors, thus initializing the transfer of data to downstream components. The reverse process will also be supported: users will be able to stream data from components into NeXus files; these transformations will be guided by a complementary set of renderers and parsers.

4.4.3. **Data Provenance**

Because the executive layer of the DANSE system schedules the execution of analysis components, it can maintain a record of the sequence of processing steps used to obtain a particular result, including components’ parameters and versions. Combined with the archive of analysis components, this information – recently termed “data provenance” – forms a robust archive of the data analysis.

---

8 Although NeXus is currently designed as an archival format for experimental data, it is expected that the NIAC will shortly extend NeXus to include processed data and instrument simulation data.
enabling researchers to later recall precisely what they did. Data provenance also helps the executive layer restart interrupted computations, provide more convenient debugging tools, and recreate past computations. In addition to compiling this information during execution, DANSE will archive the data provenance with generated data. Again, the use of XML tags will simplify the mapping of data to and from databases, including databases maintained by the SNS.

4.4.4. **Intellectual Property and License Monitoring**
While DANSE will not provide licenses for proprietary software, it will help users manage their licenses. At the start of the lifecycle of a component that uses licensed software, the executive layer will query the license status of that software. If the license is available, the framework will manage the license and any associated session resources, then proceed. If the license is unavailable, the component will notify the user, relaying information from the underlying license manager. The executive layer will address similar issues of intellectual property. For example, the user will be informed of special requirements of software components, such as citation requirements for codes that are research products.

4.4.5. **Documentation Librarian**
Our recent survey on media for software documentation showed that although professionally-printed books are appreciated for their quality, they are considered outmoded. The 90% response in favor of high-quality on-line documentation is considered a mandate for developing the “Documentation Librarian,” an electronic system for distributed online documentation. At the user’s request, components will query the Documentation Librarian, which will retrieve information in remote repositories through the framework's distributed capabilities, convert the information as needed, and return it to the user’s interface to be rendered locally. Much of the DANSE documentation will be formatted using DocBook, an XML format in widespread use for technical books and papers. Documentation contributed by others may be in various formats: simple text, html, and \LaTeX. Acrobat pdf documents will serve as output formats for users of proprietary software.

4.5. **Personnel and Other Resources**
The effort on central services will be directed by Michael A.G. Aivazis, who will be assisted by staff members of the Center for Advanced Computing Research at Caltech.

5. **Common Tasks for Scientific Subprojects**

§ **Science Subfields**
The field of neutron scattering is typically arranged into scientific subfields for the purposes of beam time allocation at national user facilities, for the organization of sessions at conferences, and often for the development of new instruments and capabilities. For the DANSE project, scientific subprojects proposed in the five scientific subfields are: 1) diffraction, 2) engineering diffraction, 3) small-angle neutron scattering (SANS), 4) reflectometry, and 5) inelastic scattering. Specific tasks for each of the five subprojects are defined to level 4 in detail of the WBS in Sections 6–10. Each subproject has considered the generalized roadmap shown in Figure 3, which shows three paths from experimental data to science. “Reduction,” “Modeling,” and “Simulation” are presented as independent paths to scientific results, and separate efforts on reduction, modeling, and simulation allow for a more straightforward organizational structure of the DANSE project. Nevertheless, the software components from these three efforts will be capable of a high degree of interoperability through the DANSE framework. The subproject leaders have recognized several opportunities for better science by the integration of reduction and modeling, for example.

Over the course of the DANSE design effort, we have identified data structures, user interface components, algorithms, and families of scientific software packages that are common to multiple subprojects. Extensive discussion and negotiations between the investigators and the subproject leaders led to a systematic identification of duplicated efforts, especially on software for user interfaces and visualization. Freeing the subprojects from duplicating these tasks offers good economy, and promises better uniformity of the delivered product.
Figure 3. Three paths to science, starting with neutron scattering data at upper left. The traditional approach is Reduction (Path 1 across the top row). Some science has been done with Modeling (Path 2 vertical, center). The DANSE project will also implement the Simulation path (Path 3 vertical, left).

§ Open Source and Commercial Software
The DANSE project cannot distribute commercial software such as Matlab and IDL. We will nevertheless ensure that these commercial products interoperate with other DANSE components through the pyre framework so that licensed users can use them effectively. For selected commercial software packages we will write Python bindings, and integrate their returned Python objects as framework components. The bindings will be distributed without the commercial software itself. For open source software the situation would be similar, but more flexibility is possible. In some cases it has been possible to provide Python bindings for multiple entry points into an open source package, to individual files for example, giving the pyre framework a fine degree of control over the execution of these packages.

5.1. Data Reduction
The highest priority for the SNS instruments is the traditional Reduction path shown in Fig. 3. A basic data reduction service is absolutely essential, and all subprojects propose software capabilities to obtain $I(Q)$, $S(Q)$, $S(E)$, or $S(Q,E)$ in physical units such as barns, Å$^{-1}$ and meV. Raw data, such as time histograms of counts in detector pixels, must be converted into normalized intensities (cf. Fig. 1a). Instrument backgrounds and other distortions must be removed. Components are required for interactively visualizing the data after the different steps of processing.

5.1.1. Common Data Reduction Library
In developing the ARCS data reduction software, we found that many of the transformations of data arrays can be derived from a common set of base classes. Instrument-specific classes account for only a minor fraction of the actual code. Building a common data reduction library is a task of high priority. We will develop data reduction components for single crystals, powders, and liquids. These data reduction components are typical of those in use today, and some of today’s components will be integrated into the DANSE framework.

5.1.2. Advanced Data Reduction Library
Higher-level manipulations of backgrounds, arrays, and detector calibration files would facilitate the work of the DANSE subproject teams, and advanced users. This effort will be a modest one, however.

5.2. Modeling of Data
Sections 6–10 describe the new types of science that would be enabled by the proposed software effort. We believe these new capabilities are where the investment in DANSE would produce the greatest return. Modeling of the physical scattering, a vertical chain in Fig. 3, is for comparing reduced data to models of sample structure or dynamics. This path is especially appropriate for analytical models
with adjustable parameters. Although modeling efforts are included in each science subproject, we have found that each model tends to be specific to a subfield of science. Only a few such modeling components are appropriate for inclusion into this Common subproject. These are:

5.2.1. **Texture Analysis and Visualization**
The software package MAUD (Material Analysis Using Diffraction), is a fitting program for diffraction patterns that allows for calculations of pole figures and diffraction line broadening. The functionality of MAUD will be provided as DANSE components, with a plan for interoperation with mechanics models. The core methods of MAUD will be rewritten in C++. A close collaboration will be established with the texture group at LANSCE, headed by Dr. S. Vogel, and data from the HIPPO instrument will be used for testing the code.

5.2.2. **SLD Calculator**
Calculating the scattering length density is a prerequisite for all experimental planning. The proposed SLD calculator tool will begin as part of the reflectometry effort, and will be adapted to the other science subfields after other users evaluate it.

5.3. **Full Simulations of Data**
In its purest form, a full experiment simulation (Simulation in Fig. 3) would track individual neutrons from the moderator, into the instrument, through the sample, and into the detectors. Monte Carlo simulation packages such as McStas, VITESS and Ideas [29, 30], are used today to design neutron instruments for optimum combinations of resolution and flux. A goal of the DANSE project is to integrate the structure and dynamics of the sample into these instrument simulations, and calculate the scattering from the sample into the detectors (e.g., Sections 7.3.1 and 10.3). This approach to modeling the raw data, tuned with experimental calibrations, could have advantages over methods where a model is compared to the \( I(Q) \) and \( S(Q,E) \) obtained by data reduction (the “Reduction” path 1 in Fig. 3). Resolution functions of the instrument may have irregularities that are lost when the data are rebinned into \( Q \) and \( E \), for example. Full experiment simulations will also be valuable for research on disordered solids, where there are problems with analytical models of structure and dynamics.

**Instrument Simulations**

5.3.1. **Monte Carlo Simulations**
The Monte-Carlo neutron instrument simulation packages to be ported to the DANSE system are:
- **McStas** is already running under the pyre framework. It is now straightforward to build a virtual instrument from McStas instrument components using the pyre framework rather than the McStas run module. Almost all of the adaptation to the framework was performed automatically without any changes to the McStas C language code, so new releases of McStas components can be converted automatically into pyre components. It remains to adapt GUI components for McStas.
- The NISP package from Phil Seeger will be ported to DANSE. Although this is a legacy FORTRAN code, its entry points were recently exposed to make it operate with a web interface. A preliminary assessment showed that it would be straightforward to convert the full NISP package into a DANSE component.
- The VITESS package is modular and is written in Tcl, making it promising to port to DANSE using the TkInter interface.
- In Ideas, a neutron component is a pre-compiled module that is dynamically loaded at runtime. We expect it to be straightforward to port components of Ideas into the DANSE framework.

**Materials Simulations**
An important advance in materials theory over the past decade has been the development of software for calculating the electronic energies of solids, from which the structure and dynamics of atoms and

---

9 Rietveld refinement is a well-known example where results are often reported in terms of the parameters of the refinement, sometimes parameters specific to a particular software package used for the analysis [31].
(classical) spins in materials can be predicted. In the past few years, these codes have almost become commodity items, requiring a much lower level of expertise than a decade ago. They still require some effort to initialize properly, but as more cases are run with them, the initialization is becoming easier. The DANSE project will distribute Python bindings for these packages so users can run them under the framework.

5.3.2. **Solid-State Calculations**

*Ab initio* solid-state calculations of electronic structure are practical on systems with 100 atoms or so in a periodic unit cell. These quantum mechanical calculations provide electron wavefunctions within the local density approximation and its extensions. The atom position information needed by the diffraction subprojects is available from all these codes. The linear response theory for vibrational excitations needs further development.\(^{10}\) Bindings will be developed to incorporate these packages into DANSE:

- The plane wave pseudopotential package, **ABINIT**, is an open source software project.
- The commercial plane wave pseudopotential package, **VASP**, is in widespread use, and runs well on multiprocessor hardware.
- **CPMD**, a plane wave pseudopotential code particularly useful for molecular dynamics.
- The commercial full-potential linear augmented plane wave code, **WIEN2K**, is one of the more accurate codes available for calculations involving core electrons.

5.3.3. **Hartree-Fock Methods**

Hartree-Fock codes with Gaussian basis sets are used for systems of 1,000 atoms or so. These codes are quite mature, and can provide both the positional information required for diffraction studies and the vibrational eigenfrequencies required for inelastic scattering. Bindings will be developed for these packages:

- **GAMESS**, which has a gentle licensing agreement. (It is almost open source.)
- **NWChem**, which uses Hartree-Fock methods to treat molecules and periodic solids on an equal footing. It is available to scientists at no charge.

Licensing difficulties for the well-known package **Gaussian** make it inappropriate for the DANSE project at the present time \(^{32}\).

5.3.4. **Molecular Dynamics**

Molecular dynamics (MD) simulations, which can track the time dependence of the degrees of freedom for tens of thousands of atoms or molecules, often require substantial computing resources. The time information from molecular dynamics simulations is important for calculating the energy spectra of neutron scattering from the assembly of atoms, and for calculating the quasielastic scattering caused by the movement of the atoms during the scattering process. Subprojects using elastic scattering also need molecular dynamics simulations for calculating the conformations of larger atomic systems, or for calculating the thermal factors in diffraction measurements. Bindings will be developed for:

- **NAMD**, an open source parallel MD code designed for high-performance simulation of large molecular systems.
- **AMBER**, a collection of molecular simulation programs and molecular mechanical force fields. It uses a mix of **Fortran90**, C and C++, and has a small fee for academic and government use.
- **CHARMM**, which incorporates both a force field engine and the force field itself. The **CHARMM** force fields are the core building blocks of many popular molecular dynamics packages.
- **MMTK**, a **Python**-based molecular modeling toolkit designed to be extended easily to deal with both standard and non-standard problems in molecular simulations, and to interface with external molecular viewers.

\(^{10}\) This amounts to calculating second derivatives of the electronic energy with respect to displacements of the atoms in the unit cell, meaning that the electronic structure must be calculated for a number of configurations.
5.3.5.  *Scattered Intensity*

The previous materials simulations do not include the necessary extensions to calculate the intensity of neutron scattering. Some of the necessary modules have been developed, but on an *ad-hoc* basis. We propose to design and build the DANSE extensions required to calculate \( S(\vec{Q}, E) \) or \( I(\vec{Q}) \) from the positions and motions of atoms calculated with these packages. This component will be based on the Born approximation for scattering, and will follow the approach described in Section 10.3.

5.4.  **Common Algorithms and Data Types**

5.4.1.  *Data Structures*

Data containers are as necessary as data transformations, and standardized runtime data structures will be developed for DANSE. While the most commonly-used classes will be histograms, efficient runtime structures will be needed for metadata too; DANSE will provide classes that parallel the NeXus hierarchies. These data structures will satisfy two needs: to provide efficient iteration for high performance components, and to provide convenient access to scientists. It has often proved difficult to reconcile these needs, especially with histograms. DANSE will achieve these goals for histograms by decoupling iteration and access from the underlying buffer. Furthermore, DANSE developers will work with scientists to express data transformations as operations on individual, one dimensional pixels wherever possible. A preliminary survey indicates that almost all neutron scattering data reduction transformations can be performed as pixel-by-pixel calculations, which are natural for parallel computing.

5.4.2.  *Optimization Algorithms*

Neutron scattering requires frequent comparison between data and models, for both simple data reduction and for more advanced theoretical modeling. Many functions can be well fit with nonlinear least squares techniques like Levenberg-Marquardt, but when derivatives are unavailable or prohibitively expensive to compute, techniques such as Powell’s method are useful. For complex models with many local minima, simulated annealing and reverse Monte Carlo [33] are appropriate. DANSE will provide optimization components with a common interface, allowing client components to freely interchange fit method, figure-of-merit, and model.

5.4.3.  *Numerical Libraries*

Numerical libraries offer another opportunity for the DANSE project team to share development effort. DANSE components will provide FFT capabilities, and various mathematical functions will be supplied to DANSE programmers, from basic trigonometry to special functions commonly used in scattering. Other numerical capabilities will include random number generation and matrix manipulations.

5.4.4.  *Error Propagation Algorithms*

DANSE will provide standard error propagation and error origination algorithms. These will be included with the underlying data transformations by default. However, scientists will always have the option to override the default behavior at their discretion.

5.4.5.  *Crystallography*

Crystallography is a large field unto itself. There are an enormous number of software packages in the field of crystallography. We have selected a number of these packages that can serve as utilities for several science subprojects. These are:

- **cctoolbox (cctbx)** is a collection of C++ libraries (with Python bindings) for general crystallographic applications, useful for both small-molecule and macro-molecular crystallography. Cctbx is arguably the most extensive, free, open source crystallographic library available today.

- **FOX** provides *ab initio* structure determination from single crystal and powder diffraction data. FOX is built on the ObjCryst++ library, thus it is readily customizable and extensible.

- **CrysFPL** is a Fortran95 rewrite of the Cambridge Crystallographic Subroutines Library (CCSL). CrysFPL provides core routines for crystallographic programs such as FullProf and FOURIER.

- **PowderCell** provides the generation of initial structure models which can be used in Rietveld refinement procedures. At present, PowderCell runs only on MS Windows.
- **ObjCryst++** is a C++ library, which provides high-level objects to create new crystallographic programs, and is focused on the global optimization of crystal structures from diffraction data.

- **Crystal Structure Container Class** This is a partially-complete development effort to provide a description language for structures of atoms. Support for disorder and dislocations, as well as symmetry classes will be provided. A clean separation of atomic properties from crystallographic geometry, and a hierarchy to provide a generic descriptive language for the geometry of the crystal will allow reusability as a base class for new crystallographic computer programs.

### 5.4.6. Molecular Viewers and Format Translators

The Protein Data Bank (PDB) format gives information on the atomic structures of molecules, and is in widespread use in computational biochemistry and chemistry. Access to archives of molecular structures is important for the science subprojects on SANS, reflectometry, and inelastic scattering. Software will be ported to DANSE to facilitate working with PDB files:

- **Open Babel** is a cross-platform program and library to convert between the many file formats used in molecular modeling and computational chemistry. Binding OpenBabel into DANSE will greatly reduce the vast number of chemical file format parsers needed to provide interactivity between molecular dynamics and computational chemistry components.

- **Molden** is the most common molecular viewer for displaying molecular and electronic density, molecular orbitals, molecular vibrations, atomic charges, and electrostatic potentials calculated with GAMESS. Molden also includes a Z-matrix editor and molecular builder.

- **PyMol** is Python-based molecular graphics system designed for real-time visualization and rapid generation of high-quality molecular images and animations. PyMol has support for crystallographic symmetry, POV-Ray ray tracing, and includes a molecular editor and builder.

- **ViewMol** is a high-quality molecular geometry and animation viewer. It can work with IR, Raman, inelastic neutron scattering spectra, and has capabilities for periodic systems.

- **Deep View/Swiss PDB viewer** is an excellent tool for viewing several molecules simultaneously. It is limited to reading PDB files, and we may select instead a more crystallography-oriented file format viewer. DRAWxtl is a potential choice. It is written in C++, and requires approximately the same amount of effort as binding Deep View. DRAWxtl is a realtime crystal-structure viewer capable of producing Virtual Reality Modeling Language (VRML) and POV-Ray ray tracings. DRAWxtl supports structural input from CIF, CSD, GSAS and other crystallographic file formats.

- **VMD** is a molecular viewer with many similar features to Molden. It was developed for easy integration with NAMD, supports ABMER and CHARMM file formats, and includes the Tachyon ray tracer for animations.

- The development of an additional CIF reader component is needed to support other crystallographic file formats.

### 5.5. Personnel and Other Resources

The effort on common tasks will be performed within the different scientific and central services subprojects of the DANSE project including Iowa State Univ., Univ. Maryland, Michigan State Univ. and Caltech. The DANSE Project Manager will coordinate this effort with the assistance of the P.I. and administrative staff at Caltech.

### 6. Diffraction

#### § Scope of Science for Diffraction

Knowledge of atomic scale structure is a prerequisite to understanding materials and condensed matter [34]. Single crystal and powder diffraction by x-rays and neutrons are the *sine qua non* of structure solution and refinement. Diffraction methods have been applied ubiquitously in structural studies with structural complexities from sodium chloride to proteins. These techniques are interdisciplinary; used by chemists, biochemists, biologists, earth scientists, physicists, engineers and
materials scientists. They support a large number of neutron users with a relatively high publication rate due to the higher throughput than some other sub-fields. Diffraction is also widely used outside of the world of neutron scattering with many thousands of users of laboratory based x-ray diffraction systems worldwide.

The primary goals of the diffraction software subproject are to: 1) Enable qualitatively new science by creating software that exploits advances in computing power and materials theory. New software is needed for studies of increasingly complex, multi-scale, multi-component samples, such as nanostructured materials, that are of interest to scientists and engineers these days. 2) Leverage the scientific potential of neutron beamtime by providing tools to increase the efficiency of beamtime allocation and utilization and facilitating the path from data to science. 3) Expand the neutron diffraction user base to the wider diffraction community by providing structure solution/refinement capabilities in real-time (on SNS data-collection time-scales) that are robust and easy to use with a low barrier to new user adoption.

§ Neutron Diffraction Capabilities at the SNS
Diffraction instruments presently approved and under development for the SNS include powder diffractometers POWGEN3 (2007) and NOMAD (2010) for polycrystal and disordered materials studies respectively, a high-pressure diffraction beamline, SNAP (2008) and a single crystal beamline, SCD (2009). A protein crystallography beamline (MaNDi) also has been proposed, as has a correlation spectrometer for elastic diffuse scattering measurements from single crystals (Corelli). These SNS diffractometers will have unprecedented throughput. Parametric studies of diffraction as functions of temperature, pressure, or magnetic field will provide structural information with densities of data points comparable to measurements of electrical transport properties, for example. For the first time, it will be possible to extract first- and second-derivatives of structural changes with thermodynamic parameters. Of course this will only be possible if automated data processing and structural refinements can be performed on large data sets, preferably in real time.

6.1. Data Reduction
Data reduction components will be built to transform data from instrument coordinates (e.g., detector position, time-of-flight) into conventional structure functions for crystallographic analyses of powders and single crystals.

6.1.1. Structure functions from powder diffractometers
Data will be reduced to continuous 1-D diffraction patterns, \( S(Q) \) and \( G(r) \) (the Fourier transform of \( S(Q) \)). This is most useful if done in real-time during data collection therefore the software goal is to produce fully corrected structure functions within seconds.

6.1.2. Bragg-peak intensities from single-crystal diffractometers
Structure solution and refinement of single-crystal data is done with respect to a table of Bragg-peak intensities, \( I_{hkl} \). Fully corrected \( F_{hkl}^2 \) will also be determined from raw histogram data.

6.1.3. Instrument specific tasks
This involves application building for specific instruments using the components and methods developed in Sections 6.1.1 and 6.1.2. The scope includes POWGEN3, NOMAD and SCD. Working versions will be built for SEPD/GPPD at IPNS and NPDF at the Lujan Center.

6.2. Modeling and Co-Refinement
Diffraction research is expanding to materials of increasing complexity. This presents a double-edged sword because there are more degrees of freedom in models of the structure, but the number of experimental data-constraints (Bragg peak positions and intensities) is often reduced. This problem is particularly acute for studies of nanostructured materials that do not exhibit Bragg peaks from translational invariance, so conventional crystallographic analyses are not possible. Furthermore, to fully specify the nanoscale structure, the positions of a large number of atoms must be specified. Novel approaches to this problem include refining parameters from the atomic pair distribution function (PDF) and some progress has been made in this area of “nanocrystallography” [35].
The approach to solving structures of large-scale, complex materials should be to: 1) increase the information about the system by increasing the data constraints by co-refining multiple complementary data sets, and 2) reduce the number of degrees of freedom in the model by, for example, building in physical information in the form of inter-atomic potentials. Today there are software packages that permit co-analysis of x-ray and neutron diffraction data, but these are just a start at what is possible. These analyses should be expanded to include modeling and simulation on an equal footing. For example, structural degrees of freedom can be removed by building into a model some physical knowledge such as distance constraints, avoidance of atomic overlaps, and the specification of empirical potentials. An excellent example of this is the recent success in refining the structures of simple proteins from powder diffraction data where the positions of thousands of atoms were specified by Rietveld refinement [31] with a few hundred Bragg peaks [36].

Making use of the DANSE component architecture, we will design an extensible modular structure refinement system where multiple data-sets can be refined jointly and where constraints and restraints can be easily incorporated, including user-specified constraints. The design will allow data to be co-refined in real- and reciprocal-space and x-ray and neutron data-sets to be jointly refined. The software will include unit and functional tests, clear documentation, including pseudo-code and extensive comments, which will facilitate future extensions.

6.2.1. Full Profile model refinement in real and reciprocal space

Existing functionality will be taken from the years of development that have gone into GSAS [37] and FullProf [38] for reciprocal-space refinement, and PDFFIT [39] for real-space refinements. FullProf and PDFFIT are open source and we have copies of the GSAS sources and permission from Dr. Von Dreele to use them. The programs will be rebuilt from the ground up, however, using an extensive modular design along the lines of the well-regarded Topas [40] program. Component cores will be written in a compiled object oriented language (e.g., C++ or FORTRAN95) and optimized for speed to allow for “real-time” refinements. The component cores will be designed to allow them to be used within the DANSE framework but also to function outside the framework in the future with little modification. The power of distributed computing offered by the DANSE framework could be important for real-time operations with large matrix inversion components being executed on cluster computers, for example.

The modular extensible design will allow joint refinements; for example, joint real and reciprocal space refinements, to occur in one single package. This is an essential extension to existing capabilities allowing a more stable solution of complex structural problems which are becoming the norm. The scope includes components to read and build crystal structures from CIF files, calculate powder profiles, PDFs and single crystal intensities as well as least-squares regression capabilities. It also includes application building of an integrated, easy to use full-profile regression modeling program that incorporates both Rietveld and PDFFIT functionalities. Attention will be paid to making these application programs robust and straightforward to use to lower the entry barrier. More extensible modeling applications, of interest to expert users who desire a great deal of control, will be provided in separate tasks described below.

This is, by far, the largest task of this sub-project and is the backbone of our effort. A complete rebuild and integration of these refinement codes is necessary to tackle the complex structural questions of today and the unprecedented volumes and rates of data that will be coming from diffractometers at SNS. We will pay attention to enforcing standards of clarity, modularity, commenting and incorporation of testing code so that these codes become the basis for future developments beyond the scope of time-of-flight neutron data contained in this proposal. This task in the WBS is to develop the core functionality. Below are tasks addressing specific extensions.

6.2.2. Plug-in architecture for carrying out joint refinements

The scope is to develop plug-in components that allow for carrying out the regression using different algorithms than the conventional downhill least-squares method; specifically, Monte Carlo, genetic algorithms, and user defined methods. Also, an extendable plug-in architecture for refining different functions will be developed. The scope includes plug-ins for total scattering ($S(Q)$ and $G(r)$), neutron differential PDFs and TOF neutron diffraction intensity functions. Many of the components will be
developed under the common algorithms effort. Here we will design and test the protocol for handling
the plug-ins and make modifications to the components for incorporation into our applications.

6.2.3. Model independent peak fitting
Components will be built for carrying out model-independent decomposition of powder diffraction
patterns using both Pawley [41] and LeBail [42] methods.

6.2.4. Rietveld and PDF modeling extensions
This task specifically addresses the need to refine magnetic and incommensurate structures. Also,
components to allow user defined background and profile functions will be developed. These will
be implemented by taking advantage of analytic math modules, as has been demonstrated with the
successful Topas program.

6.2.5. Constraint/restraint plug-ins
We will build components to allow straightforward incorporation of constraints and restraints (soft
constraints) in the modeling cycle. Components will offer molecular constraints, rigid body con-
straints and analytic user-defined inter-parameter relationships.

6.2.6. Post modeling analysis tools
In this task components for post-analysis of refined models will be developed. These are a bond-
valence-sum calculator for bond-lengths and angles in crystals and glasses, and a component for
calculating all partial PDFs of model configurations.

6.2.7. Graphical applications
Most graphical and GUI developments will be handled in central services tasks and in collaboration
with the SNS. We will work with SNS instrument scientists to adapt DANSE components for the
modeling programs. The plan is to have a desktop environment for visualizing the data and the
structural model in real-time as the data are being collected. This should be interactive to allow
control of the refinements and, in principle, also the data collection. We would also like automatic
feedback control so that data collection can be terminated and a new run started when a certain
refined parameter is known with a pre-specified precision.

6.2.8. Refinement control toolkit
Rietveld refinement codes are notoriously tricky to use and there is a steep learning curve to their use.
There is a conflict between ease of use (push-button operation) and user control (expert user mode).
We propose to build inventories of default settings for different situations, and build a GUI layer over
this for novice users. For example, once an instrument and its geometry is specified (this information
will be in the NeXus file) many default initial settings can be reasonably set automatically; also
folklore about which parameters to refine in which order can be built in automatically. This task
will build the framework and the GUI for this task and start an inventory of defaults for different
situations. These can be updated later. Expert users will have complete control to override the
defaults as they wish, to make a user-defined set of defaults, or to run the program without this
layer.

6.3. Direct Experiment Simulation
We propose to start developing Monte Carlo simulations of entire experiments using realistic in-
strument and sample geometries, with the sample differential scattering cross-section as the kernel.
Putting this in a regression loop to refine the cross-section is impractical today due to inaccuracies
inherent in the simulation and the computational expense. This is a promising direction because it
could take proper account of background scattering from sample environments (which are difficult
to correct because they change depending on whether the sample is in place or not) and effects such
as sample transparency and multiple scattering. For the present, we see this as an important tool
for efficient beamtime allocation and education of new users.

We will build an interface between our modeling applications and Monte Carlo simulation com-
ponents developed as common components. Our plan is that the simulated experiment could be used
as input to the desktop environment. An upcoming experiment could be simulated at the user’s
institution before the actual experiment takes place. This has practical consequence that the users
become familiar with the software and are able to collect all the ancillary information about their sample ahead of time, set up refinement defaults etc. before coming to the experiment. They will be modeling real data in a few minutes as it starts to come in during the real experiment instead of having this process take hours, which is incompatible with the data-collection times on instruments at the SNS. Investigating capabilities of simulation methods for data analysis is proposed as a graduate student thesis project.

6.4. Personnel and Other Resources
The diffraction subproject will be centered at Michigan State University and led by Simon J. L. Billinge. It includes an international consulting group of 40 individuals who are either neutron diffraction instrument scientists or actively involved in diffraction software development. Besides the US diffraction-instrument scientists, it includes instrument scientists from ISIS in the UK and ILL in Grenoble and members of the development teams of the highly regarded ISAW, GSAS, FULLPROF, EXPGUI, TOPAS, EXPO and FOX software packages. The cooperation and support of these developers is critical. This group has been polled for its suggestions and has already provided valuable feedback on the scope of the proposed project. As software developers themselves, it is also hoped that they will become actively involved in a community development activity as the project develops.

7. Engineering Diffraction

Scope of Research
"Engineering Diffraction" refers to a sub-field of neutron scattering which investigates microstructural features that influence the mechanical properties of materials. The field of engineering diffraction has experienced tremendous growth during the past two decades. Using TOF powder diffractometers such as NPD at LANSCE, GPPD at IPNS and ENGIN at ISIS, numerous studies were performed that can be classified approximately (with examples cited as typical cases) as: 1) Residual stress measurements. Neutron diffraction is ideal for such experiments because it can measure distortions in all phases simultaneously, and obtain information about either the bulk average or spatially resolved residual strains/stresses [43, 44]. 2) Deformation studies. The effect of applied stress on phase dependent internal stress/strain evolution has been studied to deduce deformation mechanisms [44, 45, 46, 47]. 3) Texture analysis. Owing to the elastic and plastic anisotropy in most materials, texture (a measure of crystal orientations) is an important parameter that affects deformation. Texture evolution during and after deformation has been studied by numerous researchers, e.g. [46]. 4) Phase transformations. Phase changes have been studied during thermal cycling, during reactions or transformations [48], around welds, and under applied stress. 5) Dislocation structure. Dislocation density has been determined from the analysis of diffraction peak widths [47].

Recently, dedicated engineering instruments such as SMARTS [49] at LANSCE and ENGIN-X [50] at ISIS have been constructed, adding unprecedented capabilities. It is now possible to perform detailed measurements in minutes under a variety of conditions: applied stresses in excess of 3 GPa, temperatures above 1500°C, and sampling volumes as small as 1 mm³. The planned engineering diffractometer at the SNS, VULCAN, will be a third-generation instrument that will offer even more impressive capabilities owing to improved flux and detector efficiency. It will permit detailed stress mapping, parametric studies, studies of texture, kinetics of phase transformations and in-situ monitoring of processing. However, the engineering diffraction community cannot use these new capabilities adequately without new software for experiment planning, data acquisition, and analysis.

In a typical engineering neutron diffraction experiment, elastic lattice strain is measured by tracking the shifts of Bragg reflections. A careful micromechanics calculation is a crucial step in calculating stress from these strain data. For example, if one assumes an isotropic continuum, a rigorous set of equations can be derived to predict the complete stress/strain tensors from six orthogonal lattice parameter measurements [51]. Unfortunately, the heterogeneous nature of the stress/strain response of polycrystalline materials on a local scale, coupled with the selectivity imposed by the Bragg’s law, may completely invalidate the continuum-mechanical treatment [52]. Therefore, considerable
thought should be given to the design of an experiment to ensure that the data acquired are in fact compatible with the requirements of the analysis.

7.1. EXPERIMENTAL SETUP AND DATA REDUCTION

7.1.1. Experiment Planning and Setup

Large experimental errors occur unless the sample positioning is accurate to 10 $\mu$m [53]. Today, sample setup accounts for a loss of beamtime of over 20%, and this loss would be fractionally greater for a more efficient instrument such as VULCAN. Software to assist with sample positioning would enable more efficient use of the instrument. A new software package developed in IDL at ISIS and the Open University, UK, called SScanSS offers powerful tools for sample setup and experiment planning [54]. It will be developed further and adapted for use on VULCAN and SMARTS. For instance, SScanSS currently lacks tools that permit efficient experiment design. Extensive trial runs are often conducted before the details of the “real” experiment are determined. Simulations to integrate mechanics modeling and formal experimental design principles [55] would significantly improve this efficiency. Another important role of experiment planning is to estimate the minimum number of measurements required to achieve a desired accuracy for the mechanics information about the sample. These capabilities will be developed in Python, and added to SScanSS, which will run under the IDL package on the DANSE framework.

7.1.2. Data Monitoring and Assessment

Real-time data analysis is crucial, especially for modifying experimental plans after unexpected results, or stopping data collection when sufficient counting statistics are obtained (when further data collection would be a waste of beam time [56]). It is often the case that the instrument beamtime would have been used differently, or better optimized, if micromechanics calculations could have been compared with the data while the measurements were underway. For experimenters to benefit from real-time computing, the diffraction data need to be displayed promptly in a multidimensional format. For example, volumetric displays of stress or strain as a function of a three-dimensional sampling volume are necessary for residual stress mapping around a weld. We propose to build such capabilities into the DANSE system.

7.2. MECHANICS MODELING

The growth and future success of engineering neutron diffraction requires the capability to better integrate data with mechanics models. Neutron diffraction can only measure elastic lattice strain, so solid mechanics calculations are a crucial step to obtain stresses from the strain data. The problem is complicated when strain is measured in a multiphase composite with load sharing between phases. Additional difficulties arise in polycrystalline materials due to local heterogeneities in the microstructure and the selectivity of the Bragg’s law mentioned earlier. Recent applications of engineering neutron diffraction have increasingly sought to compare diffraction data with mechanics models. However, almost all of the modeling to date has been performed after the experiment. In many cases, the comparison of the data with model predictions has then required additional experiments, or has shown that the previous experiment was executed improperly. One of the central goals of the engineering neutron diffraction sub-project is to develop tools that will effectively integrate mechanics models with diffraction data and allow real-time comparison between the two.

7.2.1. Finite Element Modeling

Finite element models (FEM) offer valuable stress/strain information in samples, and these models are relatively easy to construct with sophisticated software such as ABAQUS [57]. The output of this software can easily be configured to allow direct comparison with neutron diffraction data. A special advantage of FEM is that it generates spatially resolved stress/strain data so that diffraction studies of, e.g., residual stress in weldments, can benefit enormously from its predictions. During the design phase of the DANSE project, ABAQUS was integrated into the pyre framework and was demonstrated to run seamlessly. We propose to further integrate ABAQUS by developing tools to study standard material and specimens by novice users, while allowing experienced modelers to incorporate complicated, existing models.
FEM can also be used to study micromechanical deformation, and such studies in particular are computationally intensive, so access to high-performance computing resources will be essential. Typical two-dimensional finite element calculations for today’s micromechanics research take 10–100 hours on a single processor. Three-dimensional calculations of polycrystalline materials can be performed only with massive parallel computing, as in the DOE ASCI effort at Caltech or as has been demonstrated by Dawson and co-workers [58].

7.2.2. Self-Consistent Modeling
Most finite element models lack microstructural detail that can be provided by self-consistent models (SCM). They are an extension of the classical Eshelby approach and have been used extensively over the last decade to interpret engineering neutron diffraction data [45]. An important task in the engineering diffraction sub-project will involve the adaptation and integration of existing SCM codes into the pyre framework. The first of these is the EPSC code from Los Alamos, [59] which deals with crystal plasticity in polycrystals. Another, more recent code simulates the electromechanical response of polycrystalline ferroelectrics [60, 61]. The latter is significantly more intensive computationally, and will benefit from the distributed computing capability of DANSE.

7.2.3. Inverse Problem Analysis
The uniqueness of the results is often a critical concern of engineering diffraction research. For example, a self-consistent model utilizing diffraction data can deduce the in-situ constitutive law of fibers in a composite. This is done today by trial-and-error, using judicious guesswork of important deformation parameters such as hardening coefficients in the plastic regime, and modifying the model until it fits the data. This approach is obviously not rigorous, and more importantly it fails to determine the uniqueness of the final result and its bounds. The engineering neutron diffraction sub-project will initiate a systematic effort to develop software tools to perform inverse problem analysis for model optimization, including estimates of parameters in the models, their bounds, and their uniqueness. A promising approach by Bowler et al. [62] involves numerous model calculations, necessitating powerful computing resources. Analysis tools for inverse problems will be developed for both finite element and self-consistent model calculations. The tools will also help to estimate the minimum number of data points required for a given accuracy, allowing more efficient experiment planning.

7.3. Experiment Simulation
An experiment plan requires a realistic estimate of data collection time. The simulations of experimental data, which consider both instrument optics and scattering properties of the samples, must be integrated with materials modeling tools, e.g. finite element software such as ABAQUS [57]. Simulations, either archived or performed in real-time, will be invaluable for guiding experimental measurements. A significant effort will be spent in the engineering neutron diffraction sub-project to perform and integrate direct instrument simulations with materials models.

7.3.1. Instrument Simulation
Using McStas and IDEAS (the former already adapted to DANSE), the instruments central to this sub-project (VULCAN and SMARTS) will be simulated to study the effects of instrument parameters (e.g., beam divergence, collimator settings, detector location) on data quality (e.g., peak position and profile, and strain error). These simulations will be coupled with experimental studies to validate the simulations. The latter was initiated in the design phase of the DANSE project and revealed interesting effects. For example, it was shown that beyond a certain crystallite size, dynamical diffraction effects play a significant role in determining both peak position and profile, which in turn can result in large systematic strain errors [63].

7.3.2. Microstructure Simulation
The next critical component in effective experiment simulation is the realistic modeling of specimen microstructure, which has not been accomplished adequately to date. In this task, software tools and materials models will be developed to realistically evaluate the effects of microstructural parameters such as grain size, grain shape, texture, dislocation structures, on diffraction peak profiles and positions. This task will be a graduate student thesis research project to obtain or derive the equations
that link the neutron peak shape and position to the sample and experimental parameters. These equations will be validated by experiments at various sources, and will be integrated with work on McStas and IDEAS.

7.4. Subproject Personnel

The principal investigator of the engineering diffraction effort is Ersan Üstündag, Iowa State University and DOE Ames Laboratory. An executive committee of experienced researchers in the field was formed to guide the subproject. This group includes Xun-Li Wang (instrument scientist for the VULCAN instrument at the SNS), Cevdet Noyan (Columbia University), Mark A. M. Bourke and Bjørn Clausen (LANSCE, Los Alamos), James W. Richardson, Jr. (IPNS, Argonne), Lyndon Edwards (Open University, UK) and Mark R. Daymond (Queen’s University, Canada, and ISIS).

8. Small-Angle Neutron Scattering

§ Scope of Science for SANS

Small angle neutron scattering (SANS) provides information about the shapes, sizes, and orientations of particles or mesoscopic features in materials, as well as information about particle arrangements. SANS uses relatively long wavelength neutrons (typically 4–20 Å) scattered at small angles to explore the structure of matter at the scale of 1–100 nanometers. The more recent technique of ultra small angle neutron scattering (USANS), using crystals to resolve much smaller angles, extends the accessible size range up to microns.

SANS has attracted perhaps the largest and most diverse set of neutron scattering users from physics, chemistry, engineering, metallurgy, and biology. SANS measurements are performed by users from industry, national labs, and academia. The materials studied recently by SANS are frequently complex fluids, polymeric materials, and biological materials, but other materials science research includes topics such as magnetic flux lines in superconductors and precipitates in alloys. This diverse range of science could help increase the number of users of the SNS.

A coherent, easy to use, and well documented suite of tools is needed for new or occasional users of SANS. Perhaps more so than for other methods of neutron scattering, good software could enable an explosion of new science from many important areas, the biological sciences in particular, that are effectively excluded today. If done well, this growth could parallel the growth seen in biological x-ray crystallography. A guiding principle for this subproject will be to design software with intelligent default parameters and procedures to allow access by novice users, without compromising the ability of the expert to perform unusual or unique analyses.

8.1. Data Reduction

The traditional reduction capabilities for converting detector counts into $I(Q)$ will be delivered for SANS instruments at NIST and the SNS, and will be made available for adaptation at SANS instruments at HFIR, IPNS, and Lujan. Reduction components will be delivered to process intensities from 2D data sets, sometimes measured at multiple steps in time or temperature, for example. Data reduction tasks require the adaptation of tools developed in Section 5.1, but with an integrated approach for the different types of SANS instruments, ensuring that SANS users will have consistency between different facilities, especially the SNS, Lujan, NIST, and HFIR.

USANS extends SANS measurements to much lower $Q$, but its slit geometry makes it difficult to compare USANS data to SANS data. New reduction software is needed to facilitate this comparison. Finally, with the new interest in energy-resolved and time-resolved SANS, alternate schemes for data reduction will be needed. Energy-resolved SANS is essentially an inelastic scattering technique, and extends SANS beyond traditional structure analysis into dynamics studies. Data analysis for inelastic SANS is beyond the present scope of the DANSE project, but if enough of the essential tools become available on the DANSE framework, we will consider adding such a task in the future.

8.1.1. Steady-State Reduction

At the end of each run, an intensity is obtained for each pixel on a 2D detector. This also includes timing information if choppers are being used. Software components will be delivered to reduce...
these data to $Q$-space, with the necessary corrections for flux, background, detector artifacts, and effects of sample transmission. To date this process is not fully automated, in part because accurate information must be given for sample transmission and thickness. Software alone cannot solve these problems, but giving the user better information online can help. The reduction software will provide the user with better documentation, and consistency checks for input parameters.

8.1.2. **Time-of-Flight Reduction**

Data reduction is similar for both time-of-flight (TOF) and steady state instruments, but time-of-flight data are more complex. A fundamental difference for TOF data is that transmission is no longer a constant, but is wavelength dependent. For the TOF reduction software, we propose to use wavelength information to calculate transmissions for the different time bins of the detector pixels.

8.1.3. **Ultra Small Angle Scattering (USANS) Reduction**

USANS extends SANS data to even lower $Q$, but the instrument design is quite different. The crystals used in the Bonse–Hart instrument configuration achieve high intensity in one direction at the expense of resolution in another direction. USANS data are essentially convoluted with the resolution function of a 1D slit. Further, the stability of these instruments is notoriously poor, and a full fit to the rocking curve scan of each run is often needed to accommodate shifts in the data. It is often desirable to desmear the data for quick comparisons to SANS data, but fitting routines must alter model calculations with the instrumental smearing. DANSE will provide USANS reduction software components to perform these different types of data reduction, and can be interchanged quickly on the framework through the user interface.

8.1.4. **Batch Reduction and Visualization**

A typical SANS instrument can produce hundreds of sets of data in the few days typical of beamtime allocations. Methods for automatically reducing all the data simultaneously are necessary for any modern SANS reduction package, and this must be done while providing visual feedback to show any problems in the data sets or reduction procedure. The tools described in Sections 4.3.1 and 4.3 will facilitate the development of SANS capabilities for batch reductions and parametric reductions.

8.1.5. **Experimental Planning and Optimization Tools**

Experimental planning and optimization tools offer real promise to increase the productivity of beamtime allocations by helping plan the runs, and identifying unfeasible measurements. A large return is expected from the investment in tools as simple as a handy SLD calculator (Section 5.2.2). Other important tools to be developed here will predict collection times based on fluxes on sample and required statistics, using candidate structures for the sample. Simple model-based calculations for these predictions will be integrated tightly to the user interface, and also made available as independent components.

8.2. **Analysis**

After reduction, SANS data of $I(Q)$ usually do not provide unambiguous solutions for material structures, since they are a measure of the average bulk structure in a certain plane. A sphere and a rod oriented along the direction of the incident beam look essentially the same. A unique feature of SANS analysis software is the need to include documentation at several levels of explanation to ensure that data are not over-interpreted. This instructional component of the SANS subproject would be especially helpful for new users.

8.2.1. **Standard, Non Shape Driven Analyses**

Properly reduced and normalized data can be used for standard, model-independent analyses such as Guinier fits, Porod analyses, and power-law fits. Although limited in the detail they can provide, these standard analyses give important insights into the material without many a-priori assumptions, making them invaluable for looking at new systems. These analyses will be provided, utilizing the least-squares components of Section 5.4.2.
8.2.2. 1D Model Fitting
The main method for SANS data analysis over the past several decades has been model fitting. A number of simple shapes such as solid sphere, ellipsoids of revolution, rods, discs, even squares of uniform density all have analytical form factors that have been derived for random orientation distributions. For non-particulate systems, the random phase approximation has been developed, and has been most successful for understanding polymeric systems [64, 65, 66], as have chord analyses. Many of these functions have been collected in an Igor software package by the NIST group. The core functions in these packages are quite short, and we propose to re-implement them with object-oriented programming. This quick task will also make it much easier to maintain the model functions, and add new ones later.

8.2.3. 2D Model Fitting
Anisotropic systems are increasingly of interest to SANS researchers. There are many examples in the literature of 2D anisotropic data sets that are presented qualitatively because of a lack of access to appropriate data analysis software. We propose to develop software to calculate directly the full 2D SANS data, including data from oriented particles.

8.2.4. Inversion to \( P(r) \)
Another way to process SANS data is to transform it mathematically into real-space. Fourier transformation of SANS data into real space coordinates provides a distance distribution function \( P(r) \), analogous to a radial distribution function [67]. We will adapt the components described in Section 6.2.1 for the transformation of SANS data.

Fourier transform methods have problems with statistical noise in experimental data, however, and other approaches can be appropriate. Software such as that of Otto Glatter or Demitri Svergun is a step in this direction, and is especially valuable for expert users who can add constraints to the fitting procedures. For non-experts, this method may prove useful as a survey tool to help figure out what models might be appropriate. We propose to port such a software package to DANSE. Reasonable default values, good documentation, and on line help are essential.

8.2.5. Series Analysis and Simultaneous fitting
The need to automate routine tasks and better see the results of these analyses is particularly important in SANS, with its large numbers of data sets and its diversity of users. Many investigations involve analyzing hundreds of SANS runs to measure structural changes as a function of a parameter such as shear rate, temperature, time, or magnetic field. Other analyses require the use of simultaneous fitting of a single model to a series of data sets. Such parametric analyses and simultaneous fitting will become increasingly important with the high throughput of the SNS and HFIR SANS instruments. We will build on the components described in Section 8.1.4, and ensure that the batch analysis procedures are consistent with the data reduction procedures.

8.3. Simulations and Other Advanced Analysis Tools
Beyond the essential tasks to ensure that today’s science can be done with DANSE software, there is an even greater opportunity for providing more advanced analysis tools such as simulation and modeling techniques, and integration with information from other techniques. The SANS simulation software will utilize many of the components described in Sections 5.3.3, 5.3.4, and 5.4.6 for calculating the scattering from molecular dynamics simulations, for example.

8.3.1. Simulations of Complex Systems
There are a variety of systems, including polymeric systems, where simple form factor analysis is difficult or impossible. Some of the simulation packages of Section 5.3.4, such as NAMD, are well-suited to the simulation of SANS data, utilizing scattering components developed in Section 5.3.5 and other components in Section 10.2.5. Some adaptation and experimentation with these components is proposed for SANS simulations, after which test scripts will be generalized for friendly users.

8.3.2. Real-Space 3D-Oriented Models
SANS research emphasizes shapes and their transformation into \( Q \)-space. Specifying 3D shapes in real-space can be challenging. The solution proposed in Section 4.2.2 is to give users a visually-interactive 3D model builder that offers deformable predefined shapes to build arbitrarily-complex
models. Other software components fill the shapes with scattering length density distributions, and calculate the scattering pattern for any orientation of the object(s). Objects on the 3D canvas are accessible to Python, and this allows redrawing the visualized object as fitting is performed. We propose to develop test cases for using these capabilities with SANS data, and adapt their control scripts to the user interface. This will also be an outstanding educational tool, allowing a beginner to see instantly how various shapes and orientations affect the scattering.

8.3.3. Constraining Fitting by Use of Data From Other Experimental Techniques
Making structural models by combining SANS data with constraints from other data is an area that is still in need of development [68]. Clearly missing is the ability to fully incorporate and automate the use of homology information from the ever-increasing number of crystal structures available in the Protein Data Bank, rigorous molecular dynamics, NMR spectra, Monte Carlo simulations, and information from other low-resolution techniques, such as fluorescent resonance energy transfer, to further refine the structural models. Access to PDB files will be provided by DANSE components developed in Section 5.4.6, and some components for constrained analysis will be provided by components described in Section 6.2. An integrated tool for this type of constrained analysis would be a leap forward for model generation from small-angle scattering data, and would be a boon to the biological scattering community in particular. We propose to build test cases with the components above, and adapt them to at least one SANS research problem.

8.3.4. Ab-Initio Modeling
The direction that may have the greatest potential for rapid progress, however, is model generation from scattering data. There has been a great deal of work done on ab initio model fitting using arbitrary shapes [68, 69, 70], although there is no readily available program for ab initio modeling of shapes with non-uniform densities.

8.4. Integrated User Interface
8.4.1. Integrated User Interface Design
To maximize the utility of software for SANS data reduction and analysis, particularly for the large community of non-experts that use the SANS technique, user interface design is a top priority. User interface development could consume arbitrary resources, but is facilitated in the DANSE project because the essential components for constructing user interfaces are to be developed in the Central Services efforts of Section 4.2 and 4.3. The deliverables from these efforts will allow the SANS subproject to quickly design, assemble, and test different types of user interfaces, and modify them as needed. For data reduction, user interface development will start with a presentation layer designed to include the more effective features of existing software, and will accommodate changes after an α-release. Data analysis components should not stand alone, but must interoperate with each other intuitively.

8.4.2. Integrated User Interface Testing
A testing plan for the user interface will be developed through consultation with the staff at the NIST Center for Neutron Research. The plan will engage user volunteers, and monitor their progress with the analysis of data acquired in the course of their own research. These observations and feedback from the users will guide the final design of the user interface.

8.5. Personnel and Other Resources
Paul Butler will lead the SANS subproject. He is an adjunct faculty member at the University of Tennessee and the leader of the macromolecular and microstructure sciences team (SANS group) at NIST. He will work closely with a leadership team including Dean Myles, director of the ORNL center for structural and molecular biology (which is constructing one of the new SANS instruments at HFIR), and Sean Langridge, team leader of the large scale structures group at ISIS. Both Prof. Mark Dadmun, an expert SANS user and polymer scientist at the University of Tennessee with substantial interest in Monte Carlo work, and Dr. William Heller, a bio-physicist with the CSMB at ORNL who has worked with Jill Trewhella at Los Alamos and Huey Huang of Rice who has written bio modeling code for SANS, will be involved in the project.
9. Neutron Reflectometry

§ Overview
Neutron reflectometry uses neutrons scattered at grazing angles from a flat specimen to probe its chemical layer composition and/or magnetic depth profile [71, 72]. Analogous to classical optics, neutrons striking the surface undergo refraction and reflection if the neutron refractive indices on opposite sides of an interface are different. Because of its general utility for a wide range of materials and structures, reflectometry is in high demand within the neutron user community. One or two reflectometers exist at each user facility in the U.S., and two reflectometers will be among the first working instruments at the Spallation Neutron Source.

9.1. Data Reduction

9.1.1. Data Transformations
Data reduction includes some procedural steps that depend upon the geometry of each instrument. For example, time-of-flight data differ from those taken at a reactor source. Also, neutrons may be counted by a pencil detector, or by 1- or 2-D position sensitive detectors. All data reduction procedures are similar, however, and include the following steps: 1) raw-data collection and attenuator normalization, 2) addition of multiple data files, 3) background subtraction, 4) normalization to incident intensity, 5) polarization efficiency corrections if polarization analysis is used and 6) “footprint” or sample illumination corrections. A good software design will keep these general features separated from instrument-specific ones. Reflectometry reduction procedures change over time as instrument scientists devise new ways of extracting more information from the beam. Accommodating these changes will require the flexibility of a component based architecture so that new techniques can be developed without major restructuring of the analysis codes.

9.1.2. Direct Inversion
With perfect information about reflectivity amplitude and phase, and with certain constraints on the types of systems, we can uniquely invert the reflectivity signal as a scattering length density profile. Because a reflectivity measurement only provides information about amplitude and not phase this procedure is not in common use. However, with a set of samples in which the film of interest is identical but the fronting or backing material is changed it is possible to reconstruct the phase information for the reflectivity signals and then perform the inversion.

9.1.3. Reduction User Interface
Performing a complete reflectometry reduction requires coordinated analysis of many files. Moving these files smoothly through the reduction process while monitoring the quality of the data requires a specialized user interface. This task will be made considerably easier with a rich standard widget set and other application support tools as described in Section 4.3.

9.2. Analysis
Analyzing specular reflectivity data is a tedious and facility-dependent task. This must change if the full potential of reflectivity methods is to be realized, particularly since the systems under consideration require much more sophisticated models, and the throughput of the SNS instruments will be much greater than in the past.

9.2.1. 1-D Models
With a component-based runtime environment, users can quickly set up a sophisticated analysis of their data set. The simplest analysis system has a model component, a data component and a fit component. The model component specifies a set of layers with a scattering length density profile for each layer [73], or it could be a free-form model whose profile is composed of splines [74]. The data component contains a range of $Q$ values for the reflected intensity and resolution. The fit component contains a list of model parameters to be adjusted and constraints amongst the parameters. Running the fit component adjusts the model parameters to minimize the difference between data and model. More complicated systems might have two model components and one data component, and a mixing component to add the resulting model intensities incoherently. The
fit component in this system could adjust the mixing proportion and the parameters of either model. These components will also underlie the analysis of parametric data sets, where fit parameters are varied across a set of experimental data acquired over a range of temperatures, for example. Using DANSE scripting capabilities (Section 4.3.1) users can add new components for highly specialized models.

9.2.2. Analysis User Interface
Managing simultaneous models with shared parameters in a convenient and flexible manner will be a challenging task. Particularly for magnetic experiments, many datasets are involved.

9.3. Direct Experiment Simulation

9.3.1. 3-D Models
An exciting direction within the larger DANSE project is the direct modeling of physical systems. Using 3-D modeling tools as discussed in Section 4.2.2, users can build and view a model of the physical distribution of atoms in a material. Together with a scattering kernel to convert the atomic structure into reflectivity, and accounting for instrument characteristics, the simulation would show how changes in the model are manifested in the reflectivity data. Simulations on a larger spatial scale could be utilized in a similar way. For example, a component that calculates the distribution of cholesterol molecules from a rafting model would allow users to manipulate directly the parameters of the model and immediately see the change in the resulting reflectivity. Such a tool will aid both the design and the analysis of experiments.

9.3.2. Off-specular Modeling
Measurements of diffusely scattered neutrons with or without polarization analysis provide an unprecedented level of insight into the in-plane structures of magnetic and non-magnetic thin film systems. For biological materials, off-specular scattering will provide information about the secondary and tertiary structures of small proteins vectorially oriented in single-membrane systems. For magnetic materials, off-specular measurements will probe magnetic domains in patterned structures with potential applications as high-density storage media. Theoretical treatments for the quantitative analysis of diffuse (non-specular) neutron scattering have been developed in recent years [75, 76] and it is possible, in principle, to determine length scales associated with chemical inter-diffusion, structural roughness, magnetic roughness and magnetic domains across the plane of the film. However, only a few software packages have been developed for the analysis of off-specular data [77] and their scope is limited. This area of development remains active, especially with the improvements in capabilities of neutron reflectometers for efficient measurements of diffuse scattering. With 3-D structural models from the DANSE common components, the DANSE framework provides an opportunity to analyze off-specular reflectivity directly. While not efficient enough for refining the model, this provides a useful cross check that the approximations made in the diffuse scattering calculations are reasonable.

9.3.3. Detector Simulation
The reduction process in data analysis always removes some information. For example, wavelength and angular dispersion in the neutron beam means that the reflected signal will be spread across the detector. By simulating the data on the detector we get a more accurate assessment of the effects of instrument components and measurement procedures on the resulting data.

9.3.4. OOMMF
The object oriented micromagnetic framework (OOMMF) package provides micromagnetic simulations of 2-D and 3-D spin configurations. This software is designed to determine the characteristics of in-plane magnetic domains in thin films with competing magnetic interactions. Users can input known magnetic characteristics for their multilayer or patterned film and view the resultant magnetic structure. By porting OOMMF to the DANSE framework, the user can convert the calculated structure into specular and off-specular reflectivity and determine how changes in the magnetic interactions are manifest in the data.
9.3.5. \textit{GROMACS}
While similar to NAMD and other molecular dynamics simulations packages, \textit{GROMACS} is used particularly for coarse-grained models in the biology community. Atomistic simulations from NAMD can take on the order of a day per nanosecond per node on the relevant models. Porting \textit{GROMACS} to the DANSE framework will allow orders of magnitude speedup over NAMD for systems which can be approximated by coarse-grained models.

9.4. \textbf{Personnel and Other Resources}
Paul Kienzle and Robert Briber will lead the reflectometry effort. Algorithm development will be provided by postdocs and graduate students from the University of Maryland. They will be supported by a programmer to implement the algorithms efficiently within the DANSE framework, and provide proper testing and code documentation. We will have the services of a technical writer in the middle of the project to create tutorials and other user level documentation. Scientific advice will come from the reflectometry community, especially from the SNS and NIST. In addition we will have an international reflectometry committee representing the needs of other groups around the world.

10. \textbf{Inelastic Neutron Scattering}

\textsc{\textbf{Scope of Inelastic Neutron Scattering at the SNS}}
Inelastic neutron scattering is used to study dynamical processes in materials, molecules, and condensed matter. The experiments typically measure quantized excitations with energies from several $\mu$eV to several eV. The neutron gains or loses energy after scattering if it changes the state of the atoms, electrons, or electron spins in the sample. The dynamics information can be especially detailed when the inelastic scattering is coherent, and there is a correlation between the energy transfer, $E$, and momentum transfer, $Q$ \cite{78, 79, 80}. Analysis of the inelastic scattering intensity, $S(Q, E)$, was used to discover phonons and magnons, for example. Today the research topics for inelastic neutron scattering include: 1) spin correlations in magnets, superconductors, and materials close to metal-insulator transitions, 2) vibrational excitations in solids and their relationship to phase diagrams and equations of state of materials, 3) tunneling and reorientation transitions of small molecules on surfaces, 4) vibrational spectra of molecules or individual atoms such as hydrogen, and 5) diffusional motions and relaxation processes studied by quasielastic scattering.

Inelastic neutron scattering is usually weak and diffuse, so inelastic experiments to date have been seriously constrained by low flux. The SNS promises a qualitative change for inelastic neutron scattering experiments. It is an exciting time to plan for the new types of experiments and detailed measurements that will be possible with two or three orders of magnitude more efficiency at the SNS. Five (perhaps six) new inelastic spectrometers at the SNS will open the door to new types of investigations. The two Fermi chopper spectrometers, ARCS and SEQUOIA, are designed for studies of magnetic and vibrational excitations in materials and condensed matter, especially excitations that have coherence in space and time. Two other instruments, the Cold Neutron Chopper Spectrometer (CNCS) and the Hybrid Spectrometer (HYSPEC), address some of the same phenomena, but CNCS has a larger focus on chemical spectroscopy, and HYSPEC is designed to work with polarized neutrons for detailed studies of magnetism. Chemical spectroscopy investigations, such as rotational tunneling transitions of molecules adsorbed on a surface, will be a large part of the science program for the high-resolution Backscattering Spectrometer. This instrument is also appropriate for quasielastic scattering studies of relaxation phenomena in biophysical or complex materials. A sixth inelastic instrument, an inverse geometry spectrometer, may be built to cover the higher energies of interest in chemical spectroscopy.

The ongoing ARCS software project was planned around the scientific programs of ARCS and SEQUOIA. It overlaps some of the science of HYSPEC and CNCS. The ARCS software effort is not appropriate, however, for much of the scientific programs of CNCS, the Backscattering Spectrometer, and the proposed high energy chemical spectrometer. Most significantly, the ARCS software does not serve the needs of physical chemistry. This subfield would be served by the DANSE project.
10.1. **Data Reduction**

Perhaps surprisingly, software to do a limited set of reductions of chopper spectrometer data to \( S(Q, E) \) has become available only in the past couple of years, and software to obtain \( S(Q, E) \) has been available only somewhat longer. The ARCS software project was first to provide these computer codes in an unbroken but configurable chain along Path 1 in Fig. 3. The new package, `reduction`, is a highly-modularized, third-generation data reduction code tested with data from the Pharos instrument. It has 70 main classes, of which only 3 are specific to Pharos and require changes for ARCS or SEQUOIA once their NeXus file format is established. Many of these classes are useful for other instruments as described in Section 5.1. The ARCS software project will deliver data reduction procedures for the ARCS instrument without funds from the DANSE project. The same package can be adapted to the other inelastic instruments of the SNS.

10.1.1. **Direct-Geometry Chopper Spectrometers**

Two other SNS inelastic chopper spectrometers have similar data structures as Pharos and ARCS. The DANSE project will deliver data reduction capabilities for CNCS and SEQUOIA.

10.1.2. **Other Inelastic Spectrometers**

Two other SNS inelastic spectrometers operate on different principles from the direct-geometry chopper spectrometers. Basic data reduction for these two instruments will require some changes to the class of `reduction`, with some elimination of steps in the case of the backscattering spectrometer. The DANSE project will deliver data reduction capabilities for HYSPEC and the high-resolution backscattering spectrometer. We will work with the NIST group of R. Dimeo to ensure that their `DAVE` software package is ported to the DANSE framework for support of the high-resolution backscatter spectrometer, and the components of DANSE will be made available for use within the `DAVE` package.

10.2. **Modeling of Dynamical Processes in Solids**

Models of dynamical excitations in solids are often used after the basic reduction of data to \( S(Q, E) \), and such models are an important connection between experimental results and the theory of solids. A Born–von Kármán model and Heisenberg spin dynamics model have already been adapted to DANSE during the ARCS project.

10.2.1. **Neutron Weight Correction**

Modeling can also play a central role in data reduction, although this has not been fully appreciated to date. For example, for a binary compound an inelastic energy spectrum \( S(E) \) measured on a polycrystalline sample cannot be converted into a true energy spectrum of vibrational modes because the different elements in the compound scatter neutrons with unequal efficiencies. Unfortunately, correcting for this “neutron-weighting” of the data requires detailed knowledge of the lattice dynamics of the compound, which is the end product of the data reduction. This “chicken and egg problem” can be overcome with a computational procedure where the force constants in the dynamics model are iterated to obtain the best fit to the experimental data [81]. Interatomic force constants are outputs from this procedure, as are the partial densities of states for the different atomic species in the compound. As this procedure becomes routine, better experiments can be designed around this new capability for data reduction [82]. We propose to develop new software to perform neutron-weight corrections in near real time, employing Gilat-Raubenheimer and Pack-Monkhorst algorithms. The former is a linear interpolation scheme used to expedite calculations of densities-of-states. The latter allows evaluation of integrals over the Brillouin Zone by consideration of a finite set of special \( k \)-points. We will also evaluate the nonlinear least-squares routines described in Section 5.4.

10.2.2. **Coherent Inelastic Scattering**

For detailed measurements of coherent inelastic scattering, software limitations for chopper spectrometers have allowed reactor-based instruments to play the lead role to date [83]. We propose a development activity to address the new science that is possible with better analysis of the \( Q \) information in the data from inelastic chopper spectrometers with pixellated detectors. The deliverables will be software for the analysis of \( Q \) information from experiments on polycrystals and single crystals using components that fit noisy dispersive data to analytical functions.
10.2.3. *Separation of Nuclear and Spin Scattering*
Another important use for models in data reduction includes the separation of spin dynamics from phonon dynamics by analysis of their $Q$-dependences. Test cases based on magnetic scattering or phonon scattering will be developed to use the increase of the phonon scattering with $Q$, and the decrease with $Q$ of the computed magnetic form factor to separate these two contributions.

10.2.4. *Chemical Spectroscopy Dynamics Codes*
For chemical spectroscopy of complex non-crystalline solids, several computational tools exist for the analysis of vibrational data. Some of these will be ported to the framework as described in Section 5.3.4. We will add an empirical refinement package based on a normal mode analysis of isolated molecules, CLIMAX [84], that takes into account the details of the neutron scattering cross-section and instrumental factors to match the measured and predicted spectra.

10.2.5. *Chemical Spectroscopy Preprocessing and Postprocessing*
With packages such as VMD it is possible to select a molecule in PDB format and use it as an input to several of the molecular dynamics codes of section 5.3.4. Existing software also allows for simulations with solvation of the molecule. Unfortunately, more general capabilities for setting up molecular dynamics simulations are not available from the open source community. It is still early to make definitive statements of community needs for the setup of molecular dynamics simulations, but some capabilities should be tested. Frans Trouw is pursuing computational neutron science research on the fluctuations of polymers and micelles, and has developed tools for input preprocessing that will be adapted to the pyre framework.

Postprocessing of molecular dynamics simulations requires new software and better user interfaces. Calculating a space-time correlation function $G(\vec{r}, t)$ requires collecting a time-series of snapshots of the configuration of the system, from which the time-dependences of different spatial correlations can be later computed. This is a task that requires substantial computing resources and large data sets, low inter-node latency, and high data bandwidths. It may be a good test case for grid-based computing, and we propose to study this possibility later in the DANSE project.

10.3. *Direct Experiment Simulation*
We have recently prototyped the third path from data to science shown on the left of Fig. 3. Our “full experiment simulation” used the McStas Monte Carlo package after it was integrated with fine granularity into the DANSE system. We ran a full simulation of an actual experiment by building a complete chopper spectrometer in McStas that sent individual neutrons to a virtual sample. This sample had the lattice dynamics and neutron scattering properties of nickel, calculated with a Born–von Kármán dynamics simulation. Time-of-flight spectra were then acquired at detector pixels. These simulated data were processed with reduction to produce output like those of experimental measurements. This was all done on the pyre framework running a single Python script that could be re-arranged easily to study the effects of instrument parameters on the resolution in $E$ and $Q$ that are seen in both calculation and real measurements, for example. The DANSE system gave a seamless framework for comparing simulated and real data. We expect simulations to prove even more valuable for calculating the dynamics of disordered solids. Quantum chemistry codes are now used for calculating vibrational modes in molecular solids with hundreds of atoms [85], and their use for calculating inelastic neutron scattering spectra has been demonstrated [86, 87]. We propose to integrate molecular dynamics simulations into the DANSE system and adapt them for computing neutron scattering data.

10.3.1. *Scattering Kernel*
For lattice dynamics simulations, the scattering is treated most conveniently as individual events in the first Born approximation. This is equivalent to sampling the velocity-velocity correlation functions in a dynamics simulations [78, 79]. We propose two approaches, both based on the relation between $S(\vec{Q}, E)$ and the Van Hove correlation function, $G(\vec{r}, t)$:

$$
S(\vec{Q}, E) = \int_{-\infty}^{\infty} G(\vec{r}, t) \ e^{i[\vec{Q} \cdot \vec{r} - (E/\hbar) t]} \ d\vec{r} \ dt
$$

(1)
First, the $G(\vec{r}, t)$ will be calculated as an average over a full the dynamics simulation. This has the advantage of cleanly separating the dynamics calculation and the instrument simulation. Second, the $G(\vec{r}, t)$ will be calculated for individual neutron scatterings in the sample. This has the advantage of including multiple scattering directly in the simulation.

10.3.2. Multiple Scattering

Multiple scattering is an important consideration for experimentalists. A traditional, conservative style for experiment design is to use a sample that is a 7% scatterer. Although thicker samples would give better counting statistics, they typically require corrections for the effects of multiple scattering. Multiple scattering is a special concern for quasielastic neutron scattering experiments because the signal is by definition close to the strong elastic scattering. Multiple scattering codes exist, and have the potential to improve the quality of experimental work, but they are often too difficult to use. Nevertheless, the capability of the multiple scattering code MSCAT-85, written by John Copley, is well-suited for integration into DANSE. It has clean internal programming, but requires re-compilation for most changes of experimental conditions. We will re-implement the algorithms and ideas contained in MSCAT-85 in C++ with a modern interface for input and output.

10.3.3. Disordered Spin Dynamics

To test some of the analysis methods, we propose an investigation into the spin dynamics of disordered alloys of rare earth elements, which have classical spin behaviors. In an alloy of Ho-Tb, for example, the magnetic moments at the different atoms vary with the atom on the site, as do the exchange couplings between moments. Obtaining a realistic distribution of exchange interactions by iteratively fitting a Monte-Carlo simulation to experimental data will require high-performance computing. We propose to develop computational support for an experiment that measures the thermal decay of magnetic excitations near the critical temperature in a disordered alloy.

10.4. Personnel and Other Resources

B. Fultz would continue to lead the inelastic subproject that began under the ARCS project. F. Trouw will be a close collaborator to oversee the developments required for physical chemistry.

11. Education, Outreach, and Knowledge Transfer

11.1. Textbook Content

Good documentation is a challenge for any software project. Under the ARCS project, the inelastic scattering effort has elevated this task to an ongoing scholarly activity. The scope of the inelastic software project follows to some extent the organizational structure of a book, *Experimental Inelastic Neutron Scattering*, under development by the Caltech group and associates [88]. This manuscript includes textbook content to explain the theory of inelastic neutron scattering and excitations in condensed matter, and is being used to educate graduate students at Caltech. Textbook content also serves as reference material for the software algorithms and code, ensuring that the code follows the notation of the underlying theory. If the DANSE construction project is funded, this book project will expand to cover more of the scope of neutron science.\(^{11}\) Simon Billinge has agreed to participate in the book project, adding modern content on neutron diffraction methods. Ersan Üstündag has agreed to contribute content on engineering diffraction and solid mechanics. With extensive cross-referencing between software documentation, the underlying theories, and source code, this document will serve as the online help system for the DANSE system. The level of presentation would be similar to that of our previous texts [1, 2]. The five years for the DANSE construction project should offer enough time for a scholarly focus on textbook content. No funds are requested for this task, although we consider it an important one.

\(^{11}\) This book will be published electronically through the Caltech Open Digital Archives. Books published this way are fully open to public access and distribution, and will be maintained by the Caltech library (i.e., the library will be responsible for data format migration from pdf into the future formats).
11.2. K-12 Education

Iowa State University, with its history of collaborative projects between its College of Engineering and its College of Education, will be the lead institution for the K-12 educational effort of the DANSE project. The ISU College of Education is home for the Center for Technology in Learning and Teaching [89]. The CTLT has developed a technology-rich teacher education program, which will help channel the outreach activity of the DANSE project. Our collaborator is Dr. Larry Genalo, who developed and runs the “Toying With Technology” (TWT) Program and the Internet Explorers Program at ISU. These projects have helped ISU develop strong partnerships with K-12 schools, including schools in rural and urban settings. The history of collaboration with these schools will create a firm foundation for our K-12 outreach on crystal structure, dynamics and diffraction.

11.2.1. K-12 Lesson Plan Development

The subproject teams will serve as science content providers in a collaboration with the educational development team at ISU to develop lessons for use in science methods classes. Engineering diffraction will be the first group for these interactions. In the first year of the DANSE project, a two week unit lesson plan relating to diffraction and materials will be developed. Members of the ISU educational development team will attend the DANSE developers’ meetings to discuss this first program, and identify K-12 educational opportunities within the breadth of science of the DANSE project. Similar lesson plans will be developed in subsequent years with content provided by each subproject in DANSE. These lessons will become part of the Toying With Technology (TWT) course program.

Simon Billinge is working with teachers and students at Everett High School, an inner-city high school in Lansing, MI, to develop the curriculum for an advanced placement course in Nanotechnology under an NSF-funded NIRT grant. In association with the ISU educational development team, they will expand these activities to develop and test course materials that include some nanoscience addressed by the DANSE project. An application will be built using DANSE molecular simulation and visualization components to let students in the AP Nanotech course visualize, rotate and manipulate nanoparticles, and see the effects of temperature on atomic motion, for example. Capabilities such as scattering calculations will be added as deemed appropriate from the feedback from these high-school students.

11.2.2. Preservice Teacher Education

Preservice teacher education (the education of teachers entering the profession) is integrated at ISU in the Toying With Technology program and the associated course, Mat E 370, taken by preservice teachers enrolled in teacher education science methodology courses. Outreach to inservice K-12 teachers (practicing classroom teachers) and their students is also part of the ISU program.

11.2.3. Teacher Workshops

We propose that the DANSE project support a summer program at ISU for K-12 teachers. The educational lessons developed in section 11.2.1 will be presented and discussed in these summer programs, which will begin in the second year of the DANSE project.

11.2.4. K-12 Education Outreach

The preservice teachers will visit K-12 classrooms and teach the educational lessons. Assessment of these lessons is an important part of the K-12 curriculum development. This assessment will be performed by an ISU graduate student in education with his or her thesis adviser. As appropriate, dissemination of teaching materials will be through the Toying With Technology web site [90], and through ongoing collaborations with partner institutions such as George Mason University. Papers will be written and presented at the ASEE Annual Conference and the Society for Information Technology in the Teacher Education (SITE) Annual Conference.

11.3. Undergraduate Education

We propose that all institutions receive a budget for the summer support of one undergraduate student from an underrepresented minority group. These funds are independent of the support for undergraduate students involved in the construction of the DANSE software, although it is hoped that in some cases these students will be the same person(s). In the design phase of the DANSE project, for example, the Caltech Office of Minority Student Education helped us identify and hire
a talented minority student through the work-study program. DANSE is now offering Minority Undergraduate Research Fellowships for the summer of 2005.

11.4. Graduate Education

The DANSE project offers unique research opportunities to postdoctoral fellows and graduate students. From our experience with the ARCS software project, we have found that most postdocs and graduate students remain interested in the science that underlies the neutron scattering research. It is therefore important that their work include new science. This will be true for the DANSE project because the subprojects have tasks that require the acquisition of data for original research. It would be much harder, of course, to enthusiastically develop software to reduce experimental data if the developers had no personal stake in the outcome, or were not sensitive to the nuances of extracting scientific results from experimental data.

One of the most important skills learned by members of the ARCS team has been how to work across the interface between computer science and physical science. Simon Billinge is exploring an academic program at this interface. Recognizing that the next generation of physical scientists will require much more sophistication with software and computer architectures, the Physics and Computer Science departments at Michigan State University have begun discussions about establishing a joint graduate program on high-performance scientific computing. This program will likely require curriculum development [91]. Simulations and algorithmic analyses are not emphasized in today’s physics curriculum. Besides the obvious need for exposing physical science students to concepts of computer science, and training in modern programming practice, it is also appropriate to teach how to approach physical problems using distributed computing systems. These issues extend beyond physical science.

11.5. Meetings, Workshops, and Participant Support

The components of the DANSE system will be built primarily by scientists with experience in computer programming, but with limited experience as members of a professional software development team. By design, the component architecture itself forces adherence to good programming practice of code encapsulation with well-designed interfaces. Especially in the early stages of the project, the subproject teams will need help with component design, graphics support, and the user interface. Education workshops are important activities for the personnel in the DANSE project, especially for postdoctoral fellows who are new to the project. DANSE workshop agendas will be arranged so that members of the educational team from ISU can profitably attend. These will be open to the broader community of scientists interested in DANSE, including our international colleagues.

11.6. Collaboration with Other Software Efforts

The DANSE developers are well-acquainted with the other U.S. projects to develop neutron scattering software, especially the ISAW project at Argonne National Lab. and the Univ. of Wisconsin, and the DAVE project at the NIST Center for Neutron Research. It is a priority of the present DANSE CED project to ensure that ISAW and DAVE can run under the DANSE framework, and we have demonstrated key features required for this compatibility, such as Python bindings for IDL (for DAVE), and communication with the Java interpreter through Python programs. Other approaches will also be tested over the next few months. Both the DAVE and ISAW projects are more mature than DANSE, with active users and regular updates. From ISAW, which is an open source project distributed under the GPL license, the visualization components will be used under the pyre framework for support of the ARCS instrument at the SNS. Both the ISAW and DAVE projects have full access to DANSE software components, since DANSE is an open source project. Perhaps the most important topic for regular communication between these projects are the data object models, in which both ISAW and DAVE have substantial investments.

Feedback from meetings held at Caltech in Sept. 2003 and June 2004 showed that the most well-received presentations were those on programming practice and culture, and the technology of components and data structures.
11.7. International Outreach

International collaborators include several persons who regularly attend the workshops at Caltech. These individuals affiliated with neutrons sources in the U.K., Germany, Canada, Australia and Japan would be pleased to work with us on software compatible with the DANSE framework. They expressed a keen interest in DANSE even before funding was obtained for the design effort. We are confident that it will be easier to engage persons at other neutron sources if the DANSE construction project were funded.

The Neutron Science Software Initiative (NeSSI) is an informal collaboration between major neutron scattering facilities throughout the world (SNS, ISIS, J-PARC, IPNS, NCNR, and HFIR) to develop a common, comprehensive software infrastructure for data acquisition, analysis, and management. The partners recognize the efficiency to be gained in sharing critical resources and software in a combined development project and the collaboration is being formalized through Memoranda of Understanding allowing them to do this. The very nature of the international NeSSI collaboration provides for rapid deployment of the infrastructure and allows the major components to be developed simultaneously at different institutions. SNS is developing the Web based User Access portal, while ISIS has concentrated on the data management and cataloguing systems. J-PARC is concentrating on the analysis component libraries. The DANSE framework will fit naturally into the NeSSI infrastructure and SNS is working closely with the DANSE team to develop the interfaces.

12. Project Plan

12.1. Operational Goals

The operational goal of the DANSE project would be to build a software system for neutron scattering research that:

− integrates the basic data reduction capabilities that are available today,
− enables new types of science in all major subfields of neutron scattering research,
− provides a coherent framework onto which software components can be added by scientists,
− is fully maintained by the SNS software group before the end of the project.

12.2. Planning Process

In October and November of 2003 we prepared a first draft of a five-year project plan with a work breakdown structure to level 4 in detail. The project plan included 259 tasks, with subtasks such as designing, building, testing and revising. The total estimated cost was M$ 16.7. Although this large exercise identified most of the tasks on the present project plan, it nevertheless had a number of problems with the estimation of effort and risk, and had some duplication of effort.

A new and independent effort to develop a project plan with a WBS to level 4 in detail began in August, 2004. Triad Project Management of Pasadena, CA, was contracted to work with the P.I. on developing a WBS dictionary with tasks defined to level 5 in detail, and sometimes to level 6. This was performed with Excel spreadsheet templates for collecting task data appropriate for a WBS dictionary. A risk assessment was made for each task. Risk mitigations were suggested, often leading to revised tasks of lower risk. The biggest challenge in this effort was working with the subproject leaders to develop practical guidelines for assessing the effort and risk of level 5 tasks.

For many tasks, the required effort was assessed by first estimating the time to complete it by a trained and competent individual with a clear idea of the final structure of the code. Most tasks are more complicated than this, so a multiplier for risk of delay was developed as described in Section 13.2.2. (In most cases, however, the damage factor was not used as in Eq. 2.) Multiple iterations between the subprojects and the Pasadena group were used to make these estimates. Throughout this entire process, the emphasis was on obtaining accurate estimates of time and resources. Reliability checks were performed by using the same methods on tasks already performed for the ARCS software effort, for the present DANSE CED design effort, and for in-house software development efforts at the different institutions. The reliability of the estimates and the minimization of risk favored the use of existing software rather than developing new code, and tended to favor the assignment of professional
Table I. Proposed Milestones at Level 2

<table>
<thead>
<tr>
<th>Milestone</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prototype components</td>
<td>August, 2004</td>
</tr>
<tr>
<td>Baseline design review</td>
<td>May, 2005</td>
</tr>
<tr>
<td>α-build</td>
<td>Nov., 2006</td>
</tr>
<tr>
<td>Critical design review</td>
<td>Nov., 2007</td>
</tr>
<tr>
<td>β-build</td>
<td>May, 2008</td>
</tr>
<tr>
<td>Release review</td>
<td>Jan., 2009</td>
</tr>
<tr>
<td>Software Release 1.0</td>
<td>May, 2009</td>
</tr>
<tr>
<td>Software Release 1.1</td>
<td>May, 2010</td>
</tr>
</tbody>
</table>

Programmers to many of the tasks. To some degree, the present project plan reflects these pressures. Our best estimates were used to develop a project plan and budget with no contingency, as required by the IMR-MIP program.

Many software tasks can be performed independently. The main challenge in sequencing the tasks was to match a realistic budget profile with continuous support for personnel. Tasks for the project infrastructure, component framework, graphics, and user interface components were scheduled earlier, since these will facilitate much work for the science subprojects. The sequencing of instrument commissioning at the SNS was also used to schedule the different tasks, giving earlier emphasis to the reflectometry and inelastic scattering subprojects, and later schedules for the engineering diffraction and SANS subprojects. The five year DANSE budget was obtained by summing the costs of the 186 tasks at level 4 of the WBS for each year of the project. The Excel spreadsheet to do this was uploaded into Fastlane as part of this IMR-MIP proposal submission.

12.3. SUMMARY OF PROJECT PLAN

12.3.1. WBS Category 1 – Milestones and External Drivers
In the present plan with milestones listed in Table I, the years 2004 and 2005 include design work, prototyping, staffing, and some construction work. The years 2006 and 2007 would include the first builds of the DANSE system. The project ends with a release 1.1, which we consider essential for testing and refining the release management. If talented personnel are hired between 2005 and 2006, the beta release could occur earlier in 2008 than listed in Table I, with a release 1.0 in 2008.

12.3.2. WBS Category 2 – Project Integration
Essential to the success of the DANSE project is a full-time Project Manager, who would be hired in the first year (see Section 13.2). An administrator, or a fraction of an administrator plus office staff member is also necessary to help coordinate the project. Communications, workshops, and meetings would be an essential part of the project, and a budget to support these activities is requested.

12.3.3. WBS Category 3 – Infrastructure and Support
Support for the software developers of the DANSE project would be provided by hardware, software, and personnel. Two hardware purchases are needed over the course of the DANSE project to provide a testbed for developers. Simplifying the build procedure for the pyre framework is an early priority for software support. Many ongoing services for project infrastructure involve helping developers with version control, testing, and quality assurance.

12.3.4. WBS Category 4 – Software System Engineering
The software system engineering effort is a continuation and extension of the ongoing development of the pyre framework at Caltech. Many of the capabilities of the framework are available today, giving a development platform for the components of the DANSE system. This allows an early emphasis on tools for graphics and user interfaces that are needed for the applications developed in the science subprojects. The development of the distributed computing capabilities of DANSE continue through the full project, with more sophisticated stream handling capabilities and Grid
compatibility scheduled for somewhat later in the project, as are advanced graphics and tasks for reliability and optimization.

12.3.5. **WBS Category 5 – Common Features of Scientific Subprojects**

Work performed under the DANSE CED proposal gave the opportunity to identify software components that were common to two or more of the science subprojects. Discussion sessions were devoted to common algorithms at two of the DANSE developers' meetings in 2004, and common tasks were discussed as needed in the DANSE weekly conference calls. A number of common tasks were also identified by the P.I. when level 5 detail was provided on the tasks from the scientific subprojects. Collecting these common tasks into a separate subproject minimizes the duplication of effort. Although coordinated by the Project Manager, each individual task will be owned by the group that is best qualified and most motivated to work on it. This consolidation also helps ensure the uniformity of the DANSE system. These common tasks tend to be scheduled earlier in the DANSE project. For example, the optimization routines and numerical libraries are needed before building other types of data analysis components. Many of the common tasks involve porting software packages for materials simulations to the DANSE framework, and some packages of lower priority and similar function are scheduled later.

12.3.6. **WBS Category 6 – Diffraction**

The diffraction community has been engaged in the development of subproject activities through online surveys of software usage (see [http://danse.cacr.caltech.edu/polls/results.php?sid=22](http://danse.cacr.caltech.edu/polls/results.php?sid=22) for the results from 144 respondents), and through and requests for feedback from diffraction community listservers. Existing software packages were inventoried using the “Collaborative Computational Project Number 14” crystallographic software repository at [http://www.ccp14.ac.uk/](http://www.ccp14.ac.uk/) and the SINCRIIS software database ([http://www.iucr.org/sincris-top/logiciel/index.html](http://www.iucr.org/sincris-top/logiciel/index.html)). The online survey and the inventory of existing software gave a fairly complete picture of available software, ongoing software development projects, software usage, and needs for future capabilities.

The main objective of the diffraction sub-project is a complete ground-up rebuild of regression modeling programs. This will be a lengthy process, so initial projects will be to wrap existing PDFFIT and Rietveld codes in the DANSE framework. Data reduction codes will first be developed for existing diffractometers and tested with the goal of real-time automated data-reduction. Other important early activities will be the creation of libraries of crystallography components and detailed design of the rebuilt regression modeling programs with careful diagramming and pseudo-code generation.

12.3.7. **WBS Category 7 – Engineering Diffraction**

The engineering diffraction community has discussed software at several workshops since 1997. At least three meetings of the Instrument Development Team (IDT) of the SMARTS engineering diffraction instrument at Los Alamos had a focus on software. Three meetings of the IDT for Vulcan, the engineering diffraction instrument under construction at the SNS, had substantial discussions of software status and development. The recent Nov. 2004 meeting of the Vulcan IDT had a substantial emphasis on how DANSE could provide value to engineering diffraction research. The tasks in this DANSE subproject are in direct correspondence to these requests from the members of the Vulcan IDT and the engineering diffraction community at large.

The commissioning of Vulcan at the SNS is scheduled in 2008. Compared to other instruments at the SNS, this gives some more time for the development of software for engineering diffraction. The peak in the funding profile of the engineering diffraction subproject is in years 3 and 4 of the DANSE project. Most of the early tasks in the subproject are specific deliverables for data reduction and adaptation of software tools for the main testbed instruments in engineering diffraction. On the other hand, tasks on 3D finite element, inverse problem and microstructural analyses are more research-oriented and scheduled for the later years of the DANSE project.

12.3.8. **WBS Category 8 – SANS**

An online poll of software for the SANS community [http://danse.cacr.caltech.edu/polls/results.php?sid=21](http://danse.cacr.caltech.edu/polls/results.php?sid=21) has been completed by 53 respondents so far. There was a focus of the poll on ease of use, and this was a clear priority for respondents. Most respondents were satisfied with the level of science that they can do with today’s software, but most believed that more sophisticated software would enable more information to be extracted from SANS experiments.
The SANS instrument at the SNS is scheduled for commissioning in 2007, somewhat later than
the reflectometers and inelastic spectrometers. The efficient development of user interfaces requires
tools developed in Sections 4.2 and 4.3 of this proposal, so the peak in the funding profile of the
SANS subproject is in years 3 and 4 of the DANSE project.

12.3.9. WBS Category 9 – Reflectometry
In August of 2004, we conducted an email survey of reflectometry experts from around the world,
showing them proposed tasks for the reflectometry subproject. The results from 27 diverse respon-
dents confirmed that the goals of the subproject will meet the needs of the community.

The first tasks are to make the core functionality of data reduction available within the DANSE
framework. Viewing and reduction of SNS instrument data is the highest priority. Next, the basic
analysis components will be made available. Since these can be rearranged as needed to refine
arbitrarily complex sets of data, this functionality will already go beyond existing reflectometry
software. Direct inversion will be made available to the broader community within the reduction
application. The reflectometry team will then adapt code for translating atomistic models into depth
profiles, allowing users to perform refinements directly on 3-D models. After modeling tools are in
place, work will proceed on diffuse scattering analysis and full experiment simulations.

12.3.10. WBS Category 10 – Inelastic
Workshops and user polls helped set the directions for the ARCS project to develop software for
inelastic scattering research. The ARCS schedule met its milestone for a prototype build in the
summer of 2003. A second build in the summer of 2004 allowed for the full capabilities of data
reduction to $S(Q, E)$, and the interface to semi-phenomenological models of Sections 10.1 and 10.2.
The beta release of the software is scheduled for March of 2005. The ARCS software project has
proved invaluable for estimating the effort required by tasks in the proposed DANSE project.

The first tasks of the inelastic scattering effort in the DANSE project will be adapting the ARCS
data reduction software for other SNS inelastic spectrometers. The next tasks in the schedule involve
porting a set of dynamics simulation codes to the pyre framework. Integrated molecular, phonon,
and spin dynamics simulations are new capabilities for computational neutron science, and it is
important to have them available soon because some experimentation with them is needed.

12.3.11. WBS Category 11 – Education and Outreach
The development of textbook content for inelastic scattering, diffraction, and engineering diffraction
will be undertaken by the investigators, but are not funded as tasks under DANSE. The Iowa
State program for K-12 education, Toying With Technology, will be supported from the start of
the DANSE project. The student-teacher workshops will begin in the second year. Funding for the
“hands-on” nanotechnology high school curriculum is available for the last three years of the DANSE
project, but funds from other programs at Michigan State Univ. are available for this program in
the first two years of the project. Each subproject has an annual budget for the support of minority
undergraduate students. Graduate students are involved in many of the science subprojects, and are
supported through specific tasks of the DANSE project, as described in Section 11. Postdoctoral
fellows would be involved heavily in the DANSE project.

13. Project Management

13.1. Management Organization

13.1.1. Roles of P.I. and Co-P.I.s
The Principal Investigator of the DANSE Project, Brent Fultz, is accountable for the success of
the present Conceptual and Engineering Design (CED) project and would be accountable for the
execution of a future DANSE Construction (CNST) project in the NSF IMR-MIP program. Michael
A.G. Aivazis, a Co-Principal Investigator of the DANSE project, has worked with Brent Fultz since
2001 on the software subproject for the ARCS spectrometer project. The Co-Principal Investigator
Ian Anderson of the SNS has been working with both Fultz and Aivazis to coordinate effort with the
SNS and with international groups, especially personnel at the ISIS spallation source in the U.K.
Responsibilities of the P.I. in a working relationship with the Co-PIs include:
− Measuring project performance against established goals, including technical performance, cost levels, and schedule milestones. This includes monitoring monthly project costs, and providing this information to the NSF as requested.
− Organizing meetings to monitor and coordinate the hardware and software efforts. These meetings will serve to report progress and discuss issues.
− Other workshops for the education of the DANSE personnel and the broader neutron community will be scheduled as necessary.
− Approving and disbursing project funds.

If the construction project were approved, funds would be sent from the NSF to Caltech as a cooperative agreement award, with Brent Fultz as Principal Investigator. Funds to the subprojects would be disbursed as subcontracts between Caltech and the academic institutions of the subproject leaders.

13.1.2. **Project Manager**
If this proposal is selected for a Reverse Site Visit, Caltech will begin a search for a Project Manager for the DANSE Construction Project. The most important qualification for the DANSE Project Manager will be project controls experience. Nearly as important is experience with software development for high performance computing or networking. We hope to identify a Project Manager who is interested in a position intermediate between industry and academia, perhaps with a personal goal of testing the path back towards academia.

13.1.3. **Steering Committee**
Institutional oversight of the DANSE project would be provided by a Steering Committee, that would include representatives from the other neutron scattering facilities in the U.S. and experts in high-performance computing. Members of the Steering Committee would: 1) attend the Developers’ Meetings to assess progress and provide advice, especially advice from the larger perspective of the U.S. neutron scattering community, 2) address issues of institutional coordination among the DANSE project, subproject institutions, and the SNS, especially issues that affect the value of the DANSE project to the neutron facilities and the user community, 3) provide advice and consent to the P.I. and Co-I. if the need arises to reorganize a subproject, and 4) offer information and advice to the NSF if requested.

13.1.4. **Developers’ Committee**
The Developers’ Committee will include the P.I., Co-Is, and the subproject leaders. This Committee will meet three times annually for status meetings, followed by two-day software development workshops that will be open to the neutron community. It is hoped that personnel from the NSF or their designees would also attend these meetings. The Developers’ Committee Workshops will be the main public forum for the DANSE project. It is expected that graduate students, postdoctoral fellows and junior scientists will attend the workshop and present their work. Other meetings will be scheduled as needed to sustain momentum in the project.

13.1.5. **System Development Team**
The DANSE System Development Team (SDT) shall be composed of individuals who contribute to the DANSE proposal, and those interested in the development of DANSE. The SNS has selected the following instrument scientists for inclusion in this group: Jason Hodges (diffraction), Xun-Li Wang (engineering diffraction), John Ankner (reflectometry), Dean Myles (SANS), and Garrett Granroth and Ken Herwig (inelastic). Membership in the DANSE SDT is intended to be open to all interested parties, however, and the following individuals have become regular and active participants in this group: Ray Osborn (Argonne), Przemek Klosowski (NIST), Rob Dimeo (NIST), Dennis Mikkelsen (U. Wisc.), Tom Worlton (Argonne). Many others have expressed interest in participating in the workshops. Members of the SDT will be encouraged to attend the tri-annual Developers’ Workshops. Further into the project, members of the DANSE SDT will work with the software developers to test the releases of code, and comment on its strengths and weaknesses for their own scientific needs.

---

13 The present DANSE Design Project has no Project Manager. Brent Fultz is handling these responsibilities, but this arrangement is not appropriate for a large Construction Project.
13.2. Project Controls

13.2.1. Earned Value Management System

Ian Anderson will oversee the earned value management system for the DANSE project. It will be the same system used for instrument construction projects at the SNS.\(^{14}\) The SNS earned value system uses state-of-the-art project management software system to analyze and report technical, cost, and schedule progress on a monthly basis. It is EVMS certified as compliant with ANSI/EIA-748. This valuable service will be provided without cost to the DANSE project. In turn, the SNS will be better able to plan the transition to ownership of the DANSE system, and offer advice on the scheduling of the project tasks.

The collection of task information was described in Section 12.2. Here we reiterate that for nearly all level 4 tasks we collected task descriptions, justifications, estimates of resources for subtasks at level 5 in detail, and risk with mitigation plans. The information from these Excel spreadsheets will be used by the SNS project controls group for building a WBS dictionary if a construction project is funded. The present project plan is in Microsoft Excel and Project, but we do not intend to maintain these files once better services are available through the SNS.

Actual effort will be recorded by all individuals on a weekly basis. These data shall be used to assess the expenses accrued by each task. Level 5 subtasks or milestones have been developed for nearly all development tasks in the WBS. Completion of these level 5 tasks shall be monitored by the subproject leaders with oversight by the Project Manager, and shall be used to provide monthly data for the earned value management system. Actual monthly cost and schedule performance data will be transmitted to the SNS Project Controls group. Approximately at mid-month, an earned value report will be generated, showing the baseline cost of work scheduled, the baseline cost of work performed, and the actual cost of work performed. Procedures to generate these reports have been refined over several years, and a similar system works well for the Caltech-SNS collaboration on the ARCS spectrometer project.

The project controls system includes provisions for risk monitoring and documentation. “Risk items” will be identified each month in a category that can be quantified. “Gut feel” items will be identified each month as risks that cannot be expressed in money or time. These items will be compiled by the Project Manager into a Risk Watch List, and will be discussed as needed in the DANSE weekly conference call. Minutes of these meetings will be prepared by the Project Manager, and issued as controlled documents of the DANSE project.

13.2.2. Management of Risk and Configuration

Risk Assessment. Identifying risks is the first step in risk management. The biggest risk in software development projects is usually a misperception of specifications [92, 93, 94, 95]. The DANSE project has been organized to minimize this risk. The developers are themselves users of neutron facilities, and the SNS is well-represented in the project management. The present proposal, with its ancillary materials such as its WBS dictionary and lower-level task descriptions (sometimes to level 6 in the WBS), constitute the specification of what will be built in the DANSE project. It has been compared to two drafts of the SNS document on Functional Requirements for Data Analysis Software, by both the DANSE project team and SNS personnel.

For every task at level 4 of the present WBS, and in many tasks at level 5, risks were identified, assessed, and included in the WBS dictionary. For the assessment of risk, each task or subtask was assigned a probability of delays that increased with the innovation and work required for the task.\(^{15}\) A second factor was used to assess the damage to the DANSE project that would be caused by the loss of this task.\(^{16}\) The Risk is the product of these two factors:

\[^{14}\text{This includes the ARCS instrument, so Caltech has a working relationship with the SNS Project Controls group.}\]

\[^{15}\text{Multipliers were: } \times 1\text{ for existing code with no modification required; } \times 2\text{ for minor modifications to existing code; } \times 4\text{ for extensive modifications to existing code; } \times 5\text{ for new code using established algorithms; } \times 6\text{ for a new algorithm using established theory; } \times 9\text{ for a new algorithm requiring adaptation of an established theory; } \times 12\text{ for a new algorithm with new theory that advances the present state of the art; } \times 15\text{ for a new algorithm, way beyond the current theory.}\]

\[^{16}\text{The increments were: } 5\%\text{ if alternative exists, workaround is obvious; } 20\%\text{ for acceptable loss of functionality; } 30\%\text{ for a major loss of functionality; } 50\%\text{ for a loss of a core functionality for subproject; } 70\%\text{ if fatal to subproject; } 100\%\text{ if fatal to DANSE project.}\]
Risk \equiv \text{Probability} \times \text{Damage} \quad (2)

This assessment of risk for the WBS dictionary was performed with thoroughness because it was used in a first approach to estimate the effort requirements for the different tasks. In this first approach, the time was estimated based on the hours expected to complete a task if there were no difficulties. Additional time was allocated to each task at level 4 of the WBS in proportion to its Risk (both with and without the damage factor of Eq. 2). This additional time was not intended to be contingency, but rather a way to obtain a more accurate estimate of the time required for a software task. As noted in Section 12.2, these estimates were cross-checked by comparison to completed tasks, and some subproject leaders were also able to use their own experience in estimating effort. In the end, these estimates of effort were not applied to all subprojects or all tasks within a subproject, but they proved useful in identifying risky tasks, and often motivated other approaches to obtaining the needed functionality.

**Risk Reporting.** If DANSE is funded for construction, a first version of a Risk Watch List will be prepared from the information in the WBS dictionary. Maintaining and updating this risk watch list is of course an ongoing part of a construction project. DANSE has had a weekly conference call of the investigators and sub-project leaders since 1 Oct. 2004. These regular calls are the quickest forum for a subproject leader to report newly-identified risks, and report changes in the risks of the project tasks. The tri-annual DANSE Developers’ Meetings will include an assessment of risks.

**Risk Mitigation and Configuration Management.** Risk mitigation plans were listed for the level 4 tasks in the WBS dictionary, and for some level 5 tasks. Some of these involve rearranging the project schedule, or changing a dependency on a supporting software package, for example. Coordinating these changes will be the responsibility of the Risk Control Board. The Risk Control Board will be composed of the same individuals as the Change Control Board, ensuring informed decisions on risk mitigation and how these affect the configuration of DANSE. It will include the Project Manager, the Co-Principal Investigators, with Brent Fultz, P.I., as leader. Responsibilities of the Risk Control Board are:

- Document the risks through the risk watch list, with updates at least after each Developers’ Meeting.
- Maintain risk mitigation plans and contingency plans.
- Plan and request changes to scope or configuration after risks undergo large changes.

Configuration management is closely related to risk management, and the Configuration Control Board comprises the same individuals as the Risk Control Board and the Change Control Board. Michael Aivazis will lead the Configuration Control Board because many of the configuration issues involve software component dependencies and how software components interact. Other configuration management responsibilities include:

- Operation of the CVS source code repository.
- Approving code for release, which depends on good configuration management.
- Operation of the release repository.
- Oversight of testing practices, including unit testing, regression testing, and usage testing.

13.2.3. **Project Baseline Maintenance**

The project baseline and project plan will be updated annually with the help of the SNS Project Controls group. The Project Manager will coordinate this activity with assistance from the sub-project leaders and the investigators. In addition to a earned value assessments and an estimate of project scope at completion, a projection will be made of the status of the DANSE system at the milestone dates of Table I. Such information would help to plan the transition to the SNS operations, and may suggest changes to the project scope.

13.2.4. **Change Control**

A system for change control would be presented in a Project Execution Plan for DANSE if construction were funded, but some procedures are suggested here. Major changes to technical characteristics
(e.g., creating or annihilating a subproject), schedule (e.g., 6 month delays of major milestones), and
cost (e.g., any change in total cost), would require reporting to the NSF MPS Division. Thresholds
would be lower for triggering action of the DANSE Change Control Board on technical characteristics
(e.g., switching to a different software package at level 4), schedule (e.g., a 6 month delay in a level
4 task), or cost (e.g., a deviation of k$ 75 for a level 3 activity). The Change Control Board would
be small and engaged so it can evaluate efficiently potential changes to the scope, cost, and schedule
of the DANSE project. It would include the Principal Investigator, the Co-Principal Investigators,
and will be chaired by the Project Manager.

13.2.5. Contingency Management
Cost contingency is not allowed under the NSF IMR-MIP program. The contingency would therefore
be the scope of the project. Tasks should be descoped or rescoped annually to ensure completion of
the tasks of highest priority. The P.I. would control the contingency.

13.2.6. Reviews and Reports
A Reverse Site Visit serves as the Baseline Design Review, and acceptance of a construction proposal
constitutes the CD-2 and CD-3 milestones of the construction phase. If the Construction proposal
were funded, discussions would begin on the frequency of reporting of earned value and project status
to the NSF. It may be most convenient to compile the earned value reports on a tri-annual basis
in prior to the Developers’ Meetings, and issue a written report to the NSF after these meetings. A
plan for annual reviews and a review at 60% completion would also be developed after funding is
approved.

13.3. Transition to SNS Operations
13.3.1. Quality Assurance
The DANSE system would be used by all neutron scattering instruments operated by the SNS. As
such, the engineering, building, and installation of the DANSE system must follow quality assurance
procedures developed by the SNS Experimental Facilities Division (XFD). The Memorandum of
Agreement between Caltech, the SNS, and the NSF should state that the DANSE project would
follow the quality assurance practices of the XFD, as is the case for neutron instrument hardware.
This oversight function will further ensure that the SNS software group can smoothly accommodate
the transition of the DANSE system to the SNS through the course of the project.

13.3.2. Roles and Responsibilities
The SNS has the responsibility for supporting the future users of the DANSE system. To support
users on a system of high reliability, the SNS needs to have full control over a working version of the
data analysis system as soon as practical. It is expected that some of the software components on this
system will be validated by the SNS, some will be open source codes that are not validated, some will
be commercial packages, and others will be private for the developers or scientists who wrote them.
Policies, licenses, and authorization for access to these different codes are the responsibility of the
SNS, and have been listed in the document “Functional Requirements for Data Analysis Software
at the SNS” (SNS-IS-107020000-TD0001-R00). The DANSE software technology is consistent with
these policy needs.

The evolution of the DANSE system will be from an unstable development system into a system of
high reliability, culminating in one major upgrade before the end of the project. During the DANSE
project, the software and hardware infrastructure will be centered at Caltech, and development
efforts will be distributed nationally. At the end of the project, a high reliability system must be
available at the SNS to provide user support during SNS operations. The transition plan must be
steady and gradual, not an abrupt transfer of a final release to the SNS. Ian Anderson, Director
of the Experimental Facilities Division (XFD) of the SNS and a Co-Principal Investigator on the
DANSE CED project and this CNST proposal, will coordinate the transition plan. Members of the
SNS software group and some SNS instrument scientists have attended all the DANSE meetings
held to date. As the project matures, the DANSE workshops will focus increasingly on the needs of
the SNS. It is likely that some of the postdoctoral fellows in the DANSE project will be later hired
by the XFD. It is important for the evolving SNS software group to become as familiar as possible with all technical aspects of the DANSE system.\textsuperscript{17}

Some responsibilities for software development can be cleanly separated between the DANSE team and the SNS software group. The first focus of the SNS group will be stewardship of the data produced by the facility. This responsibility includes ensuring data integrity, archiving, proper support of NeXus standards [28], and making access to data both convenient and secure. A set of policies need to be decided by SNS personnel, including policies for data access, standards for metadata, and licensing arrangements for proprietary software. There will be hardware requirements for these functions. The SNS software group can make these decisions largely independently of the DANSE developers. At the other end of the spectrum, simulations and models can be developed by the DANSE team with input primarily from the neutron user community rather than from the SNS software group.

There are some aspects of data analysis where overlap of responsibilities occur, and where close cooperation is needed between the SNS and the DANSE team. For example:

- Security procedures must be consistent with SNS policy.
- Software components for basic data reduction and display must be validated by SNS personnel because they reflect on the quality of experimental results produced by the facility. We plan to work closely with the SNS instrument scientists to complete the tasks of performing data reduction for their instruments. This ensures that the SNS staff are familiar with the code of reduction, for example, and are able to maintain it.
- The user interfaces reflect on the user opinions of the SNS facility.
- The SNS will negotiate licenses for proprietary software, and maintain these licenses.

13.3.3. Planning the Transition to SNS Operations

Planning for the transition of the DANSE system to the stewardship of SNS Operations has been underway for some time. Some issues are: 1) the roles and responsibilities of the SNS software group and the DANSE team. These roles must evolve through the design and construction phases to match a phased transition of responsibilities to an SNS software group. 2) The assignment of priorities to the tasks undertaken by these two groups. 3) The role of DANSE within the developing international cooperation on neutron scattering data. 4) The future role of the DANSE development community after the DANSE system is maintained by the SNS.

A software engineer, to be supported by the DANSE project under a subcontract with the Software Quality Research Laboratory (SQRL) at the Univ. Tennessee, Knoxville (see section 3.2.4), would have a regular presence at the SNS, and would be able to brief the DANSE project on progress and issues of data access, data transport, and software integration as described in section 13.3.2. As part of the certification effort described in Section 3.2.4, this engineer, with support of other SQRL staff, would work with the DANSE project and the SNS to develop software acceptance criteria, a plan for software releases, and help ensure that the DANSE development effort follows the quality assurance practices of the SNS.

The intellectual property officers at Caltech, the University of Maryland, and the University of Tennessee have agreed that these DANSE subprojects would be open source projects that will distribute software under the nonrestrictive BSD license. This is our preference, and we will ensure that the DANSE framework will be distributed with a BSD license. The subprojects at Michigan State and Iowa State have not completed licensing discussions, but to date their intellectual property officers have agreed to two paragraphs of legal language that present the following concept. The federal government, including the SNS, is granted extensive rights to use, modify and distribute software developed with federal funds. MSU and ISU have agreed to extend these rights to academic and research institutions for non-commercial purposes. This includes foreign institutions, subject to applicable export controls. Licensing arrangements for future commercial uses of the software have not yet been addressed by MSU and ISU.

\textsuperscript{17} The SNS software group has had full access to the DANSE codes on the developers' CVS repository almost since its inception.
Towards the end of the DANSE project we will plan a consortium, perhaps through the Neutron Scattering Society of America, that will ensure continued user input and development work on data analysis software. The low-cost extreme is something akin to the open source projects on sourceforge.net. This organizational structure has had some fine successes, but the sourceforge model is probably inappropriate.\footnote{In the 2008 time frame it might be appropriate for the SNS to support a software users’ group, consisting of interested users and some of the DANSE developers.\footnote{It is in the best interests of both the developers and the SNS to keep the developers engaged with DANSE beyond the term of the proposed construction proposal.}}

In the 2008 time frame it might be appropriate for the SNS to support a software users’ group, consisting of interested users and some of the DANSE developers.\footnote{Coordination of user support requires a professional manager. Quality control is another challenge for open source projects, and documentation tends to be poor.} It is in the best interests of both the developers and the SNS to keep the developers engaged with DANSE beyond the term of the proposed construction proposal.

14. Summary

The DANSE system for doing neutron science on a computer would offer access to the highest performance computing hardware, the flexibility to configure computing procedures to novel experiments, and a framework to which users can add new analysis procedures that interoperate with an established set of tools. The architecture supports different user interfaces, accommodating personal or institutional preferences. In particular, it is possible to build a user interface that is easy to use without compromising the underlying framework. Several years from now the international cyberinfrastructure for Grid-based computing will have arrived. DANSE is a scientific application that fits naturally on a Grid architecture, and we propose to start this migration.

Through science-based subprojects, the DANSE project can build a system to satisfy the data analysis needs of the main fields of neutron scattering research. The DANSE architecture is especially well suited for the development of new experiments that are supported by theoretical analysis, or cross-cutting experiments that rely on more than one set of data. A full DANSE system, designed in close cooperation with the SNS, would help bootstrap the scientific operations at the SNS, and would be readily adaptable for use at other neutron facilities. The proposed tasks in the science subprojects would build enough interoperable components to establish DANSE as the system of choice for further software development by neutron scattering researchers.

In the year 2004, the specifications and concepts for the DANSE system were transformed into a project to build it, with funds from a CED grant from the NSF IMR-MIP program. Reliable assessments of effort and risk were the major activity of the DANSE investigators over most of 2004. Efforts were consolidated, and many tasks were re-structured to minimize risk. Sections 3–11 of this proposal present the Work Breakdown Structure to level 4 in detail, sometimes with level 5 detail. Sequencing of tasks emphasized the early development of the DANSE system infrastructure, and considered the SNS schedule for instrument commissioning. The proposed budget was calculated by summing the costs for all tasks in a given year.

By offering ease of use to inexperienced users, and by welcoming the intellectual capital of expert users, DANSE will enhance the scientific impact of neutron scattering research.

\footnote{The neutron scattering communities in the U.S. sometimes coalesce into “users’ groups,” which meet annually in a small conference at their neutron facility. Broad participation is encouraged, but user feedback to the facility management occurs through an executive committee of about 7-10 persons, elected by the membership of the users’ group. The facility provides funding for the meetings and some expenses.}
References

4. TeraGrid http://www.teragrid.org/
5. Center for Advanced Computing Research http://www.cacr.caltech.edu/
6. VASP Group, Theoretical Physics Department, Vienna http://cms.mpi.univie.ac.at/vasp/
9. WIEN2k http://www.wien2k.at/
29. see http://www.bannedbygaussian.org/
40. A. A. Coelho, TOPAS V2.0: General profile and structure analysis (2000).
67. A. Guinier, X-ray Diffraction in Crystals, Imperfect Crystals, and Amorphous Bodies (Dover, Mineola, NY, 1994).
77. For example, Bede REFS 4.0 http://www.bede.co.uk/overview.php?overviewID=1015939212.63555
89. Center for Technology in Learning and Teaching (CTLT) http://www CTLT.iastate.edu
90. Toying with Technology (TWT) http://ww.eng.iastate.edu/twt