

Lawrence Berkeley National Laboratory

Recent Work

Title

Software Methodologies for the SSC

Permalink

<https://escholarship.org/uc/item/6kg5j0g7>

Author

Loken, S.C.

Publication Date

1991



Lawrence Berkeley Laboratory

UNIVERSITY OF CALIFORNIA

Information and Computing Sciences Division

Presented at the Symposium on Detector Research and Development
for the SSC, Fort Worth, TX, October 15-18, 1990,
and to be published in the Proceedings

Software Methodologies for the SSC

S.C. Loken

January 1991



Prepared for the U.S. Department of Energy under Contract Number DE-AC03-76SF00098

1 LOAN COPY 1
1 Circulates 1
1 for 4 weeks 1 Bldg. 50 Library.
COPY 2

LBL-30257

DISCLAIMER

This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor the Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or the Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or the Regents of the University of California.

SOFTWARE METHODOLOGIES FOR THE SSC

Stewart C. Loken
Lawrence Berkeley Laboratory
University of California
Berkeley, CA 94720

Abstract

We review the issues related to software development for the Superconducting Super Collider. These include the methods for software design as well as the use of specific software technologies and commercial products.

Introduction

This report describes some of the considerations that will determine how we develop software for the SSC. We will begin with a review of the general computing problem for SSC experiments and recent experiences in software engineering for the present generation of experiments. This leads to a discussion of the software technologies that will be critical for the SSC experiments. We will describe the emerging software standards and commercial products that may be useful in addressing the SSC needs. We conclude with some comments on how collaborations and the SSC Lab should approach the software development issue.

Parameters for SSC Software

In this section we review the characteristics of the SSC computing problem. The critical areas are the following:

- Storage and data management
- Computing environment
- Software development

What is missing from this list is a discussion of the computing power required to analyse the experiments. The recent developments in RISC computers and their use in loosely coupled systems make it clear that the CPU needs of experiments can be satisfied. It will, however, require some care to assemble systems that work efficiently.

Storage and Data Management

Each experiment will generate 10 - 1000 Terabytes per year of raw data. In addition, the typical event processing will double the amount of data. Most of these

data will need to be accessible to physicists at remote institutions as well as at the SSC site.

The data management requirements are also severe. Some analysis will need to access very few variables for each event but need many events. On the other hand, to search for rare processes, it may be necessary to display all raw data from the detector in a 3-dimensional image.

The data must be organized to permit efficient access to both types of information. This presents a significant challenge to database technology.

Computing Environment

SSC physicists will need good interactive computing and efficient access to large databases. The environment should include statistical packages and powerful graphics. It will also be useful to have intuitive interfaces to analysis packages and data management tools. These should permit building complex applications from simple modules.

An important goal for the SSC software systems will be to make all programs appear as if they reside on the physicist's local workstation. The physicist should not care where the application runs or where the data are stored.

Software Development

Present experiments such as CDF now have approximately one million lines of Fortran code. It is typically developed by programmers who are not computing professionals and who are located at many different institutions in many countries.

The development of such large software systems would constitute a challenge to any group even using modern software engineering tools. In most experiments, however, the development process does not use formal methods or modern tools. As a result the software that exists is often unreliable and difficult to maintain.

*Supported by the U.S. Department of Energy under Contract No. DE-AC03-76SF00098.

Methodologies

The development of software systems for high energy physics experiments has been traditionally unstructured. Groups have defined some general guidelines regarding the use of common blocks for event data and other structures and some standards for FORTRAN code. Beyond this, individuals have been free to add software based on their own ideas of what was needed.

Recently, some groups have tried to introduce more formal methods into the process. These have been based on the "waterfall" model of software development. This assumes that the requirements can be clearly specified and that the design and code can proceed from those.

Structured Methods

In the past, discussions on software engineering seemed to focus on one methodology, Structured Analysis/Structured Design (SA/SD). It is becoming clear, however, that this is only one possible part of the solution to the problem of engineering software systems. The full solution will require coding rules, code management and distribution, and many other pieces.

It is interesting, however, to review the success that projects have had using SA/SD. It has been applied to a number of experiments including ALEPH[1], D0[2] and ZEUS[3]. The results have been mixed but, in all cases, the groups agree that the methodology did improve the software design. In all cases, however, the lack of tools with which to maintain the analysis and design was a major problem. It was not possible to update the documentation easily and, as a result, it became obsolete. The SA/SD methodology was not followed through the coding phase.

An important aspect of the analysis/design efforts was the modeling of data using the Entity Relationship Model[4] or the ALEPH Data Model (ADAMO)[5]. These tools provide a way to represent the complex relationships between data objects. Data modeling is an increasingly important part of the general problem of Database Management which is discussed below.

Problems with Structured Methods

As indicated, the introduction of SA/SD had some positive effects but was not a complete success. The lack of tools to support the methods was an important factor. Another important factor, however, was the fact that it is impossible to complete a full description of requirements early in the development cycle. The requirements will change as the experiment is designed. In addition, some of the software techniques must be tested before they can be introduced into the software.

These considerations lead to the idea of a prototyping step. This is similar to the method for designing hardware. Critical elements of the software are identified and tested in simple programs. The technologies can then be incorporated into larger system. Some of

the key technologies that will be needed for SSC experiments are discussed in the following sections.

Software architectures

It is clear that systems are becoming larger and more complicated. We need to improve the level of management on computing projects and we need to become more aware of the need to develop an integrated system. It will be difficult to enforce engineering rules across the project and for this reason, we need to look at system architectures that will isolate people who do not follow the rules.

Architecture is an over-worked phrase these days. In our context we mean a software structure that allows us to build robust programs. It has been clear for some time that we need to design systems that have small modules that work, and can be tested, independently. This approach has been demonstrated to improve programmer productivity and reduce software maintenance costs.

One technology for building modular systems that interoperate is called the software bus[6]. There are now a number of commercial implementations of this technology and there is increasing interest in defining a standard for a software bus. A pilot project using one bus standard was described by Grieman[7]. While this approach requires more effort in the implementation phase, the result, so far, has been modules that are easier to maintain and to reuse in other applications.

A formal approach to the development of software modules is Object-oriented Programming. While this approach is not yet wide-spread in High Energy Physics, some pilot projects show considerable promise. The paper by Kunz[8] provides an introduction to Object-oriented Programming techniques. These techniques have been used very successfully in the Reason Project described below.

Commercial products

Many of the software products needed by future experiments can, and should, be supplied by industry. Some obvious examples are databases, human interfaces and graphics. The use of industry standards will ensure that software can evolve as future systems are introduced.

Databases

It has become increasingly clear that database management is an important component of High Energy Physics software. The problems will be even more severe for SSC and LHC experiments with more complex detectors and more events to deal with.

The use of commercial Relational Database Management Systems and of home-grown packages has been reviewed by Rimmer[9]. Commercial systems have been used successfully by the LEP builders and by some of

the LEP experiments. These systems do satisfy, at least, some of the requirements of the machines and of experiments. There are, however, features that are needed and are not provided by the commercial systems and many groups have developed packages to provide the needed functions. Many of these have been built of the ZEBRA/RZ packages.

The optimal use of database systems requires the use of data modeling techniques to describe the objects in the database and the relationships among the objects. The Entity-relationship Model has been very successful in providing such a description. Some experiments have used this model as part of SA/SD or other software engineering. The ALEPH group has integrated data-modeling into their software design and has developed the ALEPH DATA Model (ADAMO). The ZEUS experiment has also adopted ADAMO.

The use of database systems and data modeling will continue to increase as the data management challenge increases. Groups need to experiment with these techniques and get experience with different products. To avoid becoming dependent on specific products, it is useful to adopt a high-level standard such as the Standardized Query Language (SQL).

Human interfaces

The development of intuitive user-interfaces is critical to improving the quality of scientific software. The popularity of graphical interfaces in the personal computer market demonstrates the need to follow this example in the development of programs for High Energy Physics. The Reason Project[10] has developed an interactive data-analysis package that utilizes the interface-building software that is supplied in the NeXT computer. The program does not yet have full functionality but it demonstrates the great potential for this approach. The ease with which the developers have developed the program indicates the benefits of using high-level software tools to develop scientific software.

Another new software product that is likely to have a significant impact on scientific computing is the Application Visualization System (AVS) from Stardent Computers. Using AVS, the physicist can develop sophisticated graphics applications without writing code. The D0 experiment has used AVS for 3-D event displays[11]. The approach shows great promise and will be used to develop more general tools.

The Use of Standards

The lifetime of experiment is now significantly longer than the career of a graduate student or post-doc. We need to recognize this in planning experiments. We cannot depend on students or post-docs to maintain code for the life of the experiment. We must design software that is reliable and can be maintained by others. The

use of software engineering tools for design, documentation and testing will be critical.

The present generation of experiments already spans more than one cycle in computing technology. This introduces new challenges for the software developer. For the SSC and LHC era, the problems will be even greater. Experiments must plan from the beginning to evolve as the technology changes.

To ensure that systems will survive changes in technology, new software should be developed to utilize commercial products to the greatest extent possible. All software, whether developed or purchased, should adhere to standards. The industry will then assist in moving critical software to new technologies.

We are already seeing standards having a significant impact of computing in High Energy Physics. The use of UNIX is widespread, both for high-end workstations and as a source of cheap computing cycles. The Internet protocols (TCP/IP) which were virtually unknown in the High Energy Physics community a few years ago are now an important part of the network traffic. X-windows and new graphics standards (GKS and PHIGS) are gaining acceptance. This evolution will surely continue. The use of standards will ensure that investments in software will maintain their value as technology changes.

Getting it all together

The development of software systems for the SSC experiments will offer many new challenges and will require a change in the way that physicists work. The effort should be a collaboration of three groups, the experiments, the SSC lab and the computing industry. Each of these can make a unique contribution.

The Role of Experiments

The experiments must plan and design computing systems with as much care as they put into any other aspect of the detector system. They should set priorities for funding all aspects of the detector including computing systems.

While the experiments will define the requirements for computing systems and software, they should work with laboratory management to define the scope of the SSC Lab central facilities. There should be a continuing interaction between the experiments and the SSC computing management to ensure that the laboratory support satisfies the needs of the experiments.

The Role of the Laboratory

The SSC Lab should provide centralized services for the experimental program. These are likely to include the following:

- Data storage and handling systems

- Networking
- Data acquisition and control systems
- Widely applicable software

The laboratory should establish a mechanism by which the experiments can guide the choice of centrally supported systems and can review the quality of the support.

The laboratory should also promote appropriate software standards. It should monitor developments in the computing industry and should act as a focus for industry collaboration.

The Role of Industry

While much of the needed software and hardware should come from industry, it will be necessary in many cases to modify or extend existing products. Physicists and industry must work together to understand the requirements of the experiments and to evaluate existing commercial solutions. As new solutions are developed they must be tested in the SSC applications.

Conclusions

We are beginning to see a major change in the way that physicists do their computing. The availability of high-level interfaces and other commercial products will have a very significant impact of future projects. As with any new tool or technique, we need to get experience with the new software technologies.

Future software systems will be developed by collaborations of experimenters, the SSC Lab and industry. We should begin now to form these teams to address some of the critical software issues.

References

- [1] G. Kellner, "Development of Software for ALEPH using Structured Techniques," in *Computing in High Energy Physics: Proceedings of the International Conference on Computing in High Energy Physics, Asilomar, 1987*, p 229.
- [2] J.J. Linnemann *et al.*, "The Use of SA/SD Methods in D0 Software Development," in *Computing in High Energy Physics: Proceedings of the International Conference on Computing in High Energy Physics, Asilomar, 1987*, p 245.
- [3] R. Loveless *et al.*, "ZEUS Hardware Control System," in *Proceedings of the International Conference on Computing in High Energy Physics, Oxford, England, 1989*, p 313.
- [4] P. P. Chen, *ACM Transactions on Database Systems* 1 9, (1976).

- [5] S. M. Fisher and P. Palazzi, "Using a data model from software design to data analysis: what have we learned?" in *Proceedings of the International Conference on Computing in High Energy Physics, Oxford, England, 1989*, p 169.
- [6] D. E. Hall *et al.* "The Software Bus: A Vision for Scientific Software Development", in *Proceedings of the International Conference on Computing in High Energy Physics, Oxford, England, 1989*, p 313.
- [7] W. Grieman *et al.* "Experience Using a Software Bus to Build Reuseable Scientific Software", in *Computing for High Luminosity and High Intensity Facilities: Proceedings of the Conference on Computing in High Energy Physics, Santa Fe, 1990*, p 276.
- [8] P. F. Kunz, "An Introduction to Object-oriented Programming Techniques for High Energy Physics", in *Computing for High Luminosity and High Intensity Facilities: Proceedings of the Conference on Computing in High Energy Physics, Santa Fe, 1990*, p 264.
- [9] E. M. Rimmer, "Databases for Large Detector Systems — Experiences at LEP and Future Needs" in *Computing for High Luminosity and High Intensity Facilities: Proceedings of the Conference on Computing in High Energy Physics, Santa Fe, 1990*, p 341.
- [10] B. Atwood *et al.*, "The Reason Project", in *Computing for High Luminosity and High Intensity Facilities: Proceedings of the Conference on Computing in High Energy Physics, Santa Fe, 1990*, p 320.
- [11] T. G. Trippe, Private communication.

LAWRENCE BERKELEY LABORATORY
UNIVERSITY OF CALIFORNIA
INFORMATION RESOURCES DEPARTMENT
BERKELEY, CALIFORNIA 94720