– DRAFT –
PROJECT EXECUTION PLAN FOR THE `DANSE` SOFTWARE PROJECT

Submitted by:

Brent Fultz, Principal Investigator, `DANSE` Project

Concurrences:

Kenneth W. Herwig, Co-Principal Investigator, `DANSE` Project

Michael A.G. Aivazis, Co-Principal Investigator, `DANSE` Project

Approval:

Guebre Tessema, Division of Materials Research, NSF

DRAFT – The `DANSE` Project Execution Plan – DRAFT

SNS Document DANSE-00-PN0001-R01(?)

Brent Fultz
California Institute of Technology, W. M. Keck Laboratory, Pasadena CA 91125 USA

December 1, 2008

Abstract. A large software construction project, built by university teams for use at the Spallation Neutron Source and other national neutron facilities, with funding from the National Science Foundation, is a challenge to manage and execute. The plan for doing so is presented here.

Keywords: project execution plan, `DANSE`, SNS, management, schedule, cost, baseline scope, project controls, software

Table of Contents

Table I. Acronym List

BES	Office of Basic Energy Sciences
BAC	Baseline Budget at Completion
Caltech	California Institute of Technology
`DANSE`	Distributed Data Analysis for Neutron Scattering Experiments
DOE	Department of Energy
EAC	Estimate at Completion
ES&H	Environment, Safety and Health
FEM	Finite Element Method
FY	fiscal year
GUI	Graphical User Interface
HQ	Headquarters
SDT	Software Development Team
IPT	Integrated Project Team
ISM	Integrated Safety Management
MC	Monte Carlo
MD	Molecular Dynamics
NFDD	Neutron Faciliites Development Division
NSF	National Science Foundation
NSSD	Neutron Scattering Science Division
ORNL	Oak Ridge National Laboratory
ORO	Oak Ridge Operations Office
PCR	Project Change Request
PD	Program Director
PEP	Project Execution Plan
PI	Principal Investigator
PM	Project Manager
QA	Quality Assurance
SNS	Spallation Neutron Source
SQRL	Software Quality Research Laboratory
TPC	Total Project Cost
TPS	Total Project Scope
WBS	Work Breakdown Structure

1. Introduction

This document is a Project Execution Plan (PEP). It outlines the plan of the Principal Investigator to execute a project to construct a software system for distributed data analysis of neutron scattering experiments (`DANSE`), to be used at the Spallation Neutron Source (SNS) in Oak Ridge, Tennessee, and by the international community of neutron scattering scientists. The present document was prepared in August 2008.

2. Mission Need

The Spallation Neutron Source (SNS), constructed in Oak Ridge, Tennessee with a budget of B$ 1.411, was for a time the world's largest science construction project. It began operations in 2006, and in 2007 it began to 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.

Software produced by the `DANSE` project will perform data analysis to facilitate scientific interpretations of data acquired from the instruments at the Spallation Neutron Source. The `DANSE` system, or its parts, will 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.

The `DANSE` system will provide users a seamless access to high-performance computing resources, configurable procedures for data analysis, and compatibility with a user's favorite tools. The specific tasks in the `DANSE` project were selected to satisfy the requests from a broad range of neutron scientists in workshops, surveys and polls. The proposed `DANSE` system will 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).

3. Project Description

3.1. Technical Description

The `DANSE` project contains two development efforts. The first is an activity centered at the Center for Advanced Computing Research at Caltech for developing the software tools and infrastructure needed for modular computations in neutron scattering research. The second activity will build the modules for scientific computations in the different subfields of neutron scattering research. `DANSE` will be a software system for doing the computations needed in experimental neutron science, integrated with software for doing computational neutron science. `DANSE` will be organized with the well-known *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

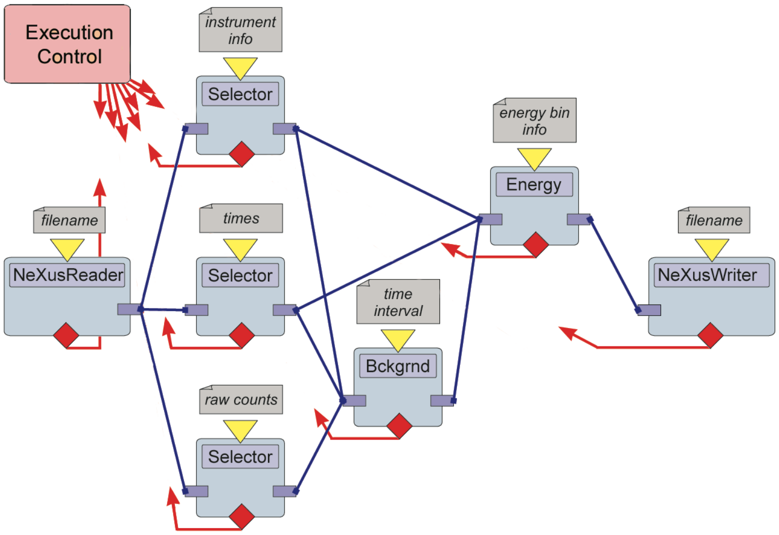

Figure 1. A data reduction application, built from components. The figure shows concepts for execution control, user input directly to components, and component cycle life management.

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 to the framework that could be passed to the user interface.

Technically, `DANSE` is a "component-based runtime environment." This means that the components are pre-compiled, and interconnected by the user at runtime.[1] 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.[2] 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. Users could deploy parts of `DANSE` locally on their laptops, 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. Key functionalities of `DANSE` are summarized in Table II. Appendix C describes the integrated applications to be

---

[1] Some software components could be: C++ codes for data reduction, user-interactive graphics packages, commercial data analysis environments such as `Matlab`, `Fortran` codes for electronic structure calculations on Beowulf clusters, and Monte Carlo simulations of the processes of neutron scattering.

[2] The user interface is a layer independent of the components, and can be replaced or modified without affecting the core functionalities of `DANSE`.

Table II. Key Functionalities of `DANSE` System

Subfield	Reduction	Modeling	Simulation
Diffraction		Real and k-space	
Eng. Diffraction		self-consistent	FEM
SANS		Real and k-space	MD
Reflectometry	Liquids, Magnetism	Reflfit	MD
Inelastic	ARCS, SEQUOIA	phonon, magnon,	MD
		chemical	ab-initio

Group	Support	Software Tools	Release Tasks
Central Services	developer training	framework, build system	management
Common Algorithms		graphics, GUI, optimization	
SQRL	developer training	testing tools	testing

built for each subfield of neutron scattering research. Further detail is provided in the `DANSE` Work Breakdown Structure (WBS).

3.2. OPERATIONAL GOALS

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

- enables new types of science in all major subfields of neutron scattering research,
- provides the basic data analysis capabilities that are available today,
- provides a coherent framework onto which software components can be added by scientists,
- is available at the SNS, and
- is maintainable after the end of the project.

3.3. KEY FUNCTIONALITIES

Key functionalities for `DANSE` are presented in Table II. More detailed explanations of these functionalities are described in detail in the `DANSE` WBS.

3.4. PROJECT MILESTONES

Project milestones are presented in Table III. The milestones are based on releases, which occur approximately annually ahead of NSF project reviews. Over the releases, the `DANSE` software will progress from prototype status to production software. The quality standards for the release of components and applications are listed in Section

Table III. Proposed Milestones at Level 2

Prototype Components	August 2004 (achieved)
Baseline Design Review	Dec. 15 2005 (achieved)
Prototype Release plus Day-One Components	Oct. 2006 (achieved)
Rebaseline	April 2007 (achieved)
α-Release plus Day-One Components	Mar. 2008 (achieved)
Second Year Review	May 2008 (achieved)
Rebaseline	Oct. 2008
β-Release	Mar. 2009
Construction Review	May 2009
Software Release 1.0	Mar. 2010
Readiness Review	Mar. 2010
Software Release 1.1	Mar. 2011
Project Complete	May 31 2011

8. In each release there is expected to be a mix of components and applications that have met different quality criteria. For each full release of the `DANSE` system listed in Table III:

- The goal of the prototype release is education of the `DANSE` team in the practice of releasing software.

- The alpha release is intended for internal use by members of the `DANSE` team.

- The beta release will be subjected to usage testing with interested users outside the `DANSE` team. The results of this usage testing and quality assurance testing will be a central part of the subsequent Construction Review.

- The Release 1.0 is expected to provide robust functionality for the baseline project scope. Some patches and updates may be issued over subsequent months.

- The Release 1.1 shall address feature requests from users of Release 1.0. This release may also include contingency functionalities that were not available in Release 1.0.

3.5. PROJECT REVIEWS

The release dates listed in Table III should be coordinated with the reviews of the `DANSE` project by the NSF. Scheduling a review 2 months after a release will allow enough time for the contents of the release to be evaluated. Based on the results of this evaluation, a detailed plan for the next year will be prepared. This detailed plan will be reviewed internally and presented at the review.

Table IV. Budget Periods and Funds for the `DANSE` Project

Fiscal Year	Construction Funds	Duration	Dates
FY2006	$ 1,441,799	9.5 months	6/1/06 - 3/14/07
FY2007/8	$ 2,583,296	14.5 months	3/15/07 - 5/31/08
FY2008/9	$ 2,956,278	12 months	6/1/08 - 5/31/09
FY2009/10	$ 2,827,653	12 months	6/1/09 - 5/31/10
FY2010	$ 2,164,244	12 months	6/1/10 - 5/31/11
5 year total	$ 11,973,270	60 months	6/1/06 - 5/31/11

4. Resource Requirements

4.1. BUDGET PERIODS

Authority and responsibility for managing the `DANSE` IMR-MIP construction program resides with the National Science Foundation. Specific authority for managing IMR-MIP Program falls under the authority of the `DANSE` Program Manager within the Division of Materials Research. The NSF will provide funding for the `DANSE` Project, and has delegated to the `DANSE` Principal Investigator the responsibility for the design, fabrication, installation, and the initial stages of commissioning of the `DANSE` system. These funds shall be provided in five budget periods presented in Table IV.

4.2. DISBURSEMENTS OF FUNDS

Funds will be made available to the NSF through legislation passed by the U.S. Congress. Budgets adequate to support the project baseline will be submitted by Caltech to the Division of Materials Research of the NSF, following the schedule in Table IV. The funds will be disbursed to Caltech as a university cooperative agreement, with Brent Fultz as Principal Investigator. Cost-reimbursable university subawards for scientific subproject support shall be established between Caltech and Columbia University, Iowa State University, University of Tennessee at Knoxville, and the University of Maryland as required.

Figure 2 shows the channels for financial disbursements within the `DANSE` project. The Principal Investigator, Brent Fultz, is responsible for directing all funding in the `DANSE` project, and for reporting all costs and financial status to the NSF. This includes coordination with Dr. Guebre Tessema, the `DANSE` NSF Program Manager. It is likely that the distribution of funds between institutions will vary as the `DANSE` project requires different services from different organizations. Decisions on these distributions will be made by the Principal Investigator through actions of the Change Control Board.

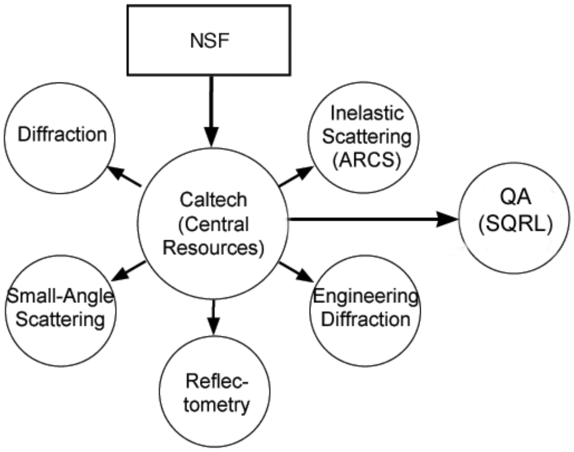

Figure 2. Funding channels for the `DANSE` project.

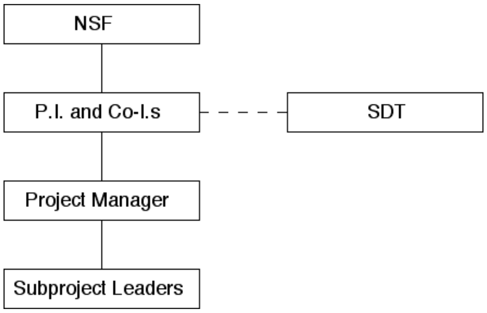

Figure 3. DANSE Project Organization Relationships.

5. Management Structure

The `DANSE` Project Management Organization is shown in Fig. 3. The responsibilities of each group are described next.

5.1. NSF Division of Materials Research: IMR-MIP Program

Within the National Science Foundation, the Division of Materials Research is responsible for monitoring the `DANSE` Project. The Division of Materials Research shall:

- Review and approve the technical, cost, and schedule baselines at the Work Breakdown Structure (WBS) level 4.
- Monitor the technical, cost, and schedule milestones of Tables III.
- Review the `DANSE` Project. Site visit reviews will be scheduled after each release. The NSF shall review the monthly progress report submitted by the project. This report will include technical progress and earned value performance data.
- Provide funds on a timely basis, and provide timely information on changes in the funding profile. Work with the `DANSE` team to ensure a suitable funding profile.

- Coordinate with the DOE Office of Basic Energy Sciences as required.

5.2. Caltech: Principal Investigator

The Principal Investigator, Brent Fultz, is responsible for executing the project. Besides the role of project coordinator, other specific responsibilities of the Principal Investigator are:

- Ensure the measurement of project performance against established goals, including technical performance, cost levels, and schedule milestones.
- Organize meetings to monitor and coordinate the science and software efforts.
- Monitor monthly project costs, and provide this information to the NSF as requested.
- Approve and disburse project funds.

5.3. Software Architect

The Software Architect, Michael Aivazis, is responsible for the design and integration of the technical systems of the `DANSE` system. He is located at Caltech where he interacts closely with the technical staff and reports to the PI.

5.4. SNS Liaison

The SNS Liaison, Kenneth W. Herwig (formerly Ian S. Anderson), is responsible for coordinating the interactions between the technical staff at the SNS and the members of the `DANSE` project.

5.5. Project Manager

The person serving in the central position of Project Manager, Mike McKerns, is located at Caltech. The Project Manager will coordinate the construction across all tasks of the `DANSE` project. The Project Manager reports to the PI. The Project Manager maintains the project plan and the earned value system.

5.6. Interface Between `DANSE` and the SNS

The software architecture is designed by Dr. Michael Aivazis, Member of the Professional Staff at the Center for Advanced Computing Research at Caltech. Steve Miller of the SNS is involved in major decisions such as data storage, data standards, software architecture, programming languages, and user interfaces. The Neutron Scattering Science Division of the SNS will ensure that appropriate staffing levels are maintained to install and maintain the `DANSE` system at the SNS. The `DANSE` system will be documented and components certified according to SNS policies. Particular attention is needed for coordination with the data reduction functions and the data access functions of the SNS software system.

Table V. Overview of Subcontracts

WBS No.	Name	Leader	Institution
6.0	Diffraction	Billinge	Mich. State U.
7.0	Engineering Diffraction	Üstündag	Iowa State U.
8.0	Small-Angle	Butler	U. Tn.
9.0	Reflectometry	Kienzle	U. Md., NIST

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. The `DANSE` software technology will be consistent with these policy needs.

The intellectual property officers at Caltech, the University of Maryland, the University of Tennessee, Michigan State University, and Iowa State University have agreed that whenever possible, software developed under the `DANSE` project shall be distributed under the nonrestrictive BSD license in Appendix B.

5.7. Interface Between Caltech and the Subprojects

Table V lists the subcontractors in the `DANSE` project. The individuals listed are responsible for control accounts at the Project's WBS Level 3.These persons serve as the Control Account Managers (CAM) of the scientific subprojects, and are responsible for communicating monthly financial information to the SNS Project Controls system. These subproject leaders report to the Project Manager.

A Statement of Work accompanying the budget requests from the four subproject institutions (Michigan State Univ., Iowa State Univ., Univ. Tennessee, Univ. Maryland) to the California Institute of Technology will be the formal basis for cooperation on the `DANSE` project. This SOW will identify specific tasks to be undertaken by the university subcontracters in the `DANSE` project, and the expectations for their resources and financial support.

The Caltech Central Services effort shall provide technical support for the subprojects. The expectation is that early in the construction project, the subprojects will be able to develop components on their own. Nevertheless, Caltech is expected to assist with component development, code review, documentation, testing, and release management.

5.8. Interface Between the `DANSE` Project and the Neutron Scattering Community

Software developers, not funded by the `DANSE` project, have been regular attendees at the Workshops and weekly videoconferences that discuss technical issues. The most active persons are listed in Appendix A as the `DANSE` Software Development Team (SDT). All have given advice on both the science and technology of the `DANSE` project, and have been given help in the use of `DANSE` software. The SDT will help evaluate the beta release of the `DANSE` software. This SDT is expected to grow as others express interest in the project. Members will serve at the discretion of `DANSE` management.

6. Project Baseline

6.1. Acceptance Criterion

The `DANSE` project is forecasted to complete on May 31, 2011. The functionality specified in Section 3.3 shall be obtained prior to project completion. Project completion approval is based on:

- demonstration of the essential functionalities listed in Table II,
- successful completion of a usability test, to be planned in consultation with the SNS,
- having the `DANSE` system installed on a computer resource usable by the SNS, and compatible with user access requirements,
- transmission of documentation and source code to the SNS.

After these conditions are satisfied, the SNS will assume user support for the `DANSE` system, at least for those components that are certified by the SNS.

6.2. Work Breakdown Structure

All work required for the completion of the `DANSE` project is organized into a Work Breakdown Structure (WBS). The WBS contains a complete definition of the scope of the project and forms the basis for planning, execution and control of the project. It includes the following 11 categories at level 2.

6.2.1. *WBS Category 1.0 – Milestones and External Drivers*

The first category is a list of milestones and external drivers.

6.2.2. *WBS Category 2.0 – Project Integration (B. Fultz)*

This category is primarily support for the core project team, including a full-time Project Manager and administrative personnel. Communications, workshops, and meetings are also in Category 2.0.

6.2.3. *WBS Category 3.0 – Infrastructure and Support (M.A.G. Aivazis)*

Support for the software developers of the `DANSE` project will be provided by hardware, software, and personnel centered at Caltech. Increasing the flexibility and usability of 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, release management, and quality assurance.

6.2.4. *WBS Category 4.0 – Central Services: Software Engineering (M.A.G. Aivazis)*

The core software engineering effort is a continuation and extension of the ongoing development of the `pyre` framework at Caltech. A set of standard graphical/plotting environments have been integrated into the `DANSE` framwork, and these will be expanded and improved. The construction of an infrastructure for creating GUIs and interfaces to web portals is a priority, as it will allow subproject developers to focus on their specific scientific tasks. The development of the distributed computing capabilities of `DANSE` is included, with Grid compatibility scheduled later in the project.

6.2.5. *WBS Category 5.0 – Common Algorithms for Scientific Subprojects (M. McKerns)*

Components that provide core scientific tasks such as optimization routines and interfaces to numerical libraries must be constructed to serve the needs of many of the data analysis and simulation applications. Common frameworks for data reduction and instrument simulation are being developed and integrated, as will common tools for crystallography and materials theory. As many of these software components are common to two or more of the science subprojects, collecting these common tasks into a separate subproject minimizes the duplication of effort. Although coordinated by the Project Manager, each individual task is owned by the group that is best qualified to work on it. These common tasks tend to be scheduled earlier in the `DANSE` project.

6.2.6. *WBS Category 6.0 – Diffraction (S.J.L. Billinge)*

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 the initial scope will be to wrap existing `PDFFIT` and Rietveld codes into the `DANSE` framework. The package `PDFFIT` will be redesigned to make it extendable to small molecules, clusters, liquids, and periodic structures. Another important effort is the detailed design of the rebuilt regression modeling programs with careful diagramming and pseudo-code generation.

6.2.7. *WBS Category 7.0 – Engineering Diffraction (E. Üstündag)*

The commissioning of Vulcan at the SNS is scheduled in 2008, so 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 instruments in engineering diffraction. Tasks on 3D finite element codes, inverse problem and microstructural analyses are more research-oriented and scheduled for the later years of the `DANSE` project.

6.2.8. *WBS Category 8.0 – Small-Angle Neutron Scattering (P. Butler)*

The SANS activity will build standard analysis and model fitting applications, including components for the simulation of real-space 3-D molecular systems. This effort will rely in part on tools dveloped in WBS Categories 4.0 and 5.0. The efficient development of user interfaces for SANS also requires tools developed in WBS Category 4.0. Further, the SANS instrument at the SNS is scheduled for commissioning in 2007, thus the peak in the funding profile of the SANS subproject is in years 3 and 4 of the `DANSE` project.

6.2.9. *WBS Category 9.0 – Reflectometry (P. Kienzle)*

The first tasks are to make the core functionality of reflectometry data reduction available to the SNS for day-one operations of the two reflectometers. The code development is consistent with adaptation to the `DANSE` framework, but some of this integration will be postponed for later in the project. Viewing and reduction of SNS instrument data is the highest priority. This will be followed by completion of basic analysis components. After modeling tools are in place, work will proceed on diffuse scattering analysis and full experiment simulations.

6.2.10. *WBS Category 10.0 – Inelastic (B. Fultz)*

The first tasks of the inelastic scattering effort under the `DANSE` project will be to adapt the ARCS data reduction software for other SNS inelastic spectrometers, especially SEQUOIA. We will work with the SNS software team to help them adapt the ARCS data reduction software to other instruments such as CNCS. The next tasks involve integrating molecular, phonon, and spin dynamics simulations from WBS Category 5.0 into the `DANSE` system.

6.2.11. *WBS Category 11.0 – Education and Outreach (M. McKerns)*

The members of the `DANSE` project will work with L. Genalo of the Iowa State University to seek funds for a program for K-12 education, *Toying With Technology.* If funded, this effort will support student-teacher workshops in the second year, with content provided by members of the `DANSE` project. Funding for the “hands-on” nanotechnology high school curriculum will be available during most of the `DANSE` project. Each subproject has an annual budget for the support of minority undergraduate students. Graduate students and especially postdoctoral fellows are the backbone of the development in the science subprojects, and are supported through specific tasks under other WBS categories of the `DANSE` project. The development of textbook content for inelastic scattering, diffraction, and engineering diffraction will be undertaken by the `DANSE` investigators.

6.3. PROJECT SCHEDULE

The project schedule is driven by early needs to provide software development tools and infrastructure, and day-one suppport for SNS reflectometers and inelastic instruments. The science subprojects follow the commissioning dates of the instruments at the SNS as they come online. Approximately, the years 2006 and 2007 focus on developing tools, and delivering software for inelastic scattering and reflectometry. The years 2008 and 2009 have a stronger emphasis on diffraction, engineering diffraction, and SANS. The year 2010 emphasizes the process for managing revisions to the software, driven substantially by user feedback, culminating with a release version 1.1.

7. Project Controls

7.1. PROJECT BASELINE AND ESTIMATES OF BASELINE AT COMPLETION

7.1.1. *Contingency*

Cost contingency is not allowed under the NSF IMR-MIP Program guidelines. The `DANSE` project therefore has scope contingency, but the scope contingency can be approximated as a dollar equivalent. Since costs are assigned to tasks in the `DANSE` project, scope contingency and cost contingency require similar management. In line with this, the costs associated with the identified scope contingency are treated as a cost contingency, and the use of this cost contingency will follow the change control procedures outlined in this document. The remaining scope is baseline scope, and the costs identified for this scope form the baseline cost estimate or Budget at Completion (BAC).

An Estimate at Completion (EAC) for completing the baseline scope reflects the best forecast by project management personnel of the final cost of each element at level 4 of the WBS. The EAC will be updated comprehensively at least annually, and will be updated monthly as better cost information is obtained for individual WBS elements.

The project contingency based on the EAC is defined as the difference between the Total Project Cost (TPC, the total cost of construction including contingency), and the EAC. This project contingency based on the EAC serves as a working estimate of the remaining project contingency. The actual remaining contingency is the TPC minus the actual baseline project cost. The total actual cost at completion cannot exceed the TPC, so the use of contingency must be managed under this constraint.

7.2. MANAGEMENT OF CONTINGENCY

Contingency funds are allocated as needed to complete the tasks of level 4 of the WBS in excess of the baseline cost. They are also allocated to add scope to the project baseline under the conditions of Section 7.3.2. The contingency management for the `DANSE` project ensures that the EAC remains close to the approved baseline cost. The EAC is updated monthly based on actual cost and performance data. The baseline project cost is updated by a procedure for approving Project Change Requests (PCRs). These PCRs shall be used to ensure that the EAC and the baseline project cost are as similar as practical. Principles of contingency management are:

- The `DANSE` Principal Investigator and the `DANSE` Project Manager have the authority to initiate a PCR to assign contingency from that available during each budget period in accordance with the project change control procedures of this section.

Table VI. Change Control Thresholds

	NSF `DANSE` Program Manager	Change Control Board
Technical	Any change in scope or performance that affect mission need requirements	Any significant change to the project scope as defined in the `DANSE` Proposal, Baseline Design, or Table II
Schedule	Any delay in a milestone from Table III by 6 months or more.	Any delay in a milestone from Table III by 3 months or more
Cost	Any change in Total Project Cost. Any baseline change equivalent to the redirection of funds more than k$ 100.	Any baseline change equivalent to the redirection of funds more than k$ 30 from one subproject to another.

- All differences between the EAC and the baseline project cost and all baseline changes shall be traceable.
- Any changes in the cost, schedule, or technical baseline require a PCR, which requires Change Control Board (CCB) approval according to Table VI before the revised baseline takes effect. Any changes in the cost baseline require the transfer of budget allocation between contingency and baseline.
- The difference between the EAC and the total baseline project cost must be maintained to be less than k$ 200 by the use of the PCR process as necessary.
- PCRs may be needed more frequently to manage schedule or technical scope changes according to the thresholds in Table VII, and may also be used as desired to shift funds between different WBS level 3 or level 4 activities in the baseline.
- The NSF `DANSE` Program Manager shall be informed monthly of any actions of the Change Control Board concerning the usage of contingency funds.

7.2.1. *Principle for Change Control*

Change Control will be based on the difference between the running EAC and the controlled BAC. When the EAC differs from the BAC by amounts listed in Table VI, action is required to update the BAC if appropriate, or change the effort within the project.

7.3. Change Control

7.3.1. *Project Change Requests*

By comparing to the TPC, the contingency based on EAC can be calculated at any time. Project Change Requests (PCRs) shall be initiated as necessary to maintain the difference between the EAC and the baseline at completion (BAC) below k$ 200. The NSF Program Manager will be provided with monthly reports on PCR activity.

Contingency is not included in the baseline for external factors that cannot be reasonably foreseen or quantified, such as major regulatory changes, annual funding shortfalls (appropriation less than baseline funding level), or ramifications from acts of international terrorism. When such circumstances occur, they are treated as "directed changes," requiring work-around plans or additional schedule and budget allowances.

7.3.2. *Project Baseline Maintenance*

The `DANSE` performance baseline will be established using a rolling-wave approach in which the near-term work will be detailed by individual, resource loaded activities at WBS level 5 with durations not exceeding 4 months. The out-year activities, while also resource loaded, will be baselined at a higher, less detailed level. After each release, an Estimate to Complete (ETC) will be completed for the remaining work. As with the establishment of the original performance baseline, during this planning the activities for the next 12 months will be detailed carefully. Level 5 detail for future years will be maintained at a more summary level. Once the ETC has been reviewed by the Change Control Board, it will be officially incorporated as the project baseline by the Baseline Change Control Process.

If at any time the assessment of overall project performance shows a contingency estimate based on EAC that is a sufficiently high percentage of the remaining baseline work, the PI or Project Manager may initiate a PCR to add scope from the "deferred scope list" to the project baseline. Future risks and current progress will be evaluated as part of the decision on what is a "sufficiently high percentage" of contingency to warrant such action. Once initiated, such a PCR would be issued through the normal change control process.

The controlled project baseline and project plan will be developed with the help of the SNS Project Controls group. Formal baseline changes will normally occur after a project-wide estimate at completion (EAC) assessment. Additionally, baseline changes will be made when changes to individual elements of the WBS no longer provide a reasonable basis for performance measurement. The Project Manager will coordinate this activity with assistance from the sub-project leaders and the Investigators. In addition to an earned value assessment and an estimate of scope at project completion, a projection will be made of the status of the `DANSE` system at the milestone dates of Table III.

7.4. EARNED VALUE MANAGEMENT SYSTEM

The `DANSE` project will be managed in accordance with the processes specified by the Project Management Institute, and incorporate the "Earned Value Management System Guidelines" (ANSI/EIA-748) for applying earned value. The project therefore reflects the"best practices" of Project Management. The project's performance baseline will be measured with an integrated system in which the resource requirements and associated estimates required for accomplishment of the defined work scope will be planned and scheduled utilizing sequences of logic driven, resource loaded activities that are structured in accordance with the Project's WBS structure.

The `DANSE` project controls system prohibits retroactive adjustment to any data, whether it be earned value (BCWP), actual costs (ACWP) or plan (BCWS) data. All changes and/or corrections will be made in the current reporting month. While this policy may result in widely fluctuating data for the current month, it preserves the integrity of historical information and permits an auditable path for any and all changes to the project baseline. The project plan will be built and statused with Microsoft Project, and earned value tracked with Microsoft Excel.

7.5. DANSE Monthly Reporting

Each subproject manager shall complete a detailed estimate of the labor and material resources required for successful completion of their assigned scope. The `DANSE` Monthly Reporting Process consists of three components:

7.5.1. *A List of Bullets describing the technical Highlights during the month*

This will be submitted by each subproject manager in a format established by the project.

7.5.2. *Earned Value Performance Analysis (including variance analysis)*

Earned value analysis requires two monthly inputs: i. Schedule status for the month, and ii. actual costs incurred (from the financial system). The schedules will be statused at the activity level by the subproject manager (or his/her designee). Work will only be performed on authorized activities approved in the project baseline or added to the baseline based on approval of a Project Change Request (PCR). Actual start and/or completion dates will be entered against the activities, producing a working schedule. Comparing the working dates to the baseline dates will facilitate variance analysis. Schedule logic will allow the impact of behind schedule activities on downstream events to be reported and summarized. The actual cost data will be submitted by each subproject in a consistent format and as soon as it is available but no later than the 15th working day of the next calendar month. Estimates of earned value are described in Section 7.6.

7.5.3. *Estimate at Complete*

The Estimate at Complete (EAC) is the Project Manager's most recent assessment of the total scope of the `DANSE` project with its subprojects. This estimate includes known issues that have already occurred (e.g., a contract was awarded greater than the baseline value) or issues that are highly likely to occur (e.g., a known salary increase or additional staff requirements that have not yet occurred but are more likely than not to occur). Inclusion of an item or issue in the EAC does not imply authorization by the Project Manager. Instead, this serves as the basis for discussions on issues and a determination of a path forward toward resolution of the problem or acceptance of the issue as a draw upon contingency.

The sum of the EAC estimates will be compared against the total project contingency to ensure that, in the event that the EAC items cannot be mitigated and comprise a draw on contingency, the remaining contingency balance is sufficient to cover unspecified, unknown risks that could occur.

7.6. Earned Value Data

Several independent sources of information on earned valued will be used.

- The first, and most reliable, is the release itself with subsequent testing. Assessment of the contents of the release is a quantitative exercise, with the only uncertainty originating with the limits of testing and code examination. The standards to be met are described in section 3.4.

This assessment of earned value through software releases is thorough, but occurs only annually. For monthly earned value reports, additional sources of information on earned value are available.

- Level 5 subtasks have been developed for all level 4 tasks in the WBS. Completion of these level 5 tasks shall be monitored by the subproject leaders, and shall be used to provide monthly data for the earned value management system.
- Section 8 lists a series of steps for software component development. These steps in the release process (design, prorotyping, reviews, etc.) will be used to monitor the status of component development. This status information can be obtained by examining the code checked in to the SVN developer's repository, including documentation and unit test suites.
- It has become established practice that each SVN check-in is accompanied by a log message. These messages give valuable information on the progress on a particular task, especially after the style of the programmer is known.

The Project Manager will assess the accuracy of these different measures of monthly earned value will be tested by comparison to the contents of the prototype release after the first year. After this assessment, the usage of level 5 tasks and SVN information will be optimized for predictive accuracy in earned value tracking. After each release, the level 5 tasks for subsequent years will be updated in a "rolling wave" through the duration of the `DANSE` project.

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. These monthly earned value reports will be available to the `DANSE` management for distribution to project personnel, and to the NSF `DANSE` Program Manager. A monthly teleconference with the `DANSE` NSF Program Manager shall be used to discuss this cost and schedule performance information.

7.7. Management of Risk and Configuration

7.7.1. *Risk Assessment*

Identifying risks is the first step in risk management. The biggest risk in software development projects is usually a misperception of specifications [1, 2, 3, 4]. The `DANSE` project is organized to minimize this particular risk. The developers are themselves users of neutron facilities, and the SNS is well-represented in the project management. The present baseline design 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 accounted for in the establishment of the baseline cost estimate for that task, and in decisions regarding which scope was to be in the baseline. Each task or subtask was assigned a risk probability that increased with the innovation of the task.[3] A second factor was used to assess the impact to the `DANSE` project that would be caused by the loss of this task.[4] The Risk Factor is the product of these two factors:

$$\text{Risk Factor} \equiv \text{Probability} \times \text{Damage} . \tag{1}$$

This assessment of risk for each task was performed with thoroughness in the baseline plan, and will be performed as part of each annual Estimate to Complete of the `DANSE` project. The greatest value of the risk assessment has been to identify tasks to be eliminated, restructured, or developed with a risk mitigation plan in mind.

7.7.2. *Risk Reporting*

After the Baseline Review, 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 Oct. 1, 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 assessments of risks.

7.7.3. *Risk Mitigation and Configuration Management*

Risk mitigation plans are listed for the level 4 tasks in the WBS dictionary. Some of these involve rearranging the project schedule, or changing a dependency on a supporting software package. 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, PI, as leader. Responsibilities of the Risk Control Board are:

- Document the risks through the risk watch list, with updates after each Developers' Meeting, and more frequently if information is available.

---

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

[4] The increments were: 0.05 if alternative exists, workaround is obvious; 0.2 for acceptable loss of functionality; 0.3 for a major loss of functionality; 0.5 for a loss of a core functionality for subproject; 0.7 if fatal to subproject; 3 if fatal to the `DANSE` project.

- 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 change 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 configuration issues involve software component dependencies and how software components interact. Other configuration management responsibilities include:

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

8. Release Management Plan

Sucessful software development requires central management, a centralized systematic tracking system, a centralized common code repository, regular automated builds, and a structured design, review, and testing infrastructure.

8.1. Inception and Design Assessment

In the inception phase for the development of a software application, the subproject leader first prepares a vision statement for the purpose, functionality, and acceptance criteria for the component. This document is reviewed and approved by the Project Manager, and is entered into the project tracking system as a feature request.

The subproject leader next derives functional requirements and constraints, and describes critical use cases. The use cases and functional requirements are reviewed and approved by the Project Manager, then committed to a versioning repository (i.e., svn). A class structure of the software is written without methods, but with enough completeness so component architecture descriptions can be generated by automated tools such as doxygen.

The subproject development team next produces prototypes that provide the critical functionalities. The purpose of the prototype is to clarify requirements, and several prototypes may be built. Tests of behavior are constructed concurrently with prototyping, and are used by the Project Manager to assess how well the design can provide the required functionalities.

The Project Manager schedules a code design review with the subproject leader's development team. A design review team includes at least the subproject leader and Project Manager, and the team may include developers from Central Services and other members of the DANSE community. Design patterns and UML diagrams describing component architecture are reviewed. Design approvals from the review are documented and committed to the versioning repository along with any updates to the specification and design documentation. The Project Manager edits the feature requests in the project tracking system, as appropriate.

The subproject leader and Project Manager informally reassess the task's cost and development schedule in the WBS.

8.2. Elaboration Phase

In the elaboration phase, working software is built from the approved design. A full product prototype with core component architecture is built to support the critical use cases. An automated certification testing suite is designed and initiated.

A critical code review is scheduled by the Project Manager to assess the restructuring of the code into more generic or reusable components, and to confirm quality standards within the code. When the code review is complete, the code is transitioned to the main release branch of the versioning repository, and the Project Manager adjusts the feature requests in the project tracking system, as appropriate. A baseline architecture has been established, and an alpha build of the component is released.

This elaboration phase may include changes to architectural design, reassessment of risk, reassessment of cost, and some change in scope. The Project Manager and subproject leader again discuss potential changes to cost and schedule, and this information is communicated to the Project Controls group as appropriate.

8.3. Construction Phase

All the remaining features are designed and implemented into the component architecture. Test cases are generated simultaneously to verify all use cases, and an automated testing suite is constructed and executed for the full code. The focus of this phase is to optimize cost, effort, and product quality. A beta version of the component, now with full functionality, is released.

8.4. Transitional Phase

The last phase of the release cycle is entered when the component is ready to be deployed to the full target domain (i.e., platforms, operating systems, compilers). Debugging and usability improvements are performed based on acceptance testing (i.e., end-user feedback). Documentation is completed, and user manuals are updated. Certification testing is completed by the Quality Assurance team.

The software development cycle is considered complete and a numbered release is distributed to the public. Further development and design changes are considered as part of release cycles for new versions.

8.5. Release Criteria

Each DANSE release is composed of a set of individual components and integrated applications. All components and applications shall be classified by these criteria:

- *Prototype* denotes components or applications that have at least completed the inception phase.
- *Alpha* denotes components or applications that have at least completed the elaboration phase.
- *Beta* denotes components or applications that have at least completed the construction phase.
- *Numbers* such as 1.0 and 1.1 denote components or applications that have completed the transition phase.

We note that there will be a mix of components and applications in each release of the `DANSE` software (e.g., the beta release of the `DANSE` software may contain some components that have achieved 1.0 status, and some applications that have achieved alpha status). A complete plan for the components in each release, and their status of completion, is given in the `DANSE` Release Plan.

9. Quality Assurance

The `DANSE` system will 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 Neutron Scattering Science Division (NSSD) The Memorandum of Agreement between Caltech and the SNS will state that the `DANSE` project will follow the quality assurance practices of the NSSD, as is the case for neutron instrument hardware.

10. Environment, Safety, and Health

The `DANSE` system must follow the Environment, Safety, and Health plan for all SNS instruments, developed by the SNS Experimental Facilities Division.

11. Transition to SNS Operations

The `DANSE` system will evolve from an incomplete set of components into a production system of applications with high reliability, culminating in one major upgrade before the end of the project. At the end of the project, a high reliability system will be available at the SNS to provide user support during SNS operations. The transition plan will be steady and gradual, not an abrupt transfer of a final release to the SNS. Kenneth Herwig, Deputy Director of the Neutron Scattering Science Division (NSSD) of the SNS will coordinate the transition plan. He is a Co-Principal Investigator on this `DANSE` CNST proposal.

Members of the SNS software group and some SNS instrument scientists have attended all the `DANSE` meetings held to date. It is important for the SNS software group to become as familiar as possible with all technical aspects of the `DANSE` system,[5] and for the `DANSE` team to be as familiar as possible with the requirements and plans of the SNS.

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 [5], and making access to data both convenient and secure. A set of facility policies will be established by SNS personnel, including policies for data access, standards for metadata, and licensing arrangements for proprietary software. 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.

Some overlaps of responsibilities do occur, so close cooperation is needed between the SNS and the `DANSE` team on issues including:

- Security procedures must be consistent with SNS policy.
- Software components for basic data reduction and display must be certified by SNS personnel because they reflect on the quality of experimental results produced by the facility.[6]
- The user interfaces reflect on the user opinions of the SNS facility.
- The SNS will negotiate licenses for proprietary software, and maintain these licenses.
- Some of the `DANSE` components and applications will be certified by the SNS, some will be open source codes that are not certified, some will be commercial packages, and others will be private for the developers or scientists who wrote them.[7] The `DANSE` project will need to work with SNS personnel to decide the software that fits into these different categories of certification and usage.

A schedule for commissioning and transferring the `DANSE` system into SNS Operations will be presented at the Construction Review following the beta release. Nevertheless, 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 phases of the project. 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.

---

[5] The SNS software group has had full access to the `DANSE` codes on the developers' SVN repository almost since its inception.

[6] 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 ARCS code `reduction`, for example, and are able to maintain it and modify it for other instruments at the SNS.

[7] 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.

Near the conclusion of the project, the Principal Investigator and Project Manager will document their “lessons learned,” “what went right,” and “what went wrong” as a part of the final review.

It is in the best interest of the neutron scattering community to sustain the coordination between the software development community and the SNS beyond the term of the `DANSE` construction project. The `DANSE` team will also work with SNS to establish a process to ensure continued user input and development work on software for neutron scattering science.

Acknowledgements

I thank Barbara Thibadeau, R. Kent Crawford, Mike McKerns, Ian Anderson and Michael Aivazis for help with this document, and for correcting errors. This work was supported by the NSF IMR-MIP program under contract DMR-0520547.

References

1. Donald J. Reifer, Victor R. Basili, Barry W. Boehm, et al., “COTS-Based Systems Twelve Lessons Learned about Maintenance,” in *Lecture Notes in Computer Science Vol. 2959*, Editors: Rick Kazman, Dan Port (Springer-Verlag, Heidelberg, 2004) pp. 137 - 145.
2. B. W. Boehm and T. DeMarco, “Software Risk Management – Guest Editors’ Introduction,” IEEE Software **14** (3) May-June (1997).
3. T. Moynihahn, “How Experienced Project Managers Assess Risk,” IEEE Software **14** (35) May-June (1997).
4. R. C. Williams, J. A. Walker, and A. J. Dorofee, “Putting Risk Management into Practice,” IEEE Software **14** (75) May-June (1997).
5. NeXus Data Format Home Page `http://www.nexus.anl.gov`
6. J. Haines and L. Radcliffe, Project Execution Plan for the SNS Instruments – Next Generation (SING) Project, SNS Document SINGPRJ-10-PN0001-R00, October 2004.

Appendix A: Members of the `DANSE` Software Development Team (SDT)

The members of the `DANSE` Development Team have contributed to the `DANSE` Proposal to DOE, and have been willing to attend meetings of the IDT. This group and others receives regular e-mail communications about the `DANSE` project. An asterisk (*) denotes the `DANSE` subproject leaders.

* M. Aivazis, California Institute of Technology, Pasadena
I. Anderson, Spallation Neutron Source, Oak Ridge
* S.J.L. Billinge, Michigan State University, East Lansing
* P. Butler, University of Tennessee, Knoxville
* B. Fultz, California Institute of Technology, Pasadena
K.W. Herwig, Spallation Neutron Source, Oak Ridge
* P. Kienzle, University of Maryland, College Park
* M. McKerns, California Institute of Technology, Pasadena
R. McQueeney, Iowa State University, Ames
D. Mikkelson, Univ. Wisconsin, Stout
R. Mikkelson, Univ. Wisconsin, Stout
R. Osborn, Argonne National Laboratory
F. Trouw, Los Alamos National Laboratory
* E. Üstündag, Iowa State University, Ames

Appendix B: `DANSE` Software License Agreement

`DANSE` Software 1.0

COPYRIGHT AND PERMISSION NOTICE Copyright (c) 2008 California Institute of Technology. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, provided that the above copyright notice(s) and this permission notice appear in all copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

All source code included in this distribution is covered by this notice, unless specifically stated otherwise within each file. See each file within each release for specific copyright holders.

`DANSE` is the name of a software system under construction with U.S. National Science Foundation funding.

Usage of DANSE software shall be acknowledged in scientific publications as: "This work benefitted from DANSE software developed under NSF award DMR-0520547."

Appendix C: Application Programs

APPLICATIONS FOR WBS 6 - DIFFRACTION

6.1 DiffLAB

This will be a modular configurable structure modeling code, as opposed to traditional monolithic structure refinement and modeling codes. Monolithic applications are not flexible enough for modern materials structure problems – the classes of models employed are becoming more diverse and structural features often require specialized software implementations. Increasingly we need a model that depends on the specific problem under study: e.g., for a molecular system that is best expressed as a Z-matrix, or for a discrete nanoparticle with no boundary conditions, a periodic system, and so on. Incorporating other types of structural data such as EXAFS and x-ray data in the analysis is also important for DiffLAB. We may want to employ a fast local-search regression algorithm such as Levenberg-Marquardt, or increase convergence using a global Monte-Carlo or genetic algorithm, or in general combine both approaches in a basin-hopping scheme. We seek the flexibility to adapt the modeling software to the needs of a particular problem.

DiffLAB will allow a modeling code to be configured at run-time from components such as function calculators (that calculate different data spectra), regression algorithms and models. The target function being optimized can be specified by the user by adding terms for more than one function calculator depending on the data available. DiffLAB will be extensible: as new opportunities and methods arise new modules can be written in the future. DiffLAB will be an application that makes use of the components in DiffPy, the library of diffraction component modules, and the `DANSE` libraries in general, to build a configurable, operating, regression modeling program on the fly. This will allow expert users to quickly and straightforwardly to build customized regression codes.

6.2 SrRietveld

SrRietveld will be a complete package for most of the tasks currently supported in existing Rietveld programs, but with the speed to match data acquisition rates from POWGEN3 (this will require distributed operation), extensibility to grow in the future to meet future needs, and the ease and speed of use that has been demonstrated by the PDFgui application. The precise requirements are under development now, but an important feature will be regression fitting directly to parametric equations such as expected temperature dependences of multiple data-sets. It will also have versatile built-in plotting and structure visualization capabilities.

6.3 SrReal

This will have similar requirements to SrRietveld but will be an application for carrying out fits to real-space PDF data and total scattering data using Monte-Carlo approaches to obtain local and nanostructure information.

APPLICATIONS FOR WBS 7 - ENGINEERING DIFFRACTION

7.1 Data Analysis

This application will generate information crucial for engineering calculations, such as orientation dependent lattice strain, peak broadening and texture from neutron diffraction data. It will use Rietveld (full-pattern) analysis to fit a crystallographic model to diffraction data or single peak fitting to obtain *hkl*-specific lattice strain and peak broadening. It will then employ appropriate averaging schemes (e.g., based on mechanics models) and calculate specimen-specific engineering data such as strain tensors and orientational distribution functions. Some data analysis functions will be performed in real time for comparison to the expected outcome. It will then help the user find new strategies if results and expectations differ significantly.

7.2 Mechanics Modeling I: Finite Element Analysis

purpose: This application will first perform mechanics modeling of materials using the ABAQUS finite element software. It will then allow comparison of model predictions with experimental data. In this comparison an optimization analysis will be performed so that material parameters can be refined to fit the data.

input:
- Lattice strain data from diffraction (after Rietveld refinement)
- Macro strain data from extensometer (or strain gauges)
- Macro stress data from load cell and sample dimensions
- Material parameters (elastic constants, dimensions, strength, etc.)
- Sample models: Sample geometry: Cylindrical, dog-bone shaped, compact tension, weld
-Microstructure morphology: Unidirectional fibers, spherical particles, concentric single fiber, laminate composite

output:
- Phase dependent and macroscopic stresses/strains
- Optimized material parameters (e.g., plastic flow curve variables) after comparison with data

methods:
- FEA using ABAQUS as a Pyre component
- Optimization algorithms: leastsq, fmin, fmin_powell, genetic and artificial neural networks

7.3 Mechanics Modeling II: Self-Consistent Analysis

This application will concentrate on mechanics modeling of materials using the EPSC self-consistent model (SCM) software. It will also allow comparison of model predictions with experimental data. In this comparison an optimization analysis will be performed so that material parameters can be refined to fit the data. A similar approach will also be followed with other SCM codes, e.g., one developed at ISU for ferroelectrics. Finally, where it is appropriate, a comparison with FEA predictions will be performed.

7.4 Experiment Design and Simulation

This application will guide the user in planning an optimum experiment by (i) collecting information on specimens and scientific goals; (ii) simulating the responses of instrument and specimen based on conditions defined by the user; (ii) performing various sensitivity studies to optimize experimental parameters; (iv) offering expert advice to the user based on past data and experience in the field.

Applications for WBS 8 - Small Angle Neutron Scattering

8.1 Model-Independent Analysis

This application will read 1D or 2D reduced SANS data and allow various manipulations by the user including simple math, Inversion to the pair density distribution function $[P(R)]$, so called ab-initio fitting, and include simple linear analysis (e.g. fractal power law ($\log(I)$ vs $\log(Q)$), Guinier ($\log(I)$ vs Q^2), Debye (I vs Q^{-2}), etc), peak shape analysis, correlation length analysis, etc. Batch mode operation for parametric analysis (e.g. peak intensity as a function of time) will also be included.

8.2 Model Fitting Analysis

This application will allow the user specify objects in a 3D modeling space, either rotationally averaged or partially oriented and produce 2D and 1D scattering patterns. The Application will provide simple models with analytical functions to produce the scattering patterns as well as allow the user to build more complex low-resolution models or import structures from PDB files which can then be inverted to q space. The objects may contain pieces with different scattering length densities and portions of the structures may be manually moved and oriented. Optimization against 1D or 2D reduced data will include standard parameter optimizations as well as constrained conformational searches. Some capabilities for exploring interaction potentials through $g(r)$ will also be included.

8.3 Experimental Planning Tools

Given source and instrument parameters, this application will allow simulation of the raw data collected as a function of experimental parameters (e.g. time, sample thickness, and size) including some system dependent background such as incoherent scattering from the sample. This will make use of the models developed in the previous applications and convolute them with the instrument specific parameters (e.g. beam flux). MC simulation of the instrument may also be possible.

Applications for WBS 9 - Reflectometry

9.1 Reduction

purpose: Transform the data into scattering intensity in Qx-Qz, allowing observation and control of every step in the process. Further reduction to intensity versus Qz allows the data to be fit with traditional reflectometry fitting programs. If enough information is available, phase determination and direct inversion of the complex amplitude can be performed. Comparisons between related datasets will be part of the general viewing capabilities of the reduction application.

input: NeXus files containing - sample data - intensity - detector efficiency
input: Q resolution - background selection

output: $R(Q_z)$, $R(Q_x, Q_z)$, $r(Q_z)$, $\rho(z)$

method: detector corrections - rotation - dead time - pixel width
method: monitor corrections - dead time - rebinning - intensity scaling
method: monitor normalization
method: specular/background decomposition from detector image
method: back reflectivity corrections - absorption - absolute Q
method: polarization efficiency estimation and correction
method: footprint corrections for scanned measurements with fixed slits
method: phase reconstruction and direct inversion
method: contrast plots for comparing data sets

9.2 1-D Analysis

purpose: Fit specular reflectivity data to 1-D models of scattering length density, absorption, magnetic scattering length density and angle. Dynamic observation and control of the search space is important since reflectometry is a poorly defined inverse problem.

input: $R(Q_z)$
input: real$(r(Q_z))$ if available

output: $\rho(z)$, $\mu(z)$, $P(z)$, $\theta(z)$
output: model parameters such as SLD of the layer $\rho(k)$, roughness of the layer $\sigma(k)$, volume fraction, *etc.* depending on the model

method: modeling of nuclear and magnetic parameters
method: constrained assignment of parameters shared amongst models
method: automatic adjustment of parameters using combinations of global optimization (genetic algorithms, simulated annealing, restarts) and local optimization (Nelder-Mead, Levenberg-Marquardt)
method: uncertainty estimation for parameters

9.3 3-D Analysis

purpose: Fit off-specular reflectivity data to 3-D models of scattering.

input: $R(Q_x, Q_z)$

output: $\rho(\bar{z})$, $\mu(\bar{z})$, $P(\bar{z})$, $\theta(\bar{z})$
output: model parameters depending on the model

method: modeling of nuclear and magnetic parameters
method: constrained assignment of parameters shared amongst models
method: automatic adjust of parameters using global and local optimization
method: uncertainty estimation for parameters

APPLICATIONS FOR WBS 10 - INELASTIC SCATTERING

10.1 Reduction

purpose: Reduction will take experimental data files in NeXus or other formats from Direct Geometry Chopper Spectrometers, including LRMECS, PHAROS, ARCS, and SEQUOIA, and transform the data into scattering intensity, with outputs in physical units such as barns, meV and $Å^{-1}$. Standard corrections for instrument and background will be performed. The functionality of Reduction will include the capabilities of the IDL Pharos code, and the basic functions of Open Genie used for data reduction on LRMECS.

Reduction was available as release 1.0 and with bug fixes as 1.1. A release 1.2 of Reduction includes multiphonon and multiple scattering corrections. A release 1.3 will

1. introduce event-mode reduction
2. improve parallel reduction to enhance performance
3. emphasize the fine-tuning of functionalities for high-level reduction routines to meet users' needs
4. provide a preliminary web-based reduction application
5. start to provide new procedures for single-crystal reduction (see Note below).

Deployment of the ARCS data reduction software will be on the analysis clusters at the ARCS instrument, and on a cluster at Caltech. Reduction software for Pharos and LRMECS will be for single processor Linux and Mac OS computers. There is no plan for a release 1.3 of Reduction for LRMECS.

The user interfaces will include:

1. high-level python commands for advanced users to customize the reduction procedure
2. configurable python scripts run in command line for quick access to reduction capabilities
3. preliminary web interface for novice users

input: NeXus (or other formats) data files containing - sample data - empty can - vanadium calibration - flux monitor - time independent background - instrument characteristics such as moderator-sample distance, detector positions
input: data file selections - output directory selection - detector masks - parameters of axes of output histograms - parameters for time-independent background selection and vanadium calibration - UB matrix information

output: selection of neutron-weighted $S(Q)$, $S(\vec{Q})$, $S(E)$, $S(Q,E)$, $S(\vec{Q},E)$
output: data files in mslice format
output: $g_{\mathrm{nw}}(E)$ for phonons or magnons
output: calibration results (detector efficiencies)

method: Standard Corrections: - detector efficiency - auto determination of additional masks - incident energy - background - empty can - black absorber - normalization
method: single crystal reduction (TBA)
method: transformations to $S(Q)$, $S(\vec{Q})$, $S(E)$, $S(Q,E)$, $S(\vec{Q},E)$, assuming appropriate input on $\vec{Q}$

method: scattering corrections - multiple scattering - multiphonon scattering (incoherent) to give $g_{\mathrm{nw}}(E)$

Note: Capabilities for $S(\vec{Q}, E)$ from single crystals will be provided by generating input files to Mslice. Improvements to this capability will be pursued at the discretion of the inelastic subgroup and community needs.

10.2 Dynamics Modeling

purpose: Dynamics Modeling will optimize the parameters of a physical model of sample dynamics to fit reduced data. The optimized model can then provide other properties of the sample, such as the scattering after correction for neutron weights, or the separation of spin and phonon scattering.

Functionalities planned for this application include:

- Fits to experimental data of results from a Born – von Kármán lattice dynamics model or a Heisenberg spin dynamics model,
- Calculating experimental data with a Van Hove function $G(r, t)$, or a magnetic susceptibility function $\chi(q, E)$,
- Neutron weight correction (magnetic or nuclear),
- Separation of nuclear and magnetic scattering by Q-dependence.

Deployment of the dynamics modeling package will be 1) through the `DANSE` software repository, with a build script that has been tested on at least one Linux platform, and 2) as a web service that supports a computer cluster.

input: neutron-weighted $S(E)$, $S(Q, E)$, $S(\vec{Q}, E)$
input: initial parameters for model (eventually from database)
input: selection of optimizer, its numerical criteria, and execution directives

output: optimized parameters of the model
output: $g(E)$, (density of states) for phonons or magnons, corrected for neutron weighting
output: $E(\vec{q})$, $S(Q, E)$, $S(\vec{Q}, E)$ corrected for neutron weighting in incoherent approx.

method: fits of model to experimental $g(E)$
method: fits of model to experimental $S(Q, E)$
method: neutron weight correction (magnetic or nuclear)
method: separation of nuclear and magnetic scattering
method: models for forward calculations - BvK lattice dynamics model - Heisenberg spin dynamics model - Van Hove $G(r, t)$ - Magnetic susceptibility function $\chi(q, E)$
method: CLIMAX
method: optimizers for model parameters - Levenberg-Marquardt - Powell - Simulated annealing - TBA

10.3 Experiment Simulation

purpose: Experiment Simulation will include sample scattering within Monte Carlo simulations of instruments, and generate NeXus files of simulated neutron scattering. This package is extending the simulation framework and pyre-Mcstas to inelastic scattering, developed in WBS 5. Extensions to inelastic scattering include the development of inelastic scattering kernels and molecular dynamics modeling.

Deployment of the experiment simulation package will be 1) through the `DANSE` software repository, with a build script that has been tested on at least one Linux platform, and 2) as a web service that supports a computer cluster.

input: instrument geometry and configuration for ARCS, Pharos, SEQUOIA, and optionally CNCS
input: sample characteristics - geometry (shape) - composition (sizes and locations of scatterers) - physics (scattering kernel selections, scattering kernel characteristics) - tentative inputs to scattering kernels using dynamic model: - BvK - Heisenberg - $G(r, t)$ function - $\chi(q, E)$ function - Molecular Dynamics inputs.
input: sample environment - temperature - pressure - magnetic field - sample holder (container) - furnace - cryogenic system
input: user parameter selections for simulations - number of neutron packets - size of neutron packet - computing resource - computing nodes selection (if parallel) - output directory

output: NeXus file
output: simulated sample properties from dynamics model and optimization
output: simulated instrument properties - flux as a function of Q, E - resolution in Q, E

method: `DANSE` instrument simulation - `DANSE` new - McStas component - Vitess component
method: neutron recorder with data manager
method: phonon scattering from $E(\vec{q})$
method: phonon scattering from $G(r, t)$
method: phonon scattering from BvK
method: phonon scattering from molecular dynamics
method: magnetic scattering from $\chi(q, E)$
method: magnetic scattering from Heisenberg model
method: simulation of intermediate results - user selection of $S(Q)$, $S(\vec{Q})$, $S(E)$, $S(Q, E)$, $S(\vec{Q}, E)$, with input on $\vec{Q}$
method: - Molecular Dynamics, which generates trajectories from which the correlation functions of inelastic scattering are calculated and used as a scattering kernel.

10.4 Ab Initio Lattice Dynamics

purpose: Ab Initio Lattice Dynamics will generate the phonon properties of crystal structures of moderate complexity from first-principles electronic structure codes such as VASP. The quantities generated will include the phonon density of states (DOS) $g(E)$, dispersions $E(\vec{q})$ in selected directions, and the scattering functions $S(Q, E)$ and $S(\vec{Q}, E)$. The scattering function $S(\vec{Q}, E)$ can then be used in a scattering kernel to generate the sample scattering with a Monte Carlo simulation.

Deployment of the Ab Initio Lattice Dynamics package will be as a web service that supports a computer cluster.

input: Crystal structure
input: Simulation parameters for *ab initio* electronic structure code

input: User selection of quantity to calculate

output: $g(E)$, with or without neutron-weighting in case non-monatomic crystals
output: $E(\vec{q})$ in selected directions
output: $S(Q,E)$, $S(\vec{Q},E)$

method: VASP to calculate the atomic forces corresponding to atomic displacements from first principles
method: Symmetry analysis to automatically generate complete set of atomic displacements
method: Fitting of forces to generate interatomic force-constant tensor
method: Linear response method for dispersions in selected crystallographic directions
method: BvK lattice dynamics to generate $g(E)$, $E(\vec{q})$, $S(Q,E)$, and $S(\vec{Q},E)$
method: neutron weighting of $g(E)$ or $S(Q,E)$ or $S(\vec{Q},E)$

10.5 Science Package – Contingency Item

purpose: Science Package will be one example of a data analysis network built from modules developed in the `DANSE` project. This application and its documentation will provide a template for how scientists could use the components of the `DANSE` system to quickly build novel applications for data analysis. The functionality of this package will be decided after the `DANSE` beta release. The example here is representative – take a set of data files acquired at different temperature, and construct an anharmonic thermodynamic partition function.

input: NeXus files with parametric variation in T
input: weights of different input files

output: partition function $Z(T)$

method: script to run multiple cases of reduction and model to get $g(E)$
method: calculate $Z(T)$ from $g(E)$