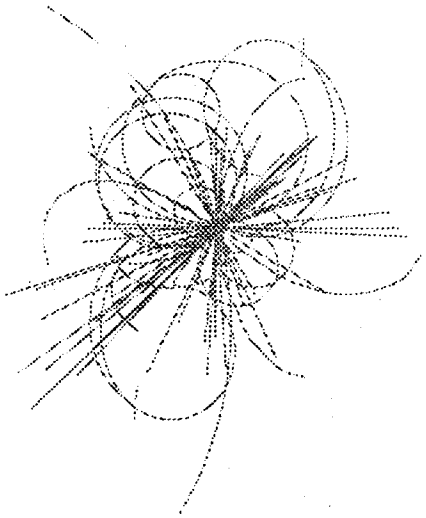
SSCL-SR-1181



HEPLIB'91

International Users Meeting on the Support and Environments of High Energy Physics Computing

SSC Laboratory, Dallas, Texas
September 19-20, 1991

Harald Johnstad
Physics Research Division
Superconducting Super Collider Laboratory*
2550 Beckleymeade Avenue
Dallas, Texas 75237

---

*Operated by the Universities Research Association Inc., for the U.S. Department of Energy under Contract No. DE-AC35-89ER40486

Preface

The HEPLIB'91 International Users Meeting on the Support and Environments of High Energy Physics Computing was held at the Superconducting Super Collider Laboratory (SSCL) in Dallas, Texas, September 19-20, 1991. This was the inaugural meeting of the HEPLIB User Group. Fifty-three registered participants discussed the current environments in high energy physics computing, including a broad range of technical issues, as well as information exchange and user support. These proceedings contain the invited and contributed talks presented at the meeting.

The participants met for two full days. A conference dinner was held at Milano's restaurant in Duncanville at the end of the first day.

The meeting was organized jointly by the SSC User's Office and the SSCL Library. On behalf of all participants and the International Advisory Committee, I would like to thank Phyllis Hale, Head of the SSC User's Office and Pat Kreitz, Head of the SSCL Library for their support. Special thanks to Joan Arruza, SSCL Library and HEPLIB'91 Secretary, for her patience and a smoothly organized meeting. Thanks also to other hard-working staff: Connie Coleman and Ljiljana Johnstad, SSCL Library, and Paige Pendleton, SSC User's Office. I also want to express my appreciation to those who worked behind the scenes from the Laboratory Technical and Building Services, who all helped to make this meeting a success.

Thanks are due to Ken Damrau, SSCL Physics Research Division, and Shane Davis, University of Texas at Dallas, for initiating the LISTSERV facility at UT Dallas, which will be used for information exchange for the HEPLIB User Group.

Last, but not least, I want to express my appreciation to all the participants and speakers in this inaugural meeting, and my personal thanks to all the members of the International Scientific Advisory Committee worldwide who were willing to work with me and to take time off their busy schedules to plan and define the appropriate scope and program for the meeting.

On behalf of the organizing committee,

Harald Johnstad
Chairman

TABLE OF CONTENTS

	Page
Preface	i
Meeting Notice	vii
Agenda	xi
List of Participants	xv
Conclusions	xvii
Welcome To SSCL *P. Leibold*	1
Opening Talk - Introduction to HEPLIB *H. Johnstad*	3
Customer View of HEPLIB in a Multi-vendor Environment *P. LeBrun*	19
HEP Code Compilation *A. Palounek*	27
CERNLIB Status and Implementation/Organization and Manpower *M. Marquina*	35
SCRI Software Products/Ideas for HEPLIB *S. Youssef*	41
SDC Requirements to HEPLIB *L. Price*	61
Overview of Physics Software at KEK *K. Amako*	75
Overview of Physics Software at DESY *J.H. Peters*	89
Overview of Physics Software at LBL *W. Koellner*	107
Overview of Physics Software at SSCL *H. Johnstad*	117
SLD Interactive Physics Analysis/Initial Experience with XMotif/ Comments on GEM *T. Johnson*	129

Table of Contents

Page

HEPLIB and Code Management/Evolution of the CMZ Code Management System
R. Brun 153

Standard HEP Particle Data Book on the Physicist Workstation
L. Garren (Presented by P. LeBrun) 161

CERNLIB Maintenance/From Patchy to CMZ/Installation Servers/ Improving the Test Phase/Information Exchange/CERN List Servers/ The ASIS World-wide CERNLIB Server/The WWW Project
M. Marquina, R. Berners-Lee, R. Caillian 179

Distributed Data Management for High Performance Data Systems/ IEEE Storage Standards
B. Coyne 205

DAFNE Software Requirements
E. Pace 241

Discussion of the Need for Active Promotion of non-HEP Applications of Advances in HEP Computing
V. Chaloupka 265

Initial Experience Porting CDF Software to RS6000
R. Hollebeek 267

Distributed Computing at CERN/The SHIFT--HOPE Expansion/ Experience with ISIS and Distributed GEANT/The Simulation Farm Project/The Visualization Farm Proposal
R. Brun 281

The SDC SHELL Project
S. Kunori 289

Parallelization Software on PDSF/The PSHELL User Interface
B. Traversat 303

A Parallel Version of GEANT
P. Rulon, S. Linn 321

CALOR on PDSF and Intel
T. Gabriel, T. Handler, B. Moore 329

Physics Analysis Tools on UNIX/RISC Workstations
F. Rouse 351

Fortran 90 Evaluation and Experience/Status Report
M. Marquina, M. Metcalf 367

Table of Contents

Page

D0 Software Library Organization and Maintenance
S. Protopopescu 373

CPS Plans and Evolution/UDP--UPS at FNAL
P. LeBrun, R. Pordes 391

Status of Graphics at CERN/GKS, X11, HIGZ, PHIGS, PEX/ The PHIGS Package Selected by CERN/Development of User Interface Systems (KUIP/MOTIF)/Real-time 3-D Viewers (with MOTIF and PHIGS/PEX)
R. Brun, C. Vandoni 405

Fermilab View of HEPLIB
I. Gaines 427

GEANT Version 3.15 and Beyond/Graphics and CAD Interface/ The New Geometry Package/Standard Interface to Event Generators (LHC/ SLUG)/The FLUKA--DTUNUC Interfaces and Other Physics/ Improvements/Parametrization
R. Brun, F. Carminati 435

The Possibilities for using the Intel iPSC Supercomputer for On-line and Off-line Analysis
P. Keener 479

SSCL Library - Mail Stop 2010
Superconducting Super Collider Laboratory
2550 Beckleyemeade Avenue, Suite 125
Dallas, Texas 75237

HEPLIB'91

International Users Meeting on the Support and Environments of High Energy Physics Computing

SSC Laboratory, Dallas, Texas
September 19-20, 1991

Dates:

By popular demand, the dates of the Meeting have been chosen to be close to the HEPiX Meeting scheduled at Fermilab on September 23-24, 1991.

Purpose:

To discuss the current and future HEP computing support and environments from the perspective of new horizons in accelerator, physics, and computing technologies.

Topics:

Topics of interest to the Meeting include (but are not limited to):

* The forming of the HEPLIB world user group for High Energy Physics computing

* Mandate, desirables, coordination, organization, funding

* User experience, international collaboration

* The roles of national labs, universities, and industry

* Range of software, Monte Carlo, Mathematics, Physics, Interactive analysis, Text Processors, Editors, Graphics, Data Base Systems, Code Management tools

* Program libraries, frequency of updates, distribution

HEPLIB'91
SSC Laboratory
Sept 19-20, 1991

* Distributed and interactive computing, data base systems, user interface

* UNIX operating systems, networking, compilers, Xlib, X-Graphics

* Documentation, updates, availability, distribution

* Code Management in large collaborations, keeping track of program versions

* Quality assurance, testing, conventions, standards

Registration:

There is a $60 registration fee, payable to HEPLIB'91 in U.S. dollars, check, or money order. Please return the enclosed registration form, along with the registration fee, to the Conference Office address shown on the registration form.

Call for Speakers:

Participants who wish to make a presentation on any of the key topics listed above should indicate a tentative title on the enclosed registration form.

Meeting Program:

There will be two full days of plenary sessions which include invited speakers and adequate time for discussions and open presentations. A detailed agenda will be sent to all registrants.

Accommodations:

Hotel information is enclosed. A small block of rooms has been reserved at the Holiday Inn in Duncanville, Texas, telephone (214) 298-8911. When calling to make reservations, be sure to identify yourself as a HEPLIB'91 participant to ensure the correct rate. Rooms will be filled on a first-come, first-served basis.

HEPLIB'91
SSC Laboratory
Sept 19-20, 1991

Further Information:

Meeting Information:

Joan Arruza, Secretary
SSC Laboratory M/S 2010
2550 Beckleymeade Avenue
Dallas, TX 75237
Tel: (214) 708-6137
Fax: (214) 708-0016
E-mail: Arruza@sscvx1.ssc.gov

Scientific Information:

Harald Johnstad, Chairman
SSC Laboratory M/S 2001
2550 Beckleymeade Avenue
Dallas, TX 75237
Tel: (214) 708-6000
Fax: (214) 708-0006
E-mail: Johnstad@sscvx1.ssc.gov

Local Organizing Committee:

P. Hale, SSC Lab Users Office
P. Kreitz, SSC Library
K. Damrau, SSC Lab
N. Danesh, SSC Lab
B. Traversat, SSC Lab

International Scientific Advisory Committee:

H. Johnstad, SSCL, Chairman
K. Amako, KEK
R. Brun, CERN
F. Carminati, CERN
S. Frederiksen, SSC Lab
W. Koellner, LBL
P. LeBrun, FNAL
M. Marquina, CERN
A. Palounek, LBL
J. Peters, DESY
L. Price, ANL
Y. Watase, KEK
S. Youssef, SCRI

HEPLIB'91 USERS MEETING
SSCL Laboratory
Building #4
Dallas, Texas
September 19-20, 1991

AGENDA

Thursday, September 19, 1991:

8:00 a.m.	Registration		
9:00 a.m.	Welcome to SSCL	P. Leibold	(SSCL)
9:10 a.m.	SSCL Computing Division	K. Rich	(SSCL)
9:20 a.m.	Opening Talk Introduction to HEPLIB	H. Johnstad	(SSCL)
9:40 a.m.	Customer View of HEPLIB In Multi-vendor Environment	P. LeBrun	(FNAL)
10:00 a.m.	HEP Code Compilation	A. Palounek	(LANL)
10:20 a.m.	CERNLIB Status and Implementation Organization and Manpower	M. Marquina	(CERN)
10:40 a.m.	COFFEE BREAK		
11:00 a.m.	SCRI Software Products Ideas for HEPLIB	S. Youssef	(SCRI)
11:20 a.m.	SDC Requirements to HEPLIB	L. Price	(ANL)
11:40 a.m.	Overview of Physics Software at KEK	K. Amako	(KEK)
12 Noon	Overview of Physics Software at DESY	J. H. Peters	(DESY)
12:20 p.m.	Overview of Physics Software at LBL	W. Koellner	(LBL)
12:40 p.m.	LUNCH		
1:40 p.m.	Overview of Physics Software at SSC	H. Johnstad	(SSCL)
2:00 p.m.	SLD Interactive Physics Analysis Initial Experience with XMotif Comments on GEM	T. Johnson	(SLAC)
2:20 p.m.	HEPLIB and Code Management Evolution of the CMZ Code Management System	R. Brun	(CERN)
2:40 p.m.	Standard HEP Particle Data Book On the Physicist Workstation	L. Garren	(FNAL)
3:00 p.m.	CERNLIB Maintenance From Patchy to CMZ Installation Servers Improving the Test Phase	M. Marquina	(CERN)

3:20 p.m.	Information Exchange CERN List Servers The ASIS World-wide CERNLIB Server The WWW Project	M. Marquina	(CERN)
3:40 p.m.	COFFEE BREAK		
4:00 p.m..	Panel Discussion: Forming of a HEPLIB User Group Manpower Organization Commitments		
6:00 p.m.	ADJOURN		
7:00 p.m.	DINNER (Milano's Restaurant, Duncanville)		

Friday, September 20, 1991:

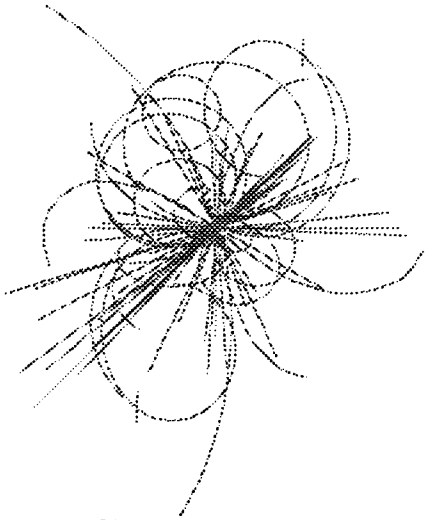
8:20 a.m.	Distributed Data Management for High Performance Data Systems IEEE Storage Standards	B. Coyne	(IBM HOUSTON)
8:40 a.m.	DAFNE Software Requirements	E. Pace	(INFN)
9:00 a.m.	CompHEP - Calculation of Feynman Diagrams	A. Kryukov	(MOSCOW U.)
9:20 a.m.	Discussion of the Need for Active Promotion of non-HEP Applications of Advances in HEP Computing	V. Chaloupka	(U. of WASH.)
9:40 a.m.	Initial Experience Porting CDF Software to RS6000	R. Hollebeek	(U. of PENN.)
10:00 a.m.	Distributed Computing at CERN The SHIFT/HOPE Expansion Experience with ISIS and Distributed GEANT The Simulation Farm Project The Visualization Farm Proposal	R. Brun	(CERN)
10:20 a.m.	The SDC SHELL Project	S. Kunori	(U. of MD.)
10:40 a.m.	COFFEE BREAK		
11:00 a.m.	Parallelization software on PDSF The PSHELL User Interface	B. Traversat	(SSCL)
11:20 a.m.	A Parallel Version of GEANT	P. Rulon	(SCRI)
11:40 a.m.	CALOR on PDSF and Intel	T. Gabriel T. Handler B. Moore	(ORNL) (U. of TENN.) (U. of MISS.)
12 Noon	Physics Analysis Tools on UNIX RISC Workstations	F. Rouse	(UC-DAVIS)
12:20 p.m.	Fortran 90 Evaluation and Experience Status Report	M. Metcalf M. Marquina	(CERN) (CERN)

12:40 p.m.	LUNCH		
1:40 p.m.	D0 Software Library Organization and Maintenance	S. Protopopescu	(BNL)
2:00 p.m.	CPS Plans and Evolution UDP/UPS at FNAL	P. LeBrun	(FNAL)
2:20 p.m.	Status of Graphics at CERN GKS, X11, HIGZ, PHIGS, PEX The PHIGS Package Selected by CERN Development of User Interface Systems (KUIP/MOTIF) Real-time 3-D Viewers (with MOTIF and PHIGS/PEX)	R. Brun	(CERN)
2:40 p.m.	Fermilab View of HEPLIB	I. Gaines	(FNAL)
3:00 p.m.	Continued Discussion on HEPLIB Proposal Organization Future Meetings		
4:00 p.m.	COFFEE BREAK		
4:20 p.m.	GEANT Version 3.15 and Beyond	R. Brun	(CERN)
	Graphics and CAD Interface	F. Carminati	(CERN)
	The New Geometry Package Standard Interface to Event Generators (LHC/SLUG) The FLUKA/DTUNUC Interfaces and Other Physics Improvements Parametrization		
4:40 p.m.	The Evolution of ZEBRA in a World with F77, F90, C, C++, etc. Adaptation of FZ and RZ to a Mutli-language Environment The Data Dictionary The Data Base Complex for a Future Experiment How to Upgrade Existing Tools (RZ, DBL3, OPCAL, FATMEN. . .) with Industry Standards (IEEE Reference Model, DCE . . .)	R. Brun	(CERN)
5:00 p.m.	The possibilities for using the Intel iPSC supercomputer for on-line and off-line analysis	P. Keener	(U. of PENN.)
5:20 p.m.	Conclusions	Speaker TBA	
6:00 p.m.	ADJOURN		

HEPLIB'91

LIST OF PARTICIPANTS

Katsuya Amaka	KEK National Laboratory for High Energy Physics
James M. Bishop	University of Notre Dame
Rene Brun	CERN
Vladimir Chaloupka	University of Washington
Larry Cormell	Superconducting Super Collider Laboratory
Bill Courtney	International Business Machines
Robert Coyne	International Business Machines
Kenneth Damrau	Superconducting Super Collider Laboratory
Nasser Danesh	Superconducting Super Collider Laboratory
Reinhard Eisberg	Deutsches Elektronen-Synchrotron
Anthony Elam	International Business Machines
Soren Frederiksen	Superconducting Super Collider Laboratory
Alan Fry	Superconducting Super Collider Laboratory
Tony Gabriel	Oak Ridge National Laboratory
Irwin Gaines	Fermi National Accelerator Laboratory
Thomas Handler	University of Tennessee
Steve Hicks	International Business Machines
Robert Hollebeek	University of Pennsylvania
Roger Howard	University of British Columbia
Tony Johnson	Stanford Linear Accelerator Center
Harald Johnstad	Superconducting Super Collider
Paul Keener	University of Pennsylvania
Werner Koellner	Lawrence Berkeley Laboratory
Thomas Kozlowski	Los Alamos National Laboratory
Shuichi Kunori	University of Maryland
Paul LeBrun	Fermi National Accelerator Laboratory
Phil Leibold	Superconducting Super Collider Laboratory
Manlio Marquez	Superconducting Super Collider Laboratory
Miguel Marquina	CERN
Charles Martin	Stanford Linear Accelerator Center
Edward May	Argonne National Laboratory
Brent Moore	University of Mississippi
Youhei Morita	KEK National Laboratory of High Energy Physics
Joseph Orgeron	University of Texas at Arlington
Elisabetta Pace	Istituto Nazionale Di Fisica Nucleare
Andrea Palounek	Los Alamos National Laboratory
Jan Hendrik Peters	Deutsches Elektronen-Synchrotron
Lawrence Price	Argonne National Laboratory
Serban Protopopescu	Brookhaven National Laboratory
Lee Roberts	Superconducting Super Collider Laboratory
Byron Roe	Superconducting Super Collider Laboratory
Forest Rouse	University of California, Davis
Phillip Rulon	Supercomputer Computations Research Institute
Brian Scipioni	Superconducting Super Collider Laboratory
Frank Shively	ASA International
Thomas Throwe	Brookhaven National Laboratory
Bernard Traversat	Superconducting Super Collider Laboratory
Willem Van Leeuwen	National Institute for Nuclear Physics and HEP
Jim West	International Business Machines
David Whitehouse	Los Alamos National Laboratory
John Womersley	Florida State University
George Yost	Superconducting Super Collider Laboratory
Saul Youssef	Florida State University



Summary and Conclusions

of

The HEPLIB'91 International Users Meeting on the Support and Environments of High Energy Physics Computing

SSC Laboratory, Dallas, Texas
September 19-20, 1991

Harald Johnstad
Physics Research Division
Superconducting Super Collider Laboratory*
2550 Beckleymeade Avenue
Dallas, Texas 75237

*Operated by the Universities Research Association Inc., for the U.S. Department of Energy under Contract No. DE-AC35-89ER40486

Introduction

The following is a summary of agreements, conclusions, and consensus which were reached at the first HEPLIB Users Meeting (HEPLIB'91), held at the SSC Laboratory, Dallas, Texas, September 19-20, 1991. Fifty-four scientists from thirty-one High Energy Physics research institutes and universities met for two full days to discuss the support and environments of High Energy Physics computing and to form and define the scope of a HEPLIB Users Group.

In the following, the word *member* is sometimes used as a substitute for the wording *Member of the HEPLIB Users Group*. By *steering committee* (or *coordinating committee*) is meant *The HEPLIB Users Group Steering Committee.*

Disclaimer

This summary has been written and edited by the chairman of the HEPLIB'91 meeting, and has not been reviewed by the organizing committee. It represents a best effort to summarize common viewpoints presented and concluded at the meeting, but does not necessarily reflect all alternative viewpoints expressed at the meeting An official conclusion of the meeting will be published by the newly formed HEPLIB steering committee and distributed to the HEPLIB Users Group at large.

HEPLIB Users Group

The HEPLIB Users Group was officially formed at this meeting.

Objectives

There was a general consensus for the following objectives:

HEPLIB is a world users group for enhancements, communications, and distribution of software in the HEP computing environments.

HEPLIB collects, maintains, documents, and distributes shareable application software for HEP computing as well as non-HEP applications, including code management systems in heterogeneous environments, data base systems, and automatic installation and test procedures.

HEPLIB promotes and recommends industry, as well as HEP computing standards as appropriate, including operating systems, distributed computing environments, quality assurance, version control, and information exchange.

There was a strong consensus that the following *are not* the objectives of HEPLIB:

It should not be a software clearing house.

It should not be an umbrella type organization in the sense that the umbrella is a group of people remote from the actual user community.

It should not in any way be a competitive organization with similar or related user groups, such as the newly formed HEPiX user group for the UNIX environments in HEP computing. On the contrary, HEPLIB should work in close collaboration with, and also participate in the works of such groups.

HEPLIB shall not in any way diminish the efforts of other groups or laboratories providing support for HEP software, such as the CERN Program Library, but shall work constructively and in close collaboration with such efforts and shall strive to minimize duplication efforts, enhance quality assurance, documentation, version control, effective code maintenance, etc.

There were opposing viewpoints on the issue of frequency of releases from a HEPLIB library organization, and whether HEPLIB should contain fully packaged software products or rather consist of independent software products to be released at individual frequencies as needed. This issue should be worked out in a subgroup and a proposal should be prepared.

There was general agreement that a software product may well be developed at one location by individuals or a group, maintained, tested, etc. at another location, and distributed from a HEPLIB distribution center. An appropriate test product would be the family of the Lund Monte Carlo programs, where an initial three-way collaboration between CERN, SSCL, and the authors have already been discussed. Other physics generator programs should be included in this effort as well. This should be a task for the Physics Generators subgroup (Group A).

Organization

Bylaws were not worked out at this meeting. It was agreed that the steering committee will provide a proposal for bylaws at the next HEPLIB Users Meeting.

It was generally agreed that the HEPLIB Users Group is a fully democratic organization and that the active users of the HEPLIB environments are themselves the governing body of the organization.

Subgroups should be formed and there should be an all-to-all communication between subgroups and the HEPLIB Users Group at large.

Steering Committee

The members listed below agreed to serve on the first HEPLIB steering committee (or coordination committee). The steering committee will initially address the forming and coordination of subgroups, initiate information exchange and communication, plan logistics, arrange for meetings, begin planning for a HEPLIB Newsletter, and look into questions of manpower and funding for the HEPLIB Users Group.

Katsuya Amako	KEK
Rene Brun	CERN
Tony Gabriel	ORNL
Irwin Gaines	FNAL
Thomas Handler	U of Tennessee
Bob Hollebeek	U of Pennsylvania
Tony Johnson	SLAC
Harald Johnstad (chairman)	SSCL
Miguel Marquina	CERN
Elisabetta Pace	INFN
Andrea Palounek	LANL
Jan Hendrik Peters	DESY
Saul Youssef	SCRI

Subgroups

The following subgroups were proposed. Other groups may be formed later. The names for each subgroup may change later. With one exception, no names were assigned to subgroups at the meeting, but members were encouraged to sign up for one or more of these subgroups by contacting other members and the steering committee.

Preliminary list of subgroups (in no particular order):

A. Physics Generators (H. Johnstad and others)
B. Physics Analysis
C. Detector Simulation
D. Mathematics
E. Code Management
F. Standards Alert
G. Operating Systems
H. Tools and Utilities
I. Documentation Systems
J. Graphics Applications
K. Distributed Computing
L. Non-HEP Applications
M. Vendor Alert

Information Exchange

There was a consensus at the meeting that effective, continuous and frequent sharing of all information on all activities within the group would be crucial for the success of the HEPLIB activities. Several media has been suggested for communication between members:

(i) *The Bitnet LISTSERV facility:* Main media for all information exchange.

The HEPMGR@CERNVM List will be co-owned by the CERN Program Library (M.Marquina) and by the SSCL Software Applications Group (H.Johnstad).

There will continue to be a HEPLIB@CERNVM List, monitored by M.Marquina at CERN dealing with the CERN Program Library in particular.

There will be a new HEPLIB@UTDALLAS List, monitored by H.Johnstad at SSCL, dealing with information pertinent to the HEPLIB User Group at large.

(ii) *HEPLIB Computing Newsletter:* Activity reports, Meetings, Vox Populi, etc.

(iii) *CERN Computing Newsletter:* Summary of HEPLIB activities.

(iv) *SSCL Computing Newsletter:* Summary of HEPLIB activities.

(v) *CERN Courier:* Summary reports on HEPLIB meetings and major decisions.

Addresses

Initially these addresses may be used to contact the HEPLIB User Group in Europe and the United States:

Europe:

Miguel Marquina
CN Division
CERN
CH-1211 Geneve 23
Switzerland

Tel (022) 767-4912
Fax (022) 767-6555

Marquina@CERNVM

United States:

Harald Johnstad
SSC Laboratory MS-2001
2550 Beckleyemeade Avenue
Dallas, Texas 75237
U.S.A.

Tel (214) 708-6000
Fax (214) 708-0006

Johnstad@SSCVX1.SSC.GOV

HEPLIB Compilation

The first compilation of HEP software was done jointly by Andrea Palounek, formerly LBL, now at LANL, and by Saul Youssef at SCRI, both members of the HEPLIB steering committee. Their first report is available from LBL, report number LBL-29115.

This report is also available as a PostScript file (DOCREF.PS) or a Latex input file (DOCREF.TEX) on SSCVX1 at SSCL in the following user area:

SSCVX1::USER1:[APTP]

Next Meeting

It was agreed that the HEPLIB User Group should meet at least annually. Initially, there should be more frequent meetings to coordinate and initiate the various efforts, subgroups, etc. It was generally agreed that a second meeting should be held early in 1992 in Europe in order to involve a broader participation from the European HEP community as early as possible. Tentatively, the next HEPLIB Users Meeting will be held in January 1992 in close connection with the HEP software meeting to be held in L'Agelonde, France, January 13-18, 1992. Further details and logistics will be looked into by the participants from CERN present at this first HEPLIB meeting (R.Brun and M.Marquina).

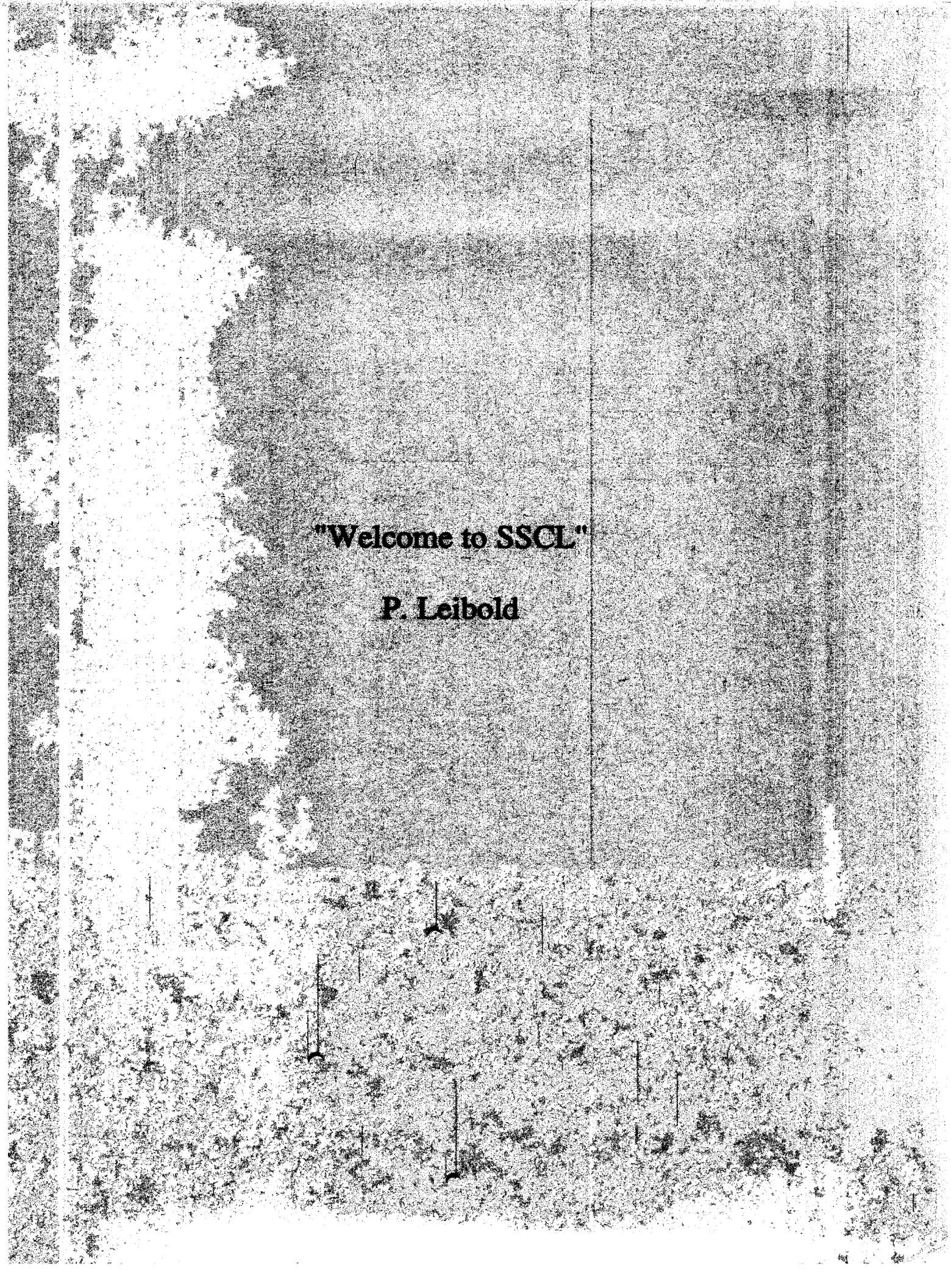
Conclusions

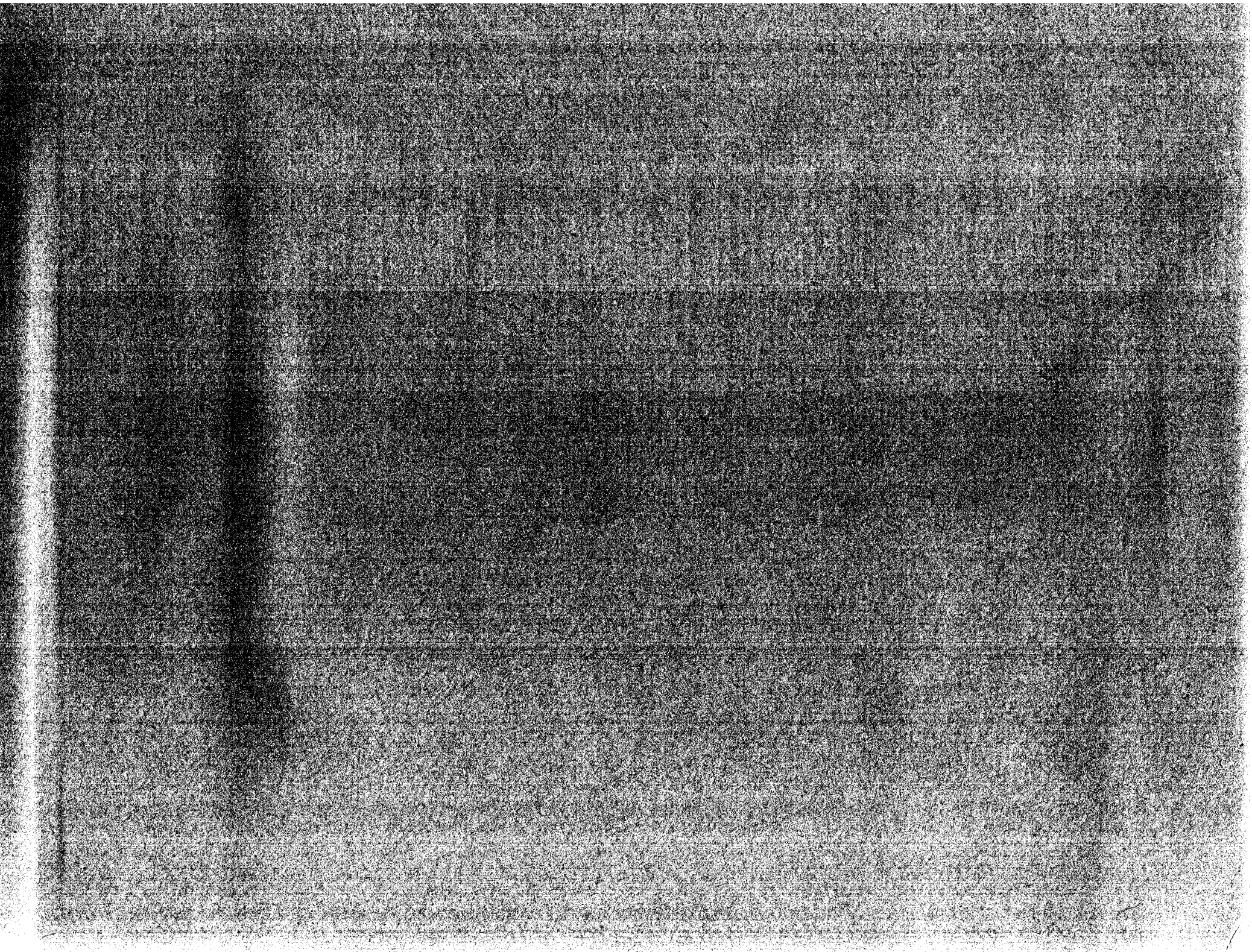
The first HEPLIB User Group Meeting was a productive meeting with many constructive discussions on a broad range of issues on the support and environments in HEP computing and beyond. The goals set for this first meeting were reached, and many important decisions were made.

The actual scope of the HEPLIB activities, including details on software products, thoughts on pressure groups, liaison functions, vendor contact, representation at standardization committees, etc. were discussed from many viewpoints.. Further details on these issues will be presented as actual proposals.

Agreements on initial information exchange and media were reached. A preliminary organization and steering committee was established. It was agreed to have a second HEPLIB Users Meeting in Europe, tentatively in January 1992.

Manpower and funding for the HEPLIB activities are major unresolved issues which needs to be looked at realistically in order for HEPLIB to become a successful organization or institution.


"Welcome to SSCL"
P. Leibold



INTRODUCTION - P.O. LIEBOLD

HEPLiB '91

Welcome to the SSCL and HEPLiB '91 users meeting. Such an excellent turnout and representation from every major High Energy Physics Research Institute and Lab indicates the importance of this meeting, and specifically the need to address the issues of physics software development, maintenance and distribution.

The formation of a users group will provide the mechanism for not only exchanging ideas and experiences, but hopefully will provide an emphasis or focus on the problems confronting the High Energy Physics Community, not only in maintaining physics codes but in coordinating a standard philosophy for future software development.

SSCL Physics has just completed the implementation of the Physics Detector Simulation Facility which provides over 1000 Vax equivalent computing resources for physicists to access for event simulation during detector design. This resource represents the effective use of RISC architectures and the ability to develop large computing resources by adhering to industry standards when specifying hardware and software. It emphasizes the ability to use 90's technologies and develop in multi-vendor environment providing great price/performance advantages.

The PDSF, of course, uses the CERN Libraries as well as other major physics applications. Fermi Labs' CPS was a major component to provide a mechanism for supporting parallel event simulation making effective use of the loosely coupled parallel architecture of

the PDSF. Porting of user codes and CERN libraries, to our RISC/UNIX environment, became a significant effort to support and maintain. However, now that PDSF is operational, we can concentrate our efforts on maintaining a production environment on that architecture.

Software development for PDSF was minimal considering the magnitude of the resource. The major efforts concentrated on development of Workstation Assignment Application (WASH), Data Management Systems, (DMS) as well as integration of various systems level codes to provide network services and linkages between major systems components. The use of structured software engineering methodologies in the design of PDSF software was a major contributing factor to the projects success and ability for the software development group to deliver systems on-time and on-schedule.

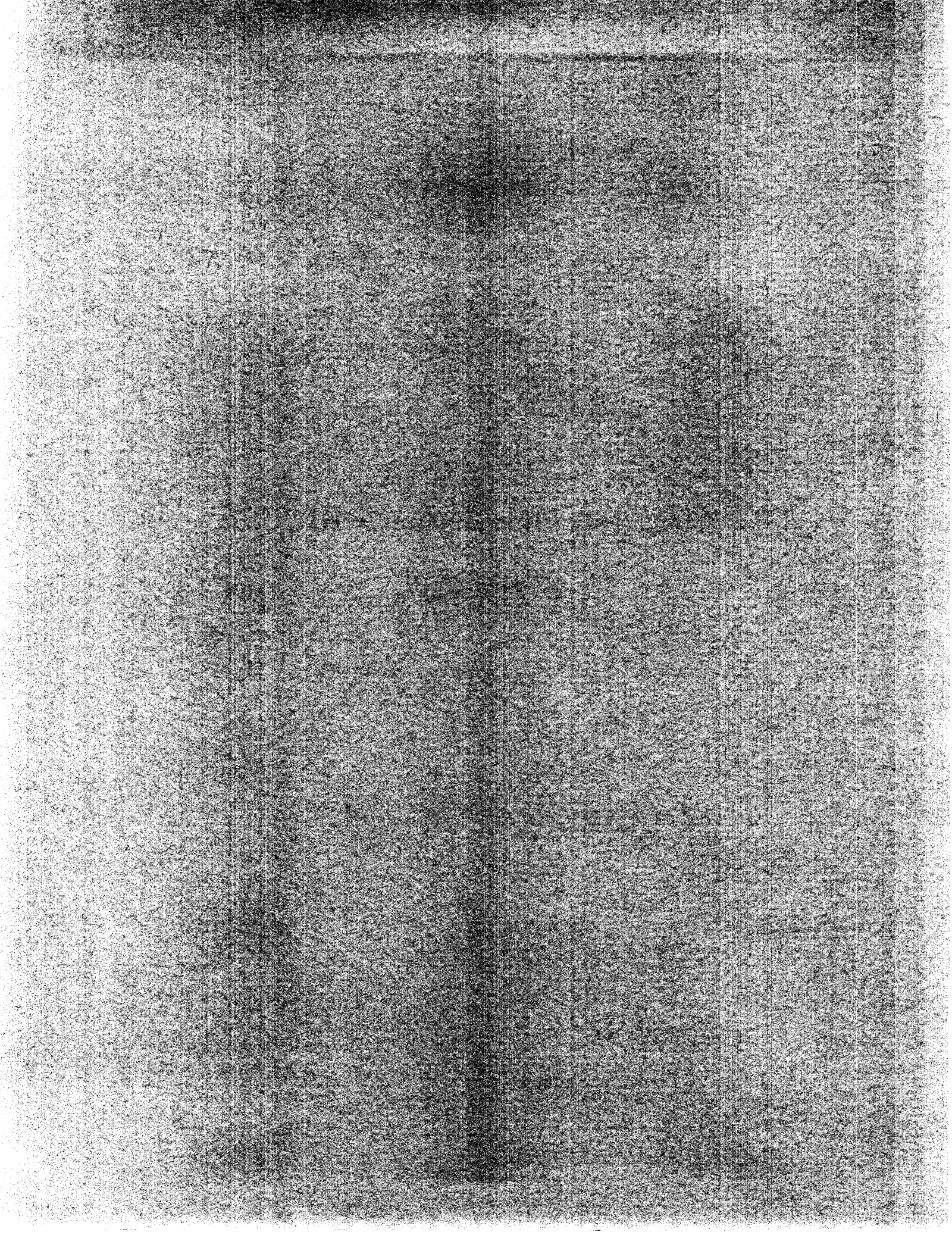
Software engineering and code development in a structured environment are issues that the High Energy Physics Community must address before we embark on future software development projects at the SSCL or for any HEP research project. In order for that to happen, there must be an understanding of the issues and a forum to establish the necessary standards. The maintenance and porting of codes must be made easier. That is the challenge that you must address.

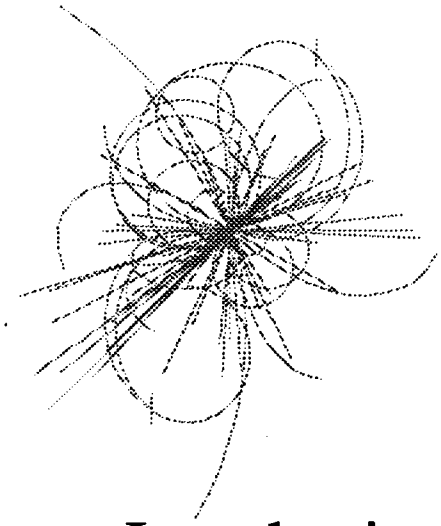
Again, welcome to the SSCL and HEPLiB!

Have a productive 2 days, Thank you.

"Opening Talk - Introduction to HEPLIB"

H. Johnstad





Introduction to HEPLIB

Opening talk presented at

The HEPLIB'91 International Users Meeting on the Support and Environments of High Energy Physics Computing

SSC Laboratory, Dallas, Texas
September 19-20, 1991

Harald Johnstad
Physics Research Division
Superconducting Super Collider Laboratory*
2550 Beckleymeade Avenue
Dallas, Texas 75237

*Operated by the Universities Research Association Inc., for the U.S. Department of Energy under Contract No. DE-AC35-89ER40486

Introduction to HEPLIB

Harald Johnstad

1 DEFINITIONS

SO FAR .....

> A dream for a foundation of organized high energy physics software.

> An informal forum or user group (peers) whose members are actively involved in the development, maintenance, and distribution of high energy physics software.

TO BECOME .....

> A formal and organized user group for the promotion community at large.

> A maintained world library for high energey physics software:

- HEP institutions will subscribe to HEPLIB
- Members will submit code to HEPLIB
- Developers and maintainers are clearly defined

- Quality Assurance and test procedures are enforced
- Availability and distribution methods are identified
- References and available documentation are identified
- Installation and test procedures are provided
- other...

> World wide distributed lists of information exchange with defined center(s) for support and distribution of HEP software, including documentation, installation, and test procedures.

2 WHAT IS HEPLIB CURRENTLY ?

- An initiative of "concerned scientists" thinking about how to organize high energy physics software world wide, with special emphasis on physics at higher energies and higher luminosities at new accelerators being planned

- Miquel's CERN Program Library LISTSERV discussion list at CERNVM

- A (newly established) Bitnet LISTSERV discussion list at UTDALLAS for the distribution of high energy physics sofwtare from the SSC Laboratory

3 HISTORY. HOW DID WE GET HERE ?

> CHEP'87 Asilomar:

Consensus for the need of:

- Shareable software
- Standards
- Code management tools
- Coordination of efforts
- Distribution of software
- Support from large facilities

> CHEP'89 Oxford:

Paul Kunz, SLAC:

"Today, in high energy physics, software is generally in a mess".

"As each new large detector comes online, the management of the software effort seems to be getting more difficult. This seems to lead to a conclusion that for the SSC we will have a major software problem."

Conclusion:

"We need to understand the real causes of our current problems before we can find the real solutions."

> Erice Workshop on Data Structures, November 1990:

- First proposal for HEPLIB.
- First committee: R.Brun (CERN), F.Carminati (CERN), O.DiRosa (CERN), H.Johnstad (SSCL), K.Leininger (FNAL), J.Peters (DESY), S.Protopopescu (BNL), F.Rademakers (CERN), J.Shiers (CERN), G.Word (SLAC), G.Xexeo (CERN), I.Zacharov (CERN), J.Zoll (CERN), W.van Leeuwen (NIKHEF)
- Conclusion: Call for a world user meeting on the HEPLIB idea.

> CHEP'91 Tsukuba:

- Panel Discussion on HEPLIB
- Announcement of First HEPLIB User Meeting

> First HEPLIB User Meeting, SSCL September 1991:

..... here we are

4 PURPOSE OF THE HEPLIB'91 MEETING

- Formal organization of HEPLIB
- Review current situation at major HEP institutions in general, as well as specific on simulation tools, code management, data bases, graphics, etc.
- Discuss requirements from large collaborations, at Fermilab, SLAC, DESY, SSC proposals, early LHC collaboartions, etc.
- Discuss strategy to find common grounds and organization of the contents of high energy physics software, code development, maintenance, and distribution
- Discuss commitments for manpower and local organization at various institutions for support, documentation, distribution, etc.

5 FORMAT OF THE MEETING

- A two-day meeting, one evening for contacts and discussions at dinner
- Institutions represented
- Agenda
- Scope: Focus on general idea and purpose
- Leave room for discussions after talks
- Social event: Dinner Thursday evening

6 MINIMAL GOALS

- Establish HEPLIB
- Understanding of what people are expecting from HEPLIB
- Draft organization
- Form kernel committee
- Agreement on announcement of HEPLIB to the HEP community
- Draft framework for initial implementation of software products
- Not to many details on requirements at this meeting
- Agree on initial test-bed candidates of various complexity for sumbission to HEPLIB
- Agreement on where to store world wide discussion lists (LISTSERV or other)
- Establish time schedule and milestones
- Estimate of manpower needed for HEPLIB to succeed
- Decide on next meeting

7 LOGISITICS

> Meeting Secretary: Joan Arruza, SSC Library

> Other helpful people: SSC Library staff

> Local Coordination: Phyllis Hale, SSCL Users Office

> Computer terminals: Users area (Building 4)

> Copy of transparencies: All speakers have been given colored folder with their name on it, one folder for each talk. Put your transparencies in the colored folder and leave for copying. The secretaries will make a master copy and then return the transparancies to you in the same folder.

> A compendium of all talks will be mailed after the meeting.

> Dinner is at Mialno's restaurant in Duncanaville on Camp Wisdom Rd.

8 FIRST HEPIX MEETING

HEPiX is a different user group being formed to focus more specifically on the sharing of experiences in the use of UNIX in the HEP computing environments. The first HEPiX Meeting will be held at Fermilab, September 23-25, 1991. An agenda is available.

International Scientific Advisory Committee:

H. Johnstad, SSCL, Chairman

K. Amako, KEK

R. Brun, CERN

F. Carminati, CERN

S. Frederiksen, SSCL

W. Koellner, LBL

P. LeBrun, FNAL

M. Marquina, CERN

A. Palounek, LBL

J. Peters, DESY

L. Price, ANL

Y. Watase, KEK

S. Youssef, SCRI

List of Represented National and International Research Labs

1. Argonne National Laboratory (ANL)
2. Brookhaven National Laboratory (BNL)
3. CERN
4. Deutsches Elektronen-Synchrotron (DESY)
5. Fermi National Accelerator Laboratory (FNAL)
6. IBM Federal Sector Division
7. Lawrence Berekely Laboratory (LBL)
8. Los Alamos National Laboratory (LANL)
9. National Institute for Nuclear Physics and HEP (NIKHEF)
10. National Laboratory for High Energy Physics (KEK)
11. Oak Ridge National Laboratory (ORNL)
12. Stanford Linear Accelerator (SLAC)
13. Supercomputer Computations Research Institute (SCRI)
14. Superconducting Super Collider Laboratory (SSCL)

List of Represented Universities

1. Florida State University
2. Louisiana State University
3. Moscow State University
4. Pennsylvania State University
5. University of British Columbia
6. University of California Davis
7. University of Hawaii
8. University of Maryland
9. University of Michigan
10. University of Notre Dame
11. University of Pennsylvania
12. University of Tennessee
13. University of Texas Arlington
14. University of Texas Dallas
15. University of Washington
16. Yale University
17. York University

HEPLIB'91

Comments to the Meeting from Yoshiyuki Watase, KEK

September 18, 1991

Dear Harald

I would like to make a few comments for the discussion at the HEPLIB meeting.

Everybody agrees that the undertaking at CERN for CERN Library is a great success. Since their efforts shall be highly appreciated for the future, too, we should hear their opinion and suggestions for the HEPLIB activities. But, people think a collaboration for HEP library is necessary and not avoidable due to the world wide collaboration in HEP experiments and the other various activities in HEP.

For making the meeting fruitful, we should discuss and come to agreement in the following subjects.

1. Definision of HEPLIB What is HEPLIB ? Which program is in HEPLIB ?

2. Key laboratories for HEPLIB distribution and maintenance.

3. Set up executive committee to coordinate the HEPLIB activities.

4. The resultant agreements should be recognized in HEP community and the activities must be authorized by each laboratory organization.

Have good meeting ! Regards,

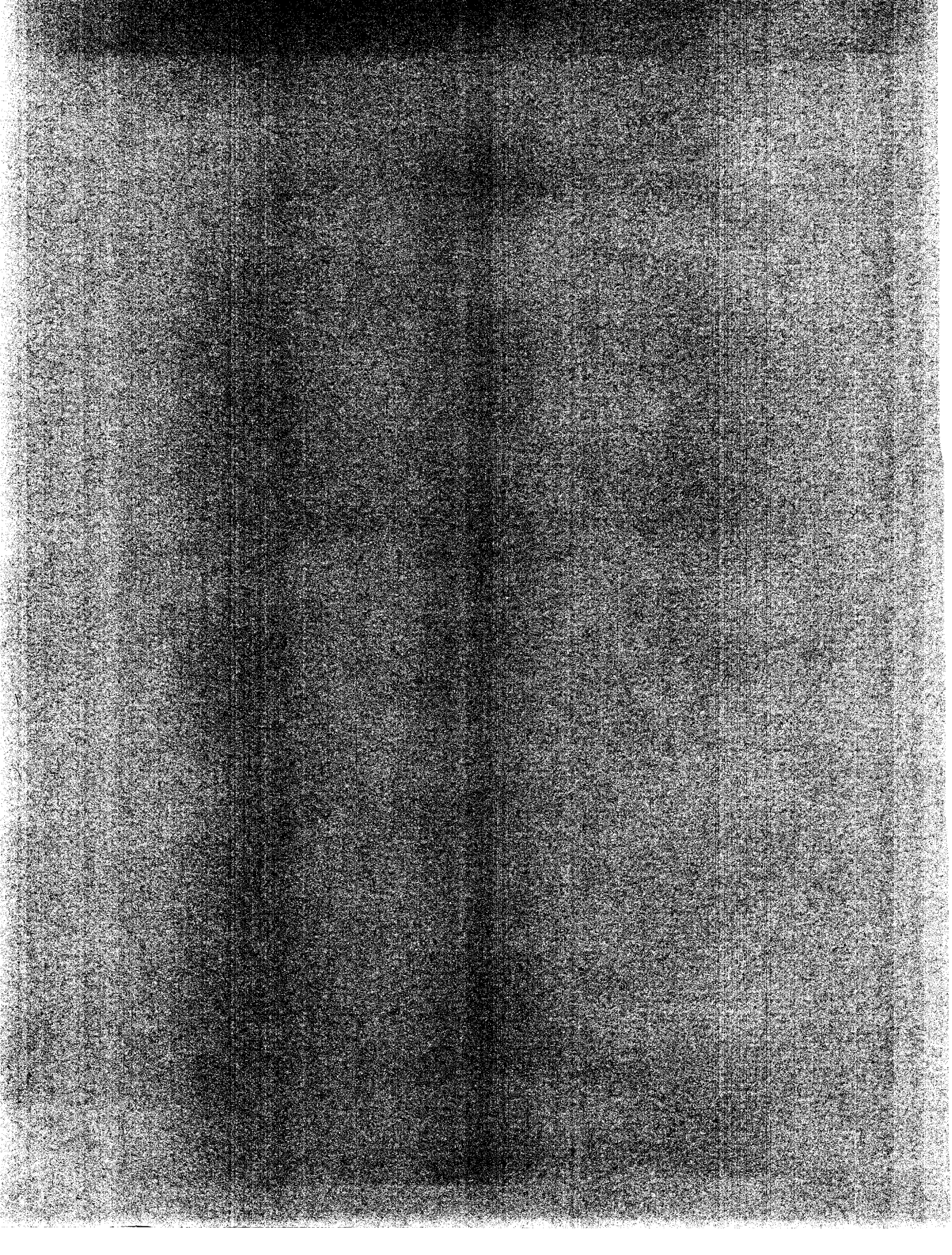
Yoshiyuki Watase, KEK

Inquiry Summary, Asilomar CHEP'87

QUESTION	YES	NO	OTHER
Is shareable software useful to the HEP community	98%	2%	0%
Is it preferable to write HEP software in Fortran	48%	30%	22%
Should we have rules on how to write Fortran programs?	80%	11%	9%
Do we need source managmenet tools?	93%	2%	5%
Are you using any at the moment?	52%	46%	2%
Do we need tools which would interface to both a local editor and a source code managment system?	63%	18%	18%
Do we want to coordinate our efforts in HEP software?	93%	2%	5%
Should HEP Labs (CERN, SLAC, FNAL,..) support the HEP community with software?	93%	2%	5%
Should these facilities distribute software?	96%	0%	4%

"Customer View of HEPLIB in a Multi-vendor Environment"

P. LeBrun



A Customer View of HEPLIB

in a multi vendor environment.

P. Lebrun

Physics Analysis Tool Group.

Fermi National Accelerator Laboratory
Batavia, Illinois 60510, USA

Outline

- Foreword
- Some User's Remarks
- User communities : Pluralism in HEP
- Attempting to understand conflicting User requirements...
- Conclusions (if i can...)

Foreword: Am i really a User ?

Member of "one of those, from the Computing Outfit.."

But i also did change a few gas bottles at E-687....

A "User" is :

- Not a Computing Expert...
- Affiliated with an HEP Experiment ✓.. or Theory group. ?
- Might become a developer, if...
  - Is not happy with what he has, and/or
  - Has no money...

A "grass root" reaction to HEPLIB

Oh! You mean CERNLIB stuff... !

Oh! I don't know ... I'll tell you what the user needs : Data ! !

Yeh, i use Randon number generator, occasionally.. !

HBOOK3 was fine, why did you changed it ??
(old!)

We must have the most recent version of CERN as soon as it is available.. CDF can't run without it !

Not so easy to understand what they want !

Pluralism

Styles & Trends...

- Commercial versus home grown Funds for software : Very limited for Universities, Existing for National Labs.

- Complete systems vs. specific tools

- HEP : The last frontier Let me do my own thing Or should be "serious professionals"



- User response to "needs" and new ideas Unpredictable !!

  (ACP or CPS were not "asked for")

- **Is commercial software relevant ?**

  - To HEP in general ? Very few tools are HEP specific. For instance, Code Management has little to do with HEP. Data Acquisition Software ? Histogramming.. HEP Monte-Carlo are truly HEP...
    (WAS NOT TRUE 20 10 years ago)

  - To HEPLIB ? How to we maintain and distribute "semi-commercial" tools ? AVS library, LAb-View Virtual Instruments ?

    CERN
    FNAL DO DISTRIBUTE COMMERCIAL SOFTWARE.

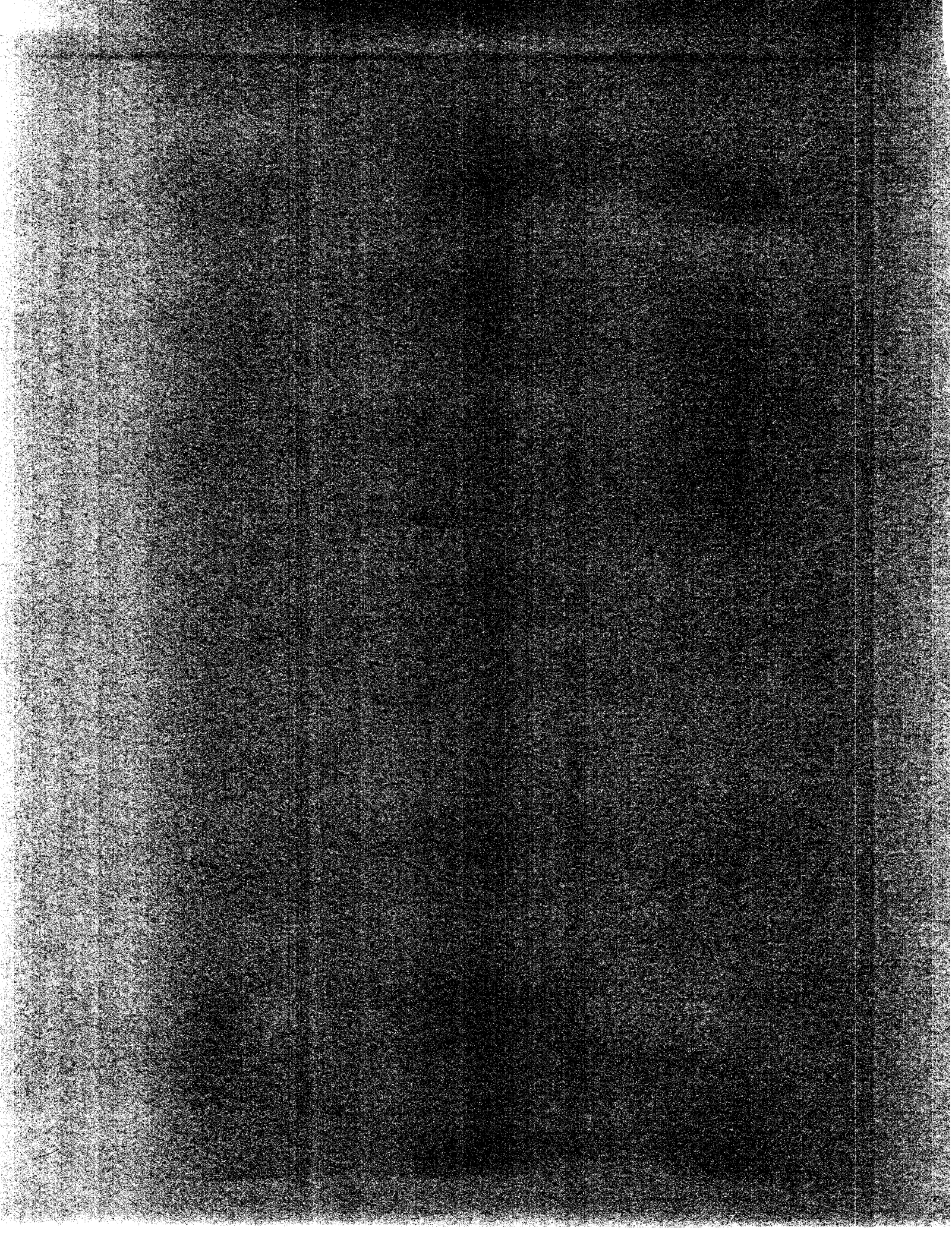
Conclusion

- User requirements are hard to get !
- They do have (and want to keep) multiple sources for their software..
  - Home-grown
  - Collaboration (SDC, GEM, E687..)
  - HEPLIB
  - Commercial Vendor
- No matter what.. they want it cheap !

Integration of HEPLIB will be hard !

"HEP Code Compilation"

A. Palounek



- many different versions of event generators:

  e^+e^-

  ep

  pp, $\bar{p}p$

  hadron-nucleus, nucleus-nucleus

  jet evolution

  special cases: J production
  Bhabhas
  ⋮

  heavy ions

  exact calculations

  many kinds of approximations

  phase space

- electromagnetic
- hadronic interactions

  different energy ranges

  special materials

    (organic scintillators,

    heavy absorbers,

    ⋮

  special neutron codes

- Frameworks

  generic

  detector-specific

- Data Managers
  - frequently experiment-specific
  - generic
  - based on C
- Analysis helpers
  - generic
  - frequently no overlap
- Other helpers
  - FORTRAN translators
  - code "management
  - spires
  - who

- Miscellaneous

  SPICE

  neural net

  parallel

Conclusion:

We are STRONG on PHYSICS

WEAK on COMPUTER SCIENCE.

but frustrated

THAT IS WHERE WE NEED CAREFUL THOUGHT.

Hints from

CERNLIB

NESC

Remember the USER!

She wants to make useful histograms NOW, without worrying about the logistics. And for free.

to get LBL-29115
aka "Compilation"

- copy
  SSCVX1::USER1:[APTP]
  DOCREF.TEX
  DOCREF.PS

- write
  CSA::APTP
  APTP@LBL
  aptp@lanlsscØ.lanl.gov

- ask secreteriat

LBL-29115
(rev. September 19, 1991)

Monte Carlo Programs and Other Utilities for High Energy Physics †

Andrea P. T. Palounek

Physics Division
Lawrence Berkeley Laboratory
1 Cyclotron Road, Berkeley, CA 94720
current address:
aptp@lanlssc0.lanl.gov or 1226::APTP
Los Alamos National Laboratory
Los Alamos, NM 87545

Saul Youssef

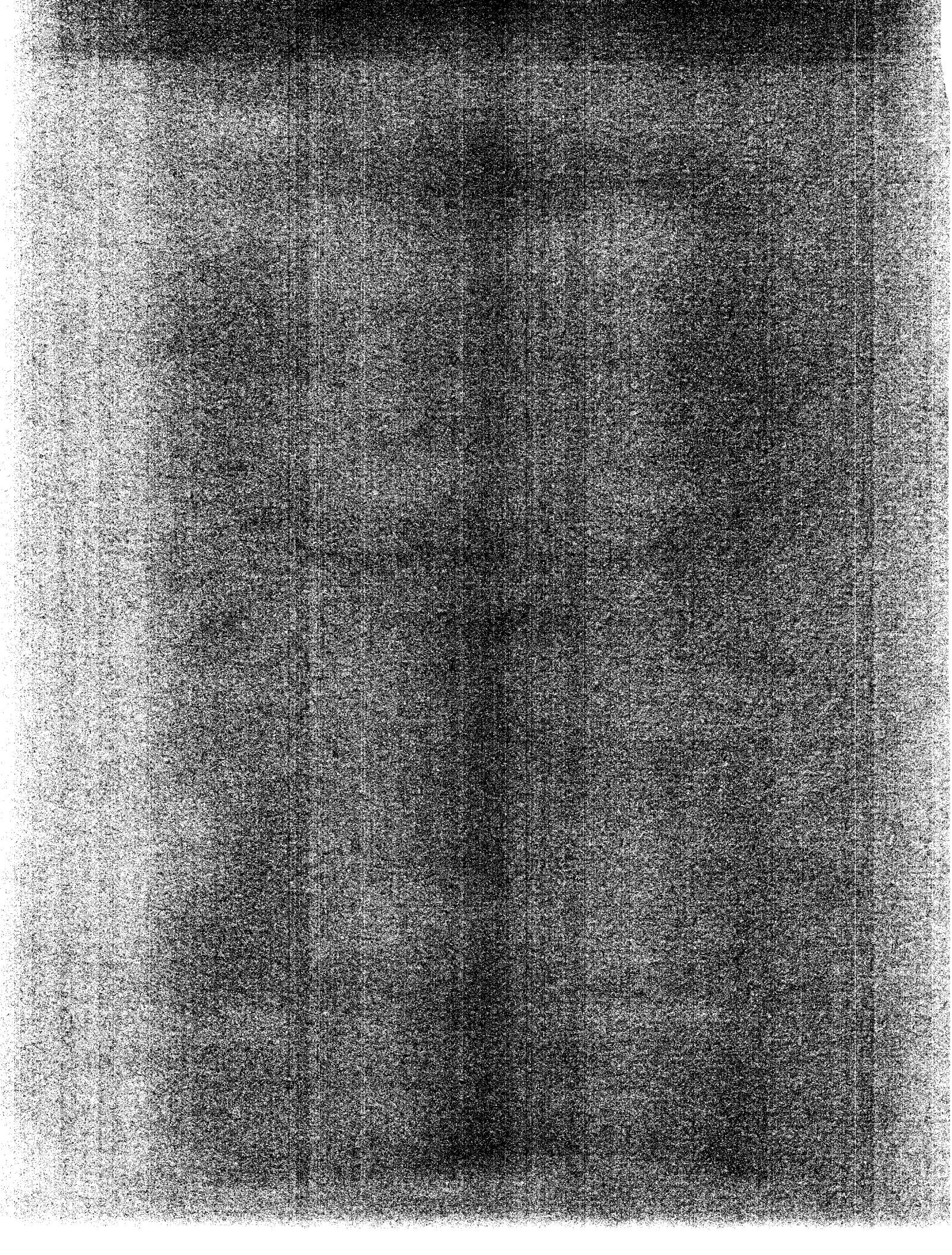
Supercomputer Computations Research Institute
Florida State University
Tallahassee, Florida 32306-4052

May 1990
revised September 19, 1991

† This work was supported by the Director, Office of Energy Research, Office of High Energy and Nuclear Physics, Division of High Energy Physics, of the U. S. Department of Energy under Contracts No. DE-AC03-76SF00098 and No. DE-FC05-85ER250000.

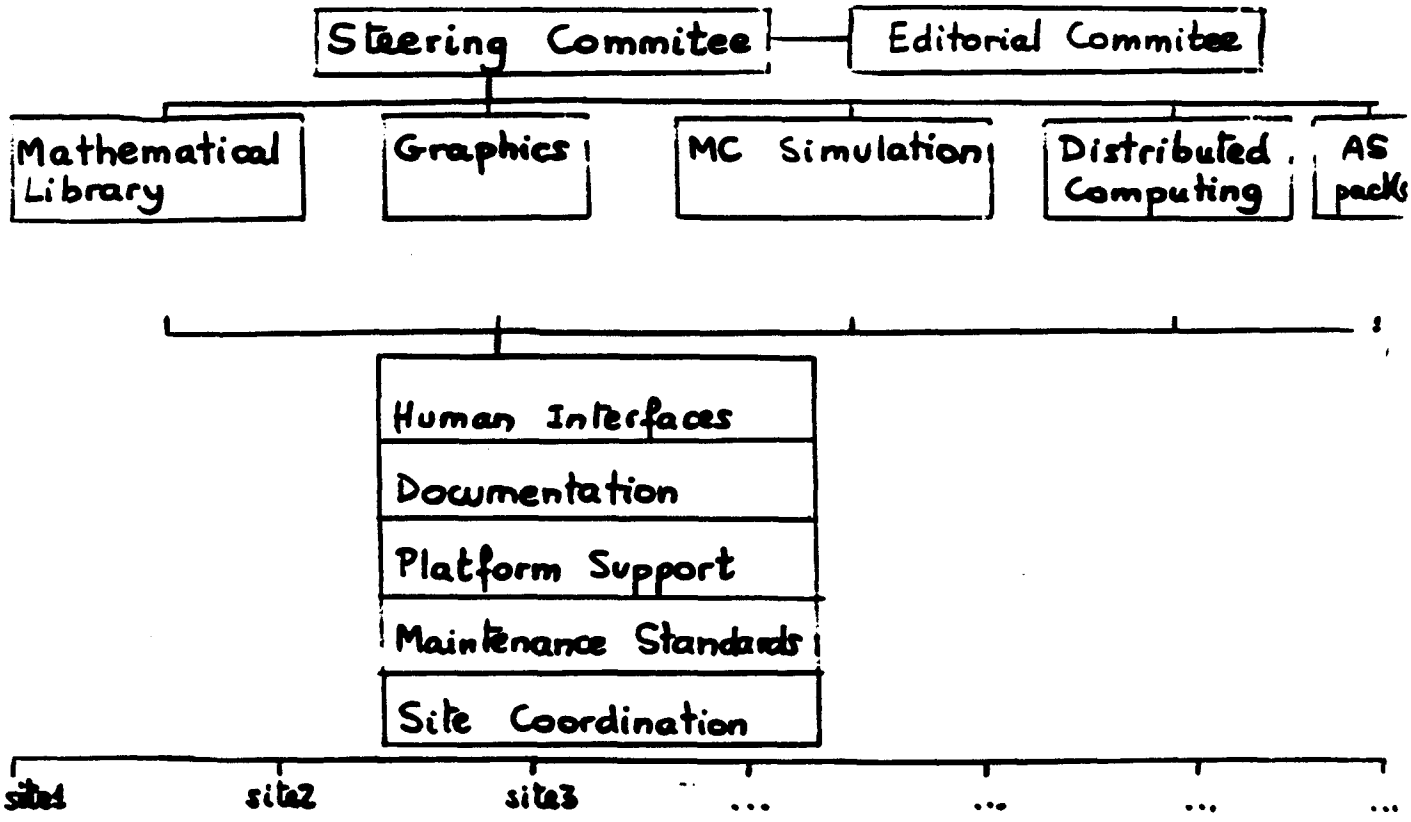
"CERNLIB Status and Implementation"
"Organization and Manpower"

M. Marquina



Proposal For HEPLIB

STRUCTURE



MATHLIB
LAPACK

HPLOT, HIGZ
PAW, PHIGS

GEANT, JETSET
PDFLIB

CSPACK, ZFTP
DBL3

MINU-
HBOOI

CERNLIB, HIGZCONV, BANNER...

Manuals

Contact with vendors, porting

F77, F90, C++... PATCHY, CMZ...

SUBMISSION PROCEDURE

LOCAL PROGRAM LIBRARY

* Code submitted locally by the author to the SITE Program Lib. fulfilling following conditions:
  - Standard FORTRAN /C for multiplatform versions
  - Porting done at least on 1 Unix platform + other (VMS, VM)
  - Test Suite with explicit messages upon succesful/unsucc. execution
  - Full documentation in LaTeX, SGML
  - Source code maintained with a SCMS(Patchy, CMZ...)
* If the code does not comply to any of these conditions, the PL staff may become the submitter, holding responsability for the necessary work

* The PL takes responsability for porting work on all their available platforms, and for maintenance of code and reporting to HEPLIB Site Coordination Commitee
* When the code is considered of general interest, submit to HSC

HSCC

* Submit to appropiate technical commitee (HMTC, HGTC...) for evaluation. If accepted, examine:

  DOC, Interfacing, Quality of code

* Iterate if required with local PL
* Pass it to Platform Support to complete porting on missing platforms. The submitter coordinates code changes

  Pass it to Editorial Commitee for inclusion into next release

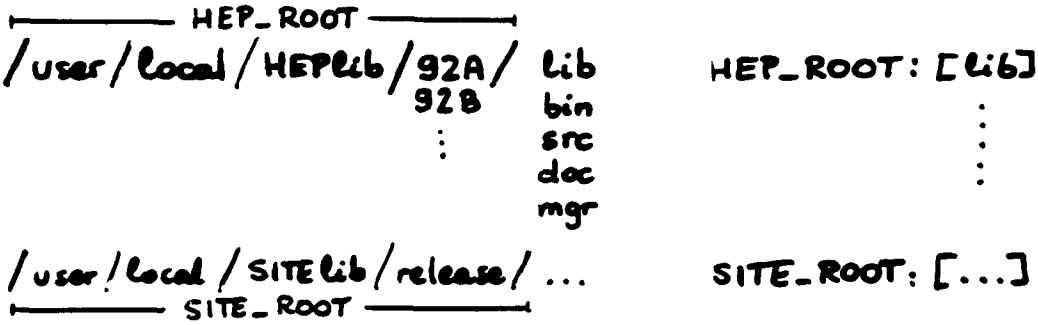
HEC

HEPLIB - CNL, Packing up of code into { BACKUP savesets, compressed tar files, ...

Notification of availability to members

CODE AVAILABILITY

* Release 1/2 per year



* Library path: $SITE_ROOT/lib : $HEP_ROOT/lib ...

CODE MAINTENANCE

* The submitter (site) maintains the code with the SCMS chosen by the Maint. Std. Commitee

* All other sites report to sub. bug fixes / porting to new platforms / improvements / extensions

CODE DISTRIBUTION

* Every site should have anonymous ftp and under "directories" { SITElib.release, HEPlib.release

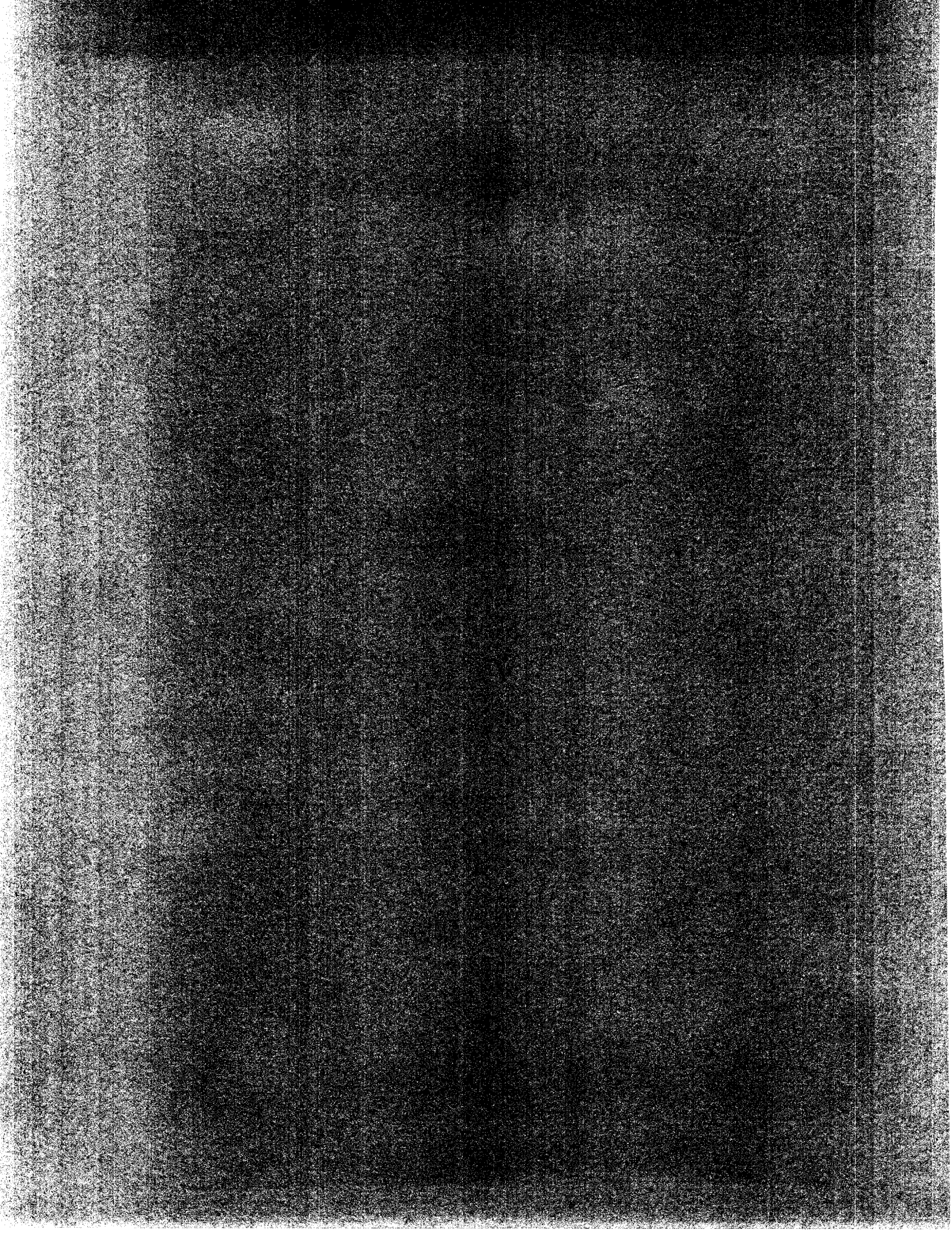
  - When code is uncorrelated to other elements (MC libs) the (submitter) site will host the distribution sets for ALL platforms (even those not directly available but contributed by Platform Support) and will inform all other members of the availability
  - When code is not a separate entity (MATHLIB) the Steering Commitee will asign coordination to a given site

CODE POLICY

* Written following language standards (no extensions)
* Avoid redundancy, unless competing app. are orthog
  in the user interface and both are felt attractive
* Tools/Utilities running on single platforms REJECTED!
* Programming Languages not available on all platforms REJ
  (unless equivalent replacement code is provided for port.
* Commercial Soft: Evaluation, contract conditions,
  endorsement

"SCRI Software Products"
"Ideas for HEPLIB"

S. Youssef



Software Products from SCRI

A data abstraction package for FORTRAN creates and manipulates typed "objects" including "universal utilities," allocatable and virtual memories, i/o, machine<->machine translation, package and help system creation etc. [TYPES]

Detects errors in GEANT geometries due to overlapping daughters or due to daughter volumes not contained by their mother volume. [GVerify]

Absorbs AutoCAD or VersaCAD DXF files into GEANT [OCTAGON]

A data definition language for TYPES [VF]

Translation from Zebra Banks to TYPES objects.
Direct HBOOK histograms from Zebra FZ files.
[dZebra/dHbook]

Creates clustering algorithms from a simple detector description. Optimizes clustering for chosen physics goals. [CTool]

A general purpose scientific visualization package running on Silicon Graphics or IBM/6000 machines. [SciAn]

A parallel implementation of GEANT using CPS
[See Phil Rulon's talk]

TYPES

- No preprocessing, calls to TYPES are in standard Fortran

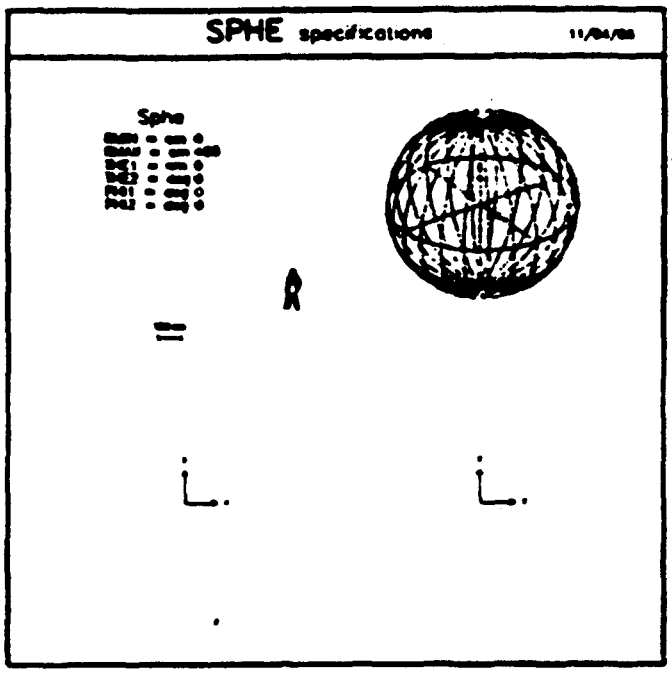
- It is easy to use bits of TYPES in otherwise normal programs. TYPES objects can be put in Local arrays, in common blocks, in Zebra banks, can be accessed from other languages.

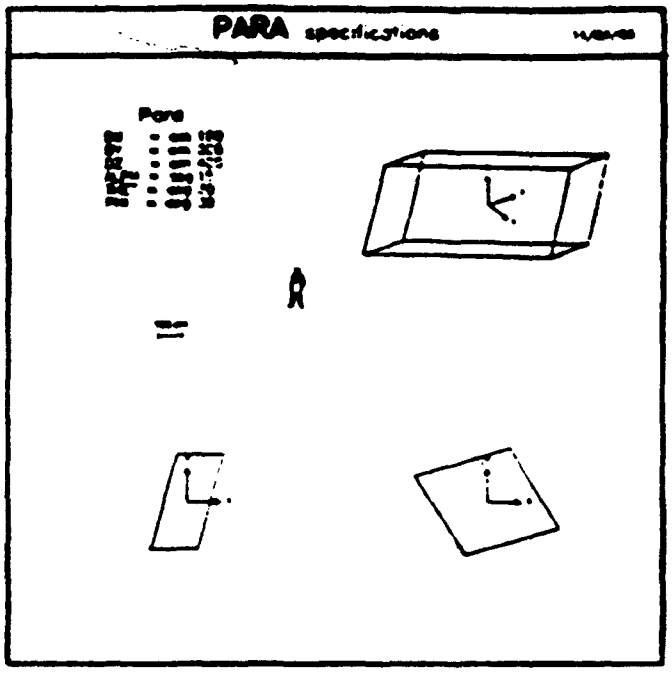
- In addition to TYPE specific operations, universal utilities e.g. CALL dDisplay(Q) or % ddisplay are very convenient.

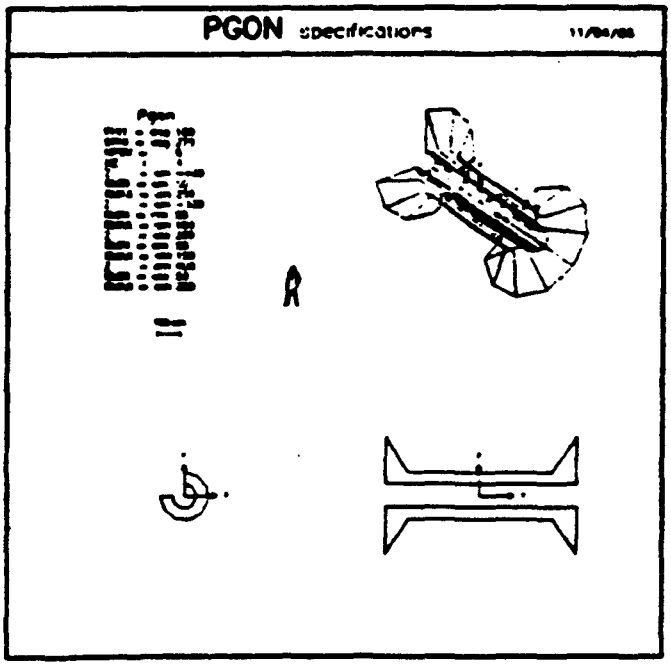
TYPES version 1.24 runs on

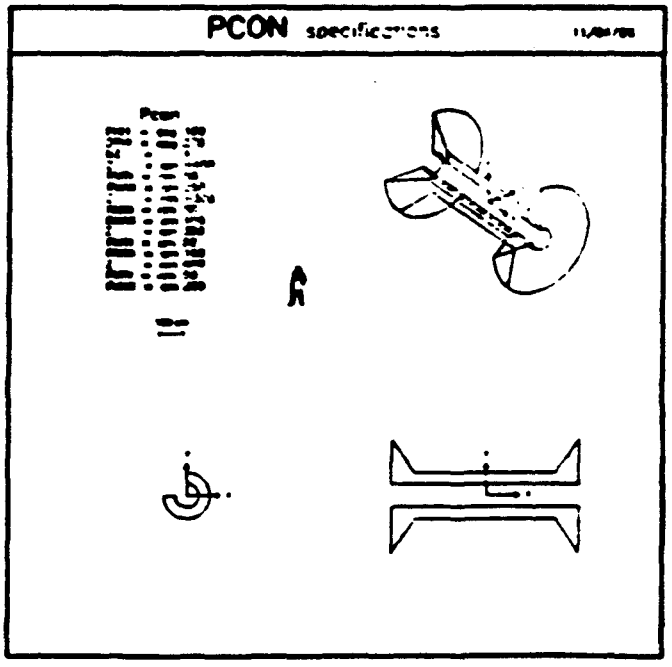
VAX/VMS, DEC/Ultrix, Silicon Graphics, SUN, Cray/Unicos and IBM/AIX.

It comes with ~50K lines of source code, an on-line help system and a users guide.


SPHE specifications
Sphe
R

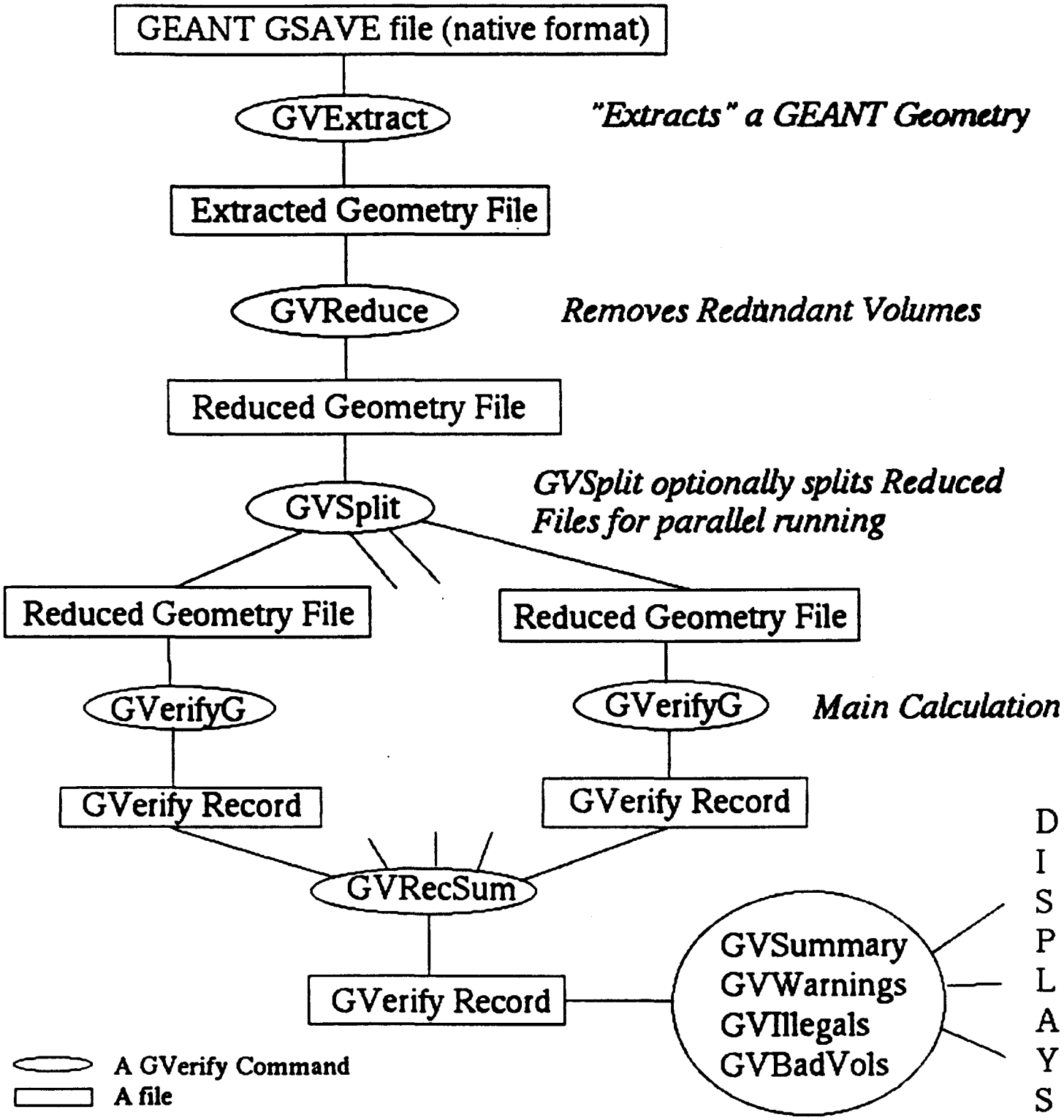

PARA specifications
Para
R


PGON specifications
R


PCON specifications
R

Running GVerify

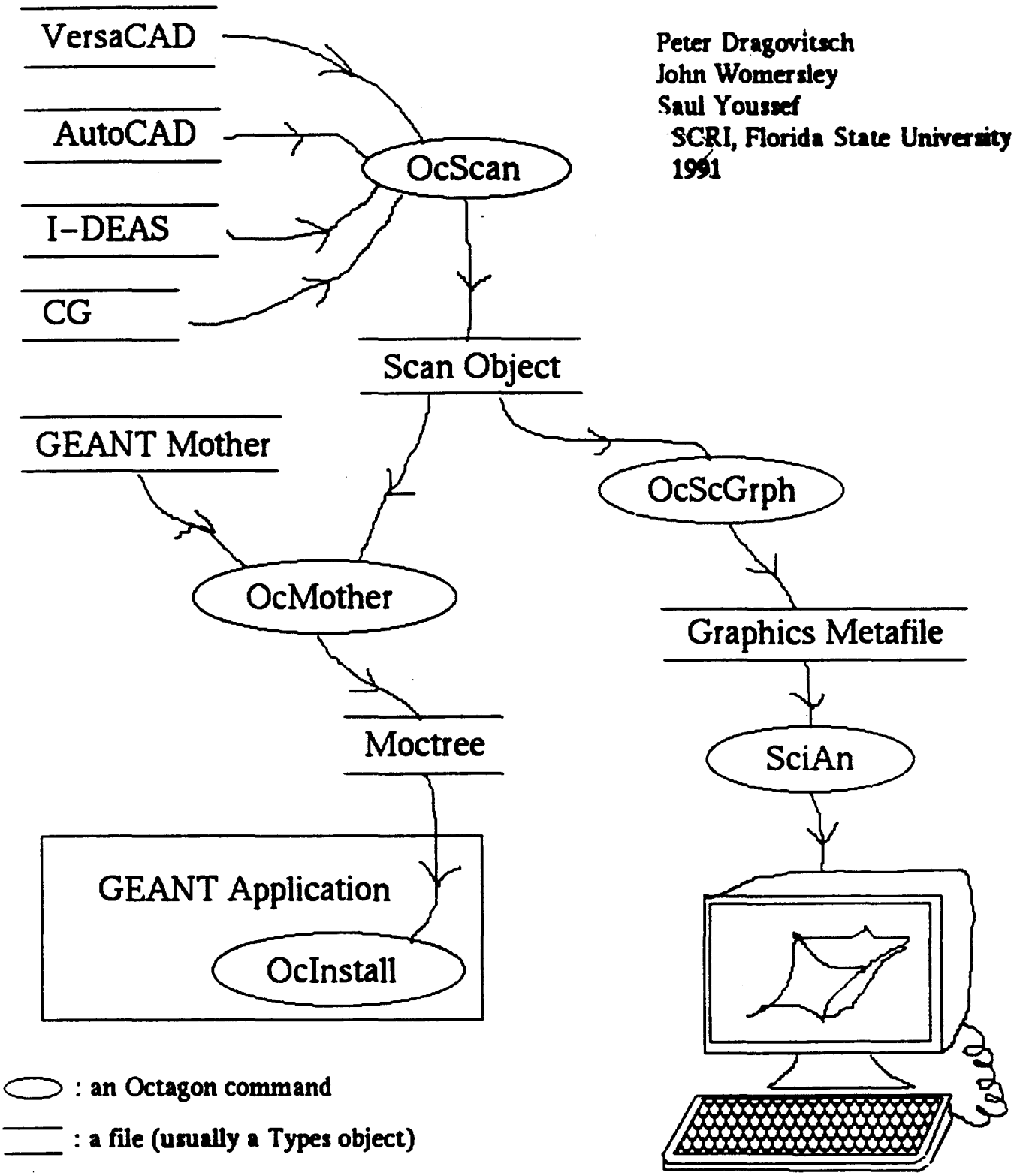
Saul Youssef
SCRI
November 1990

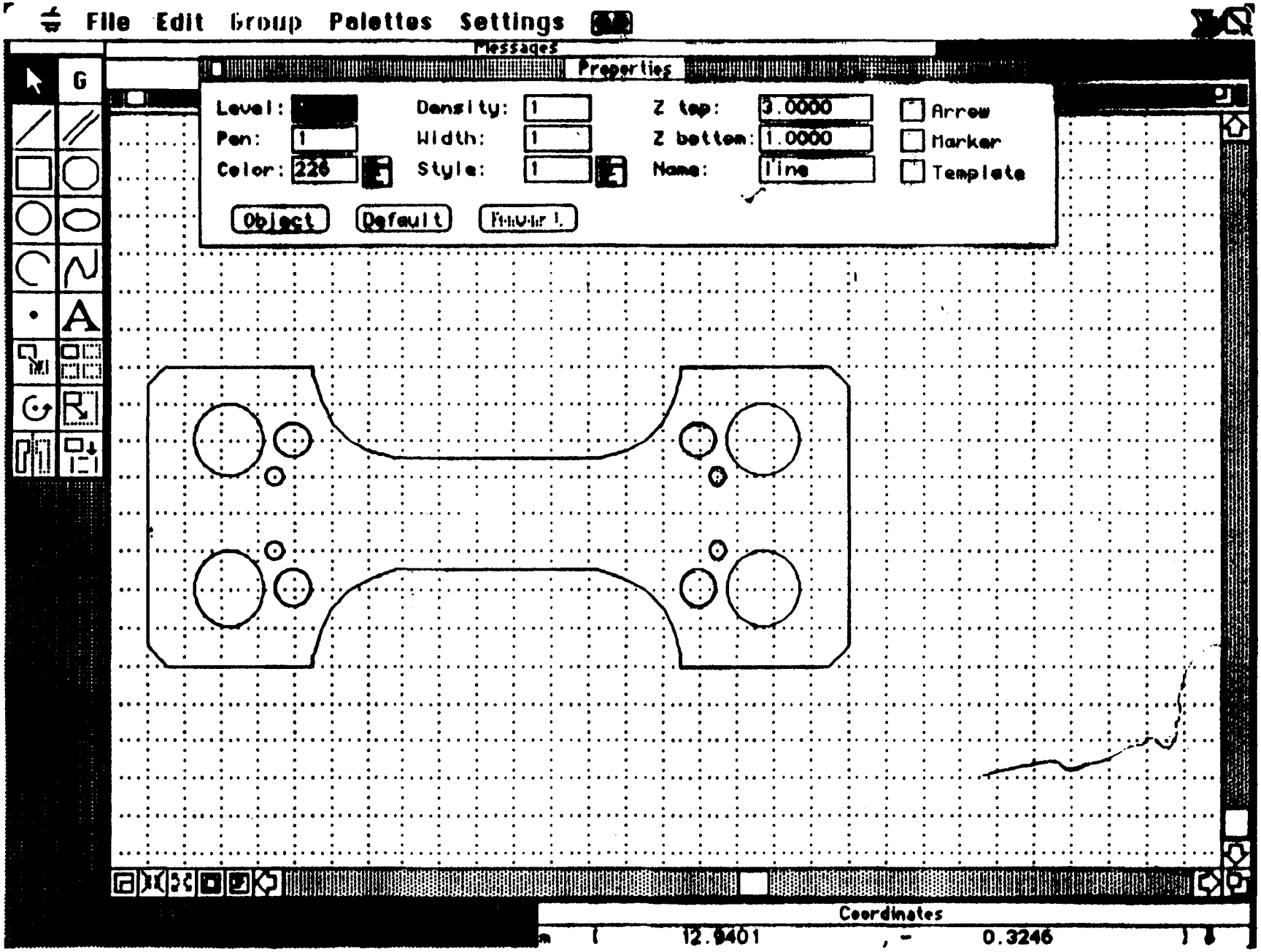


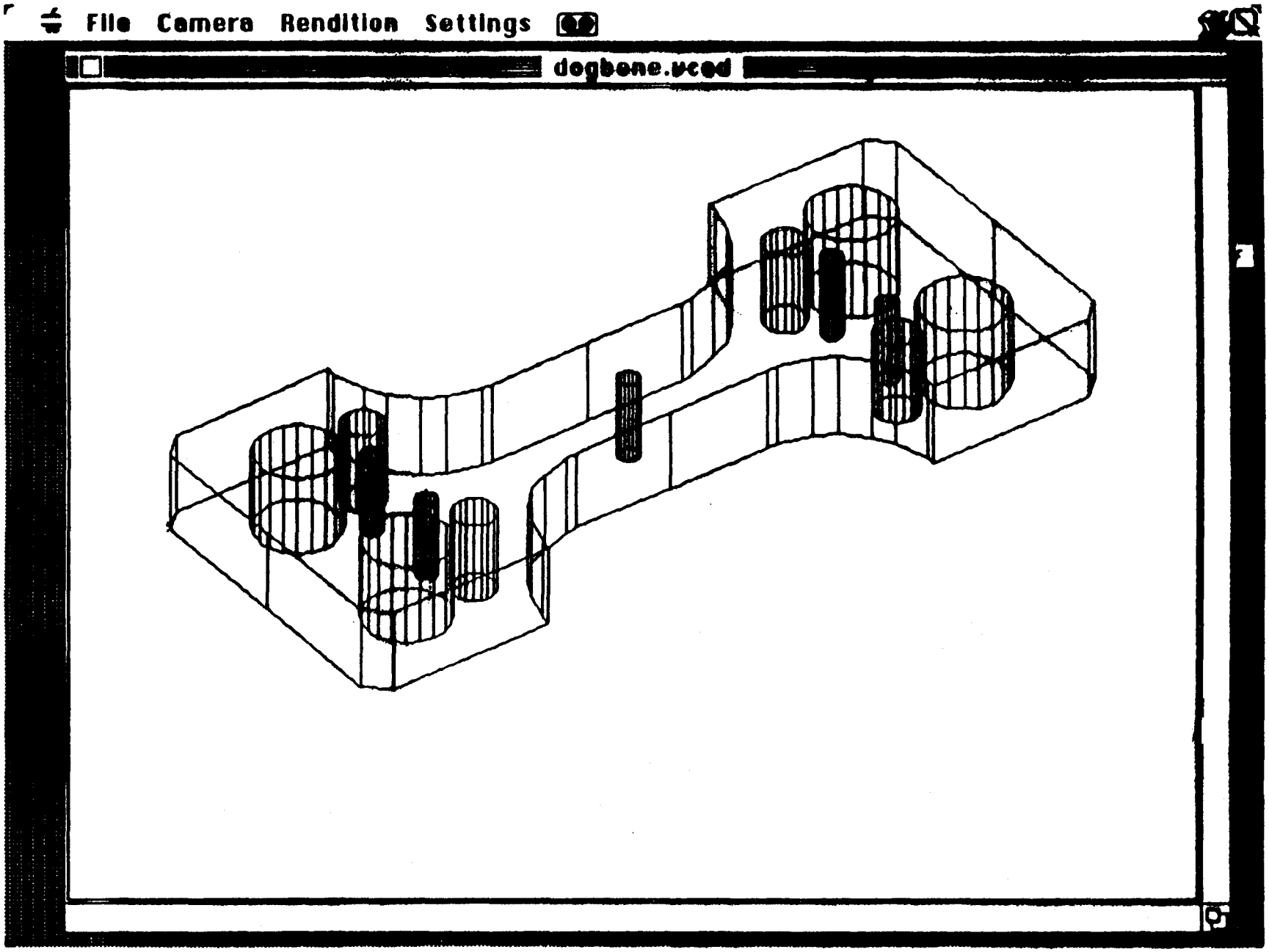
youssef@scri1.scri.fsu.edu or SCRI::youssef
youssef@

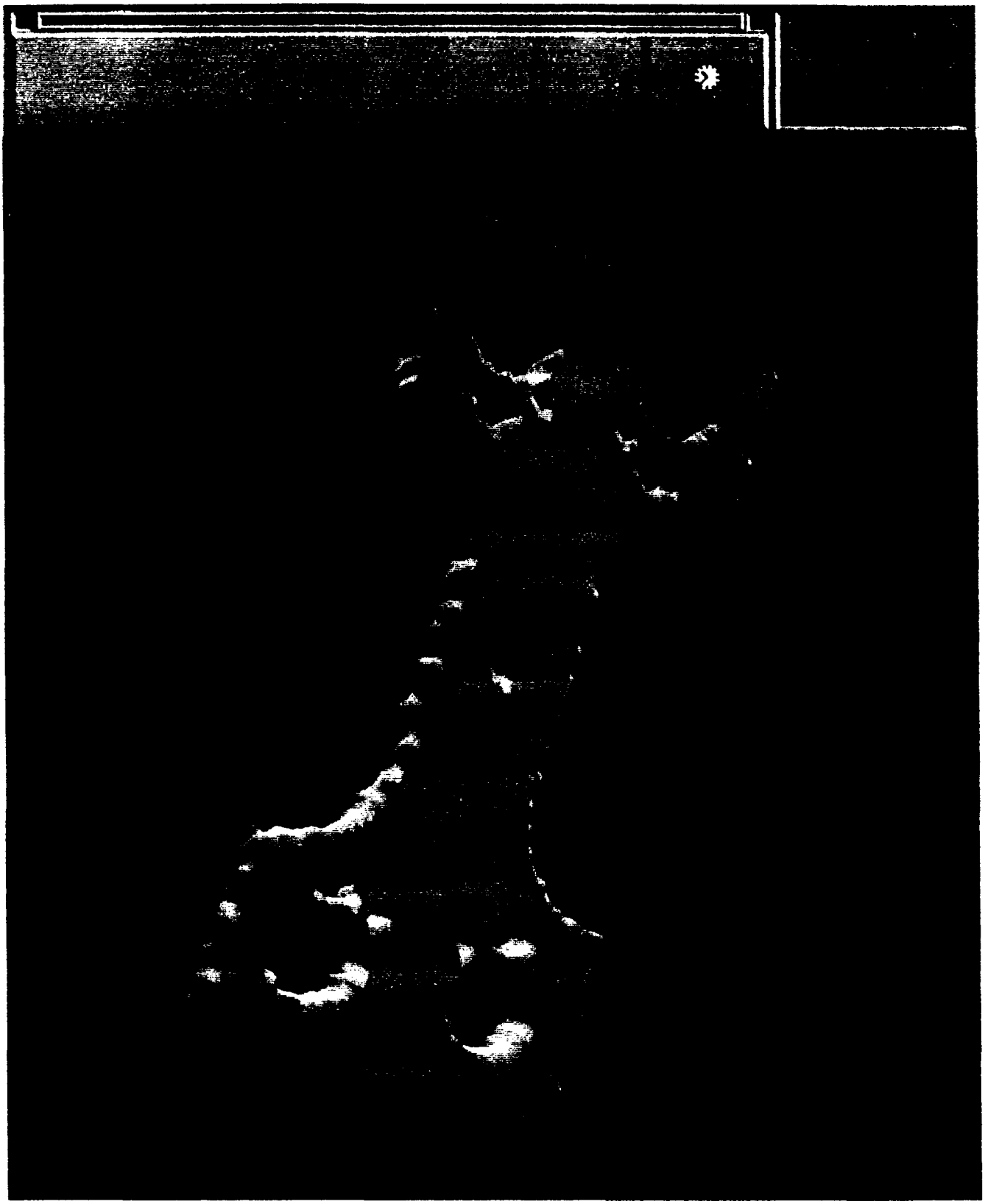
OCTAGON

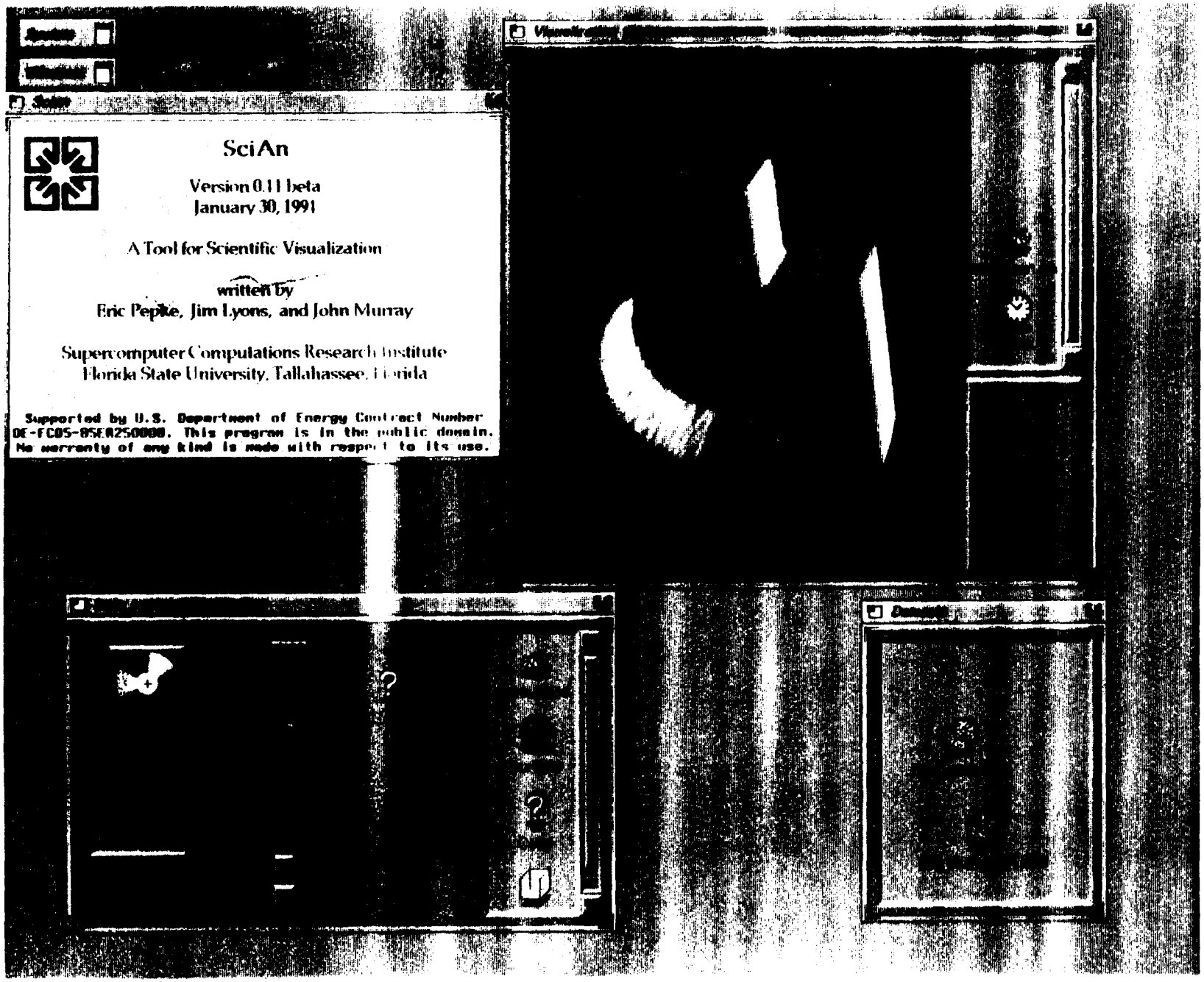
Peter Dragovitsch
John Womersley
Saul Youssef
SCRI, Florida State University
1991




File Edit Group Palettes Settings
Messages
Properties
Level:
Pen: 1
Color: 226
Density: 1
Width: 1
Style: 1
Z top: 3.0000
Z bottom: 1.0000
Name: line
Arrow
Marker
Template
Object
Default
Coordinates
12.9401
0.3246


File Camera Rendition Settings
dogbone.vcod



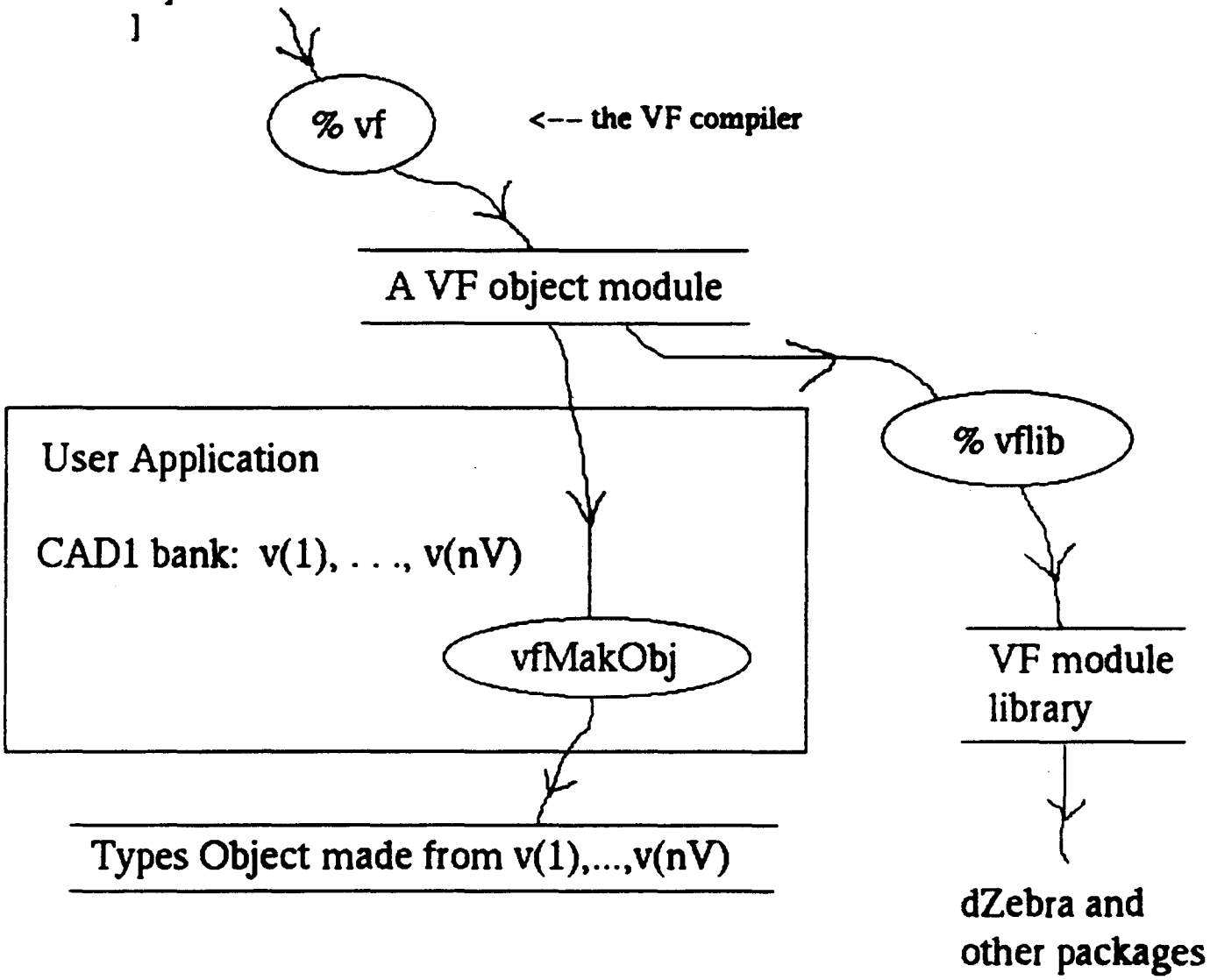

SciAn
Version 0.11 beta
January 30, 1991
A Tool for Scientific Visualization
written by
Eric Pepke, Jim Lyons, and John Murray
Supercomputer Computations Research Institute
Florida State University, Tallahassee, Florida
Supported by U.S. Department of Energy Contract Number
DE-FC05-85ER250000. This program is in the public domain.
No warranty of any kind is made with respect to its use.

VF: Vector Format Language

Saul Youssef
SCRI
Florida State University
1991

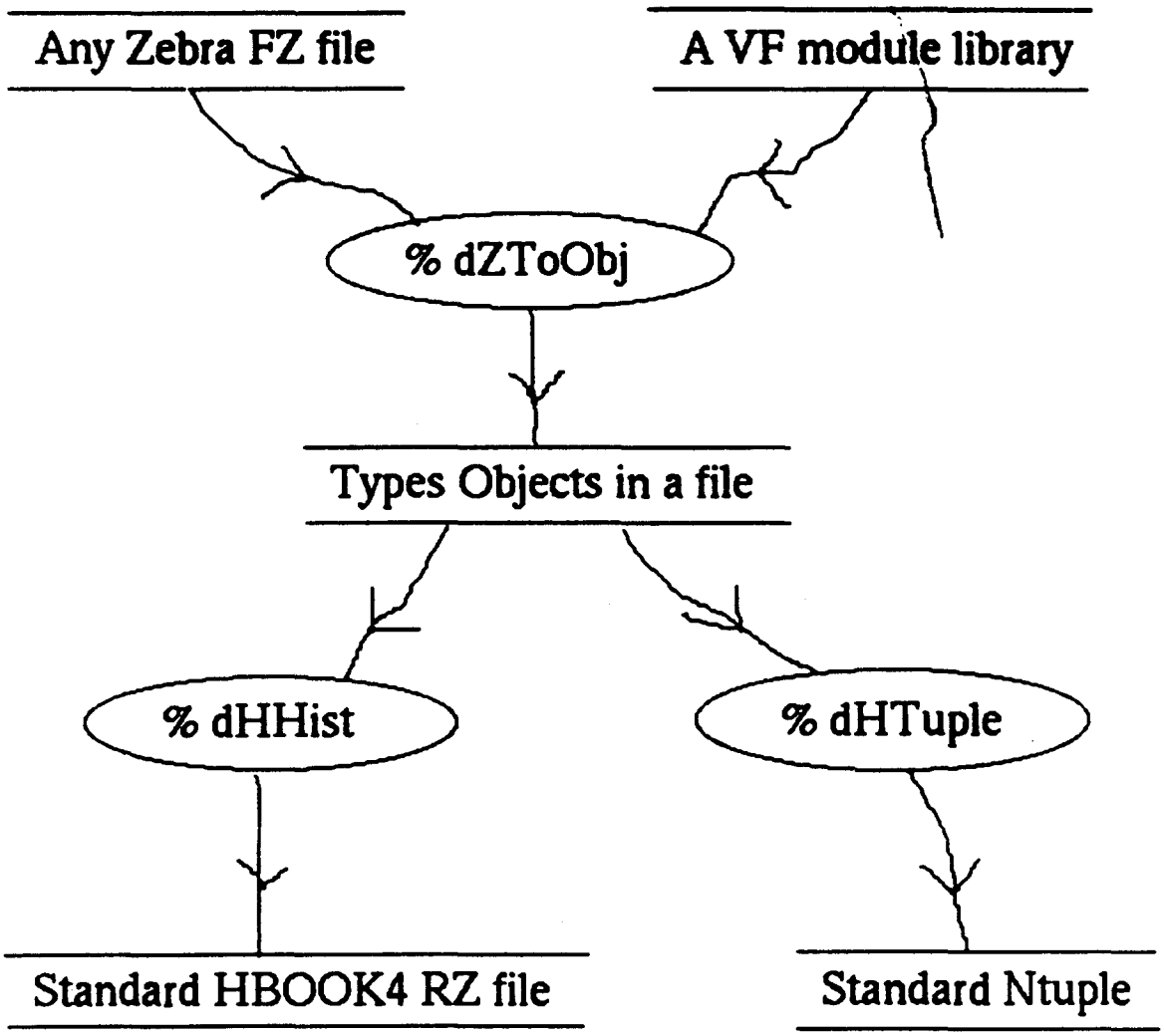
A VF program e.g.:

```
[CAD1,          ! a zebra bank description
  [Element Count,I],
  [Contents,
    #{Element Count}    ! repeat "Element Count" times
      [An element,<(address,16),(value,16)>]
  ]
]
```



dZebra / dHbook

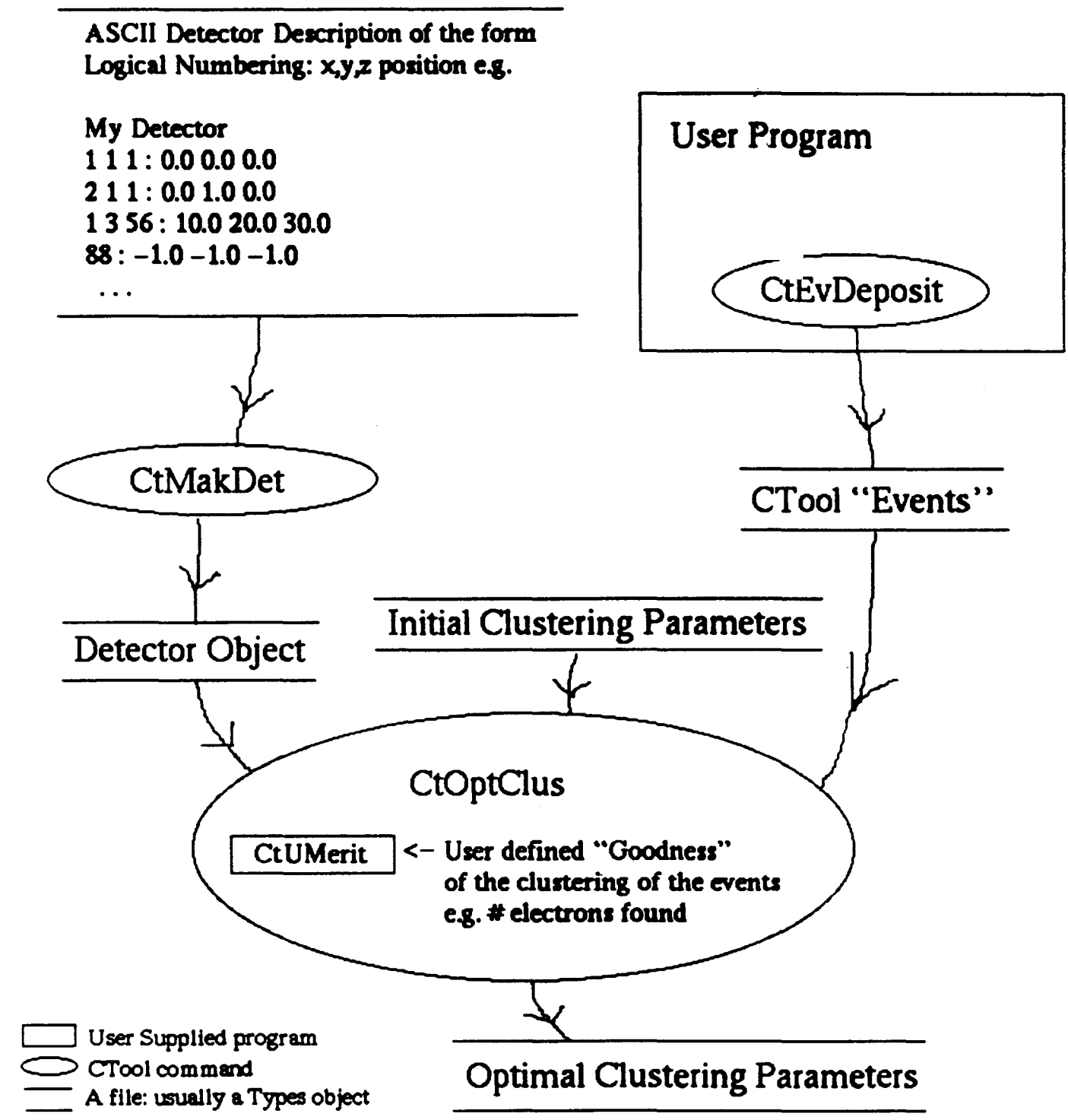
Saul Youssef
SCRI
Florida State Univ.
1991



```
!
!   This is a VF description of the CAD1 zebra bank as
!   taken from the .zeb file dated march 6,1990 by Chip Stewart
!   S.Youssef, August, 1991.
!
[CAD1,                          ! NON PULSER format
  #1                            ! hard wired number of crates
  [A Crate,
    [Header Length,I],
    [Synch Word, <(all_on,16),(Level 1 Code,16)>],
    [Controller Word,
      <(ADC mode control,8),
       (BLS mode control,8),
       (No. ADC Cards -1,8),
       (Crate Number,8)
      >
    ],
    [Software Version Number,
      <(Increm.Version Number,16),
       (User Index,8),
       (Junk,6),
       (Monte Carlo?,1),
       (NWA or D0?,1)
      >
    ],
    [Status/Vertex,<(Vertex,16),(Status,16)>],
    [Cards,
      #(No. ADC Cards -1)+1
      [A Card,
        [No.Words in this card,I],
        [Card Data,
          #(No.Words in this card)
            [A Word,<(Pulse Height,16),(Channel Address,16)>]
        ]
      ]
    ]
  ]
]
```

CTool: Clustering Toolbox

Dong Xiao
Saul Youssef
SCRI, FSU
August 1991



Method of Software Distribution

To get TYPES on machine XXX =

aix (IBM/AIX)
cra (Cray/Unicos)
sig (Silicon Graphics)
sun (SUN/OS)
ulx (DEC/Ultrix)
vax (DEC/VMS)

(a) Copy all the files from 46426::disk$sys1:[youssef.types.XXX]

(b) % source XXX_instl.com or @XXX_instl as appropriate

For any TYPES based package e.g. GVerify, VF, Octagon etc.

(a) Copy the appropriate ".pac" ascii package file e.g.

46426::disk$sys1:[youssef.vf]vf10.pac

(b) % dupackag

HEPLIB Ideas

What problem are we trying to solve?

Purpose: To give the HEP community knowledge of and easy access to useful software from both inside and outside of HEP.

This includes

- o Distributing software, documentation and instructions for getting software
- o Reviewing software developments inside and outside of HEP
- o Collecting user experience and transmitting this to the authors
- o Establishing standards when needed

HEPLIB Subject Areas

- Accelerator Simulation
- Analysis and Data Reduction
- CAD/CAE Systems
- Computer Hardware and Benchmarks
- Data Bases
- Detector Simulation
- Event Generators
- Graphics, Visualization and User Interfaces
- Languages, Parallelism and Data Structures
- Mathematical Software
  Including Symbolic Math., Linear Algebra, Integration, Statistics
- Networking, Email, Remote Conferencing and Software Distribution
- Particle Properties and Cross Sections
- Software Engineering
  Including Code Management, CASE tools etc.
- Text, Hypertext, Hypermedia, Editors and Typesetting
- Theoretical calculations, Field theory, Lattice Gauge Theory

High Energy Physics is not yet possible on other Planets !

We have a serious responsibility not to be exclusive.

HEPLIB should represent the entire world HEP community including all nations, laboratories and universities.

Prime Directive: HEPLIB must accept all software which might be generally useful in the community.

I think that HEPLIB as an organization should not attempt to write or support software packages.

This is too much of a distraction and it is (in my view) not really the problem that we are trying to solve.

A Possible Organization for HEPLIB

(a) A users group consisting of anyone who wants to join

(b) One "Editor" for each subject area. Editors are chosen for their expertise and to represent the whole HEP community.

Maintaining access to software and/or instructions for getting software in their area.

Collecting user experience and Reviewing developments in their area.

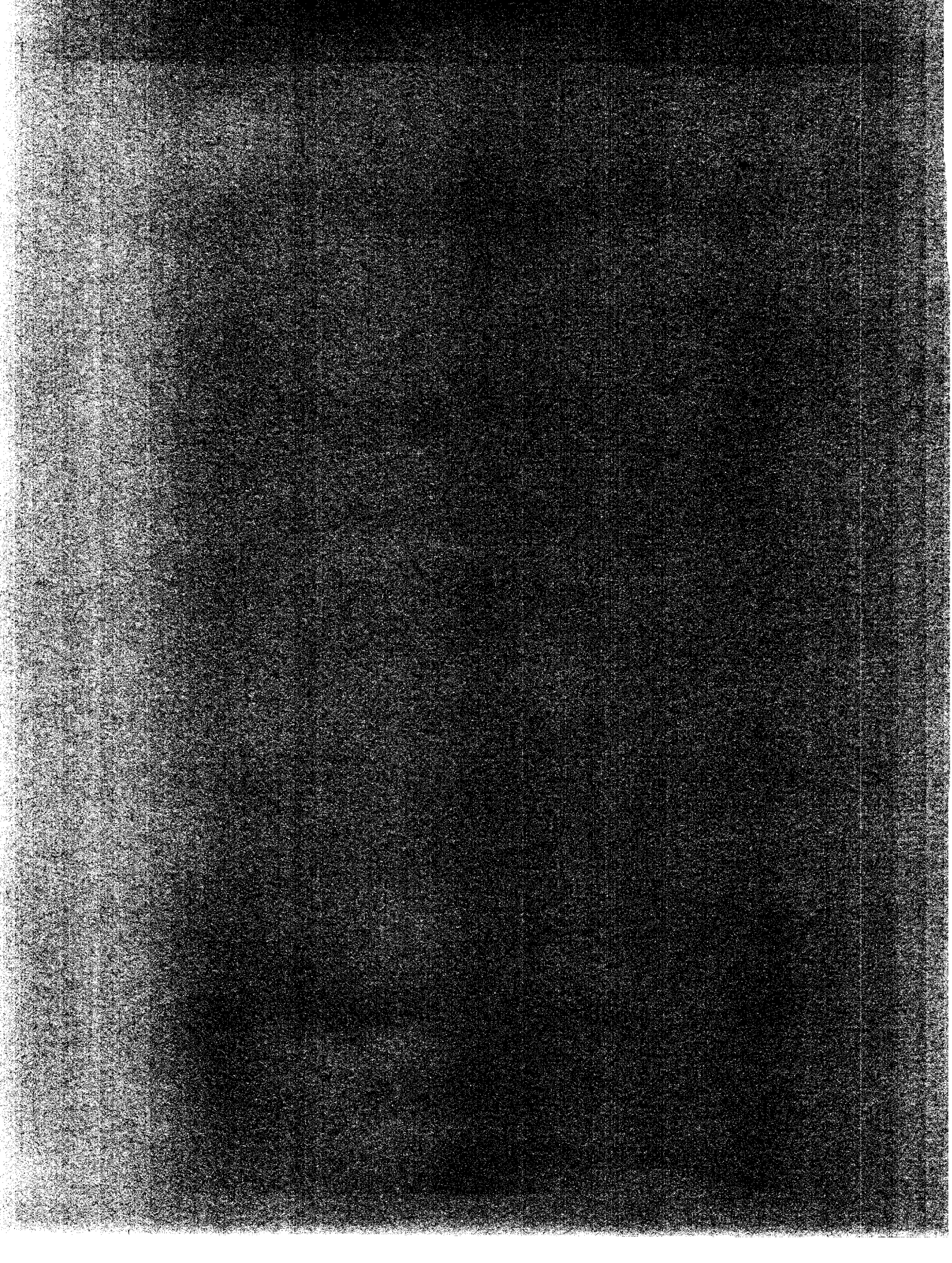
(c) A Managing Editor

To coordinate the editors. To call and chair meetings of the editors and user group. To organize the HEPLIB Journal.

Editors and the Managing Editor could have a two year term(?). How should the editors be chosen?

"SDC Requirements to HEPLIB"

L. Price



SDC AND HEPLIB

HEPLIB '91

L. E. Price
September 19, 1991

The SDC Detector

- Major, general-purpose detector for the SSC
- 2T Solenoid magnet, r=1.7m, l=8m
- Tracking: Silicon strips plus straw PWC or Fiber central tracker
- Scintillating tile/WLS fiber calorimetry
- Complete muon system
- $\approx 10^6$ electronics channels
- > 100 institutions, > 700 physicists, > 10 countries
- Approved in December, 1990 to proceed to a full proposal in April, 1992

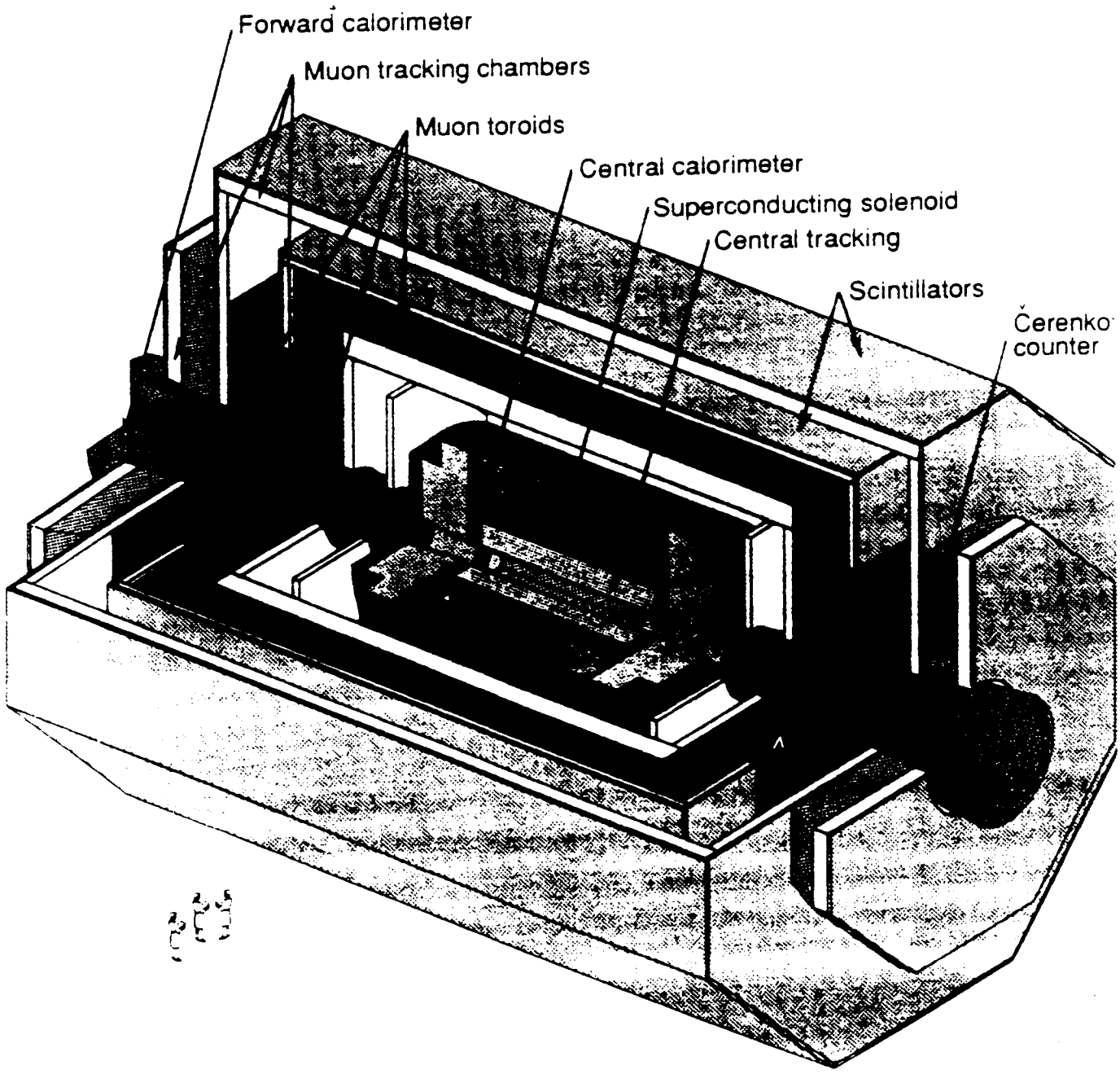


FIG. 1. Overall view of the solenoid detector.

Current SDC computing: simulation for detector design

- Draws heavily on common HEP software, especially from CERN
- SDCSIM
  - GEANT
  - KUIP
  - HBOOK
  - ZEBRA
  - PATCHY/CMZ
  - CPS

The SDC computing task

- Up to 1 megabyte per event
- Up to 100 events/second
- Reconstruction and analysis requires $10^5 - 10^6$ VUPs
- Storage requirements: up to 1000 TB (1 PB) / year
- Up to 100 MB/sec writing speed
- The data storage and access requirement is thought to be much harder to meet than the processing requirement

- Major software components
  - Run control
  - System monitoring
  - Data monitoring
  - Online database
  - Online user interface/display
  - Triggering
  - Storage manager/database
  - Reconstruction (near online)
  - Analysis
    - Storage manager/database
    - User interface
      - Histogramming/displays
    - Computing resource manager

Sources of software

- Write ourselves
- Commercial
- HEPLIB

Does SDC need HEPLIB?

- Yes!
- Much new software is needed that could be shared with other experiments
- We will try to use commercial software where possible, but...

  Buying licenses in large multiples is often prohibitive

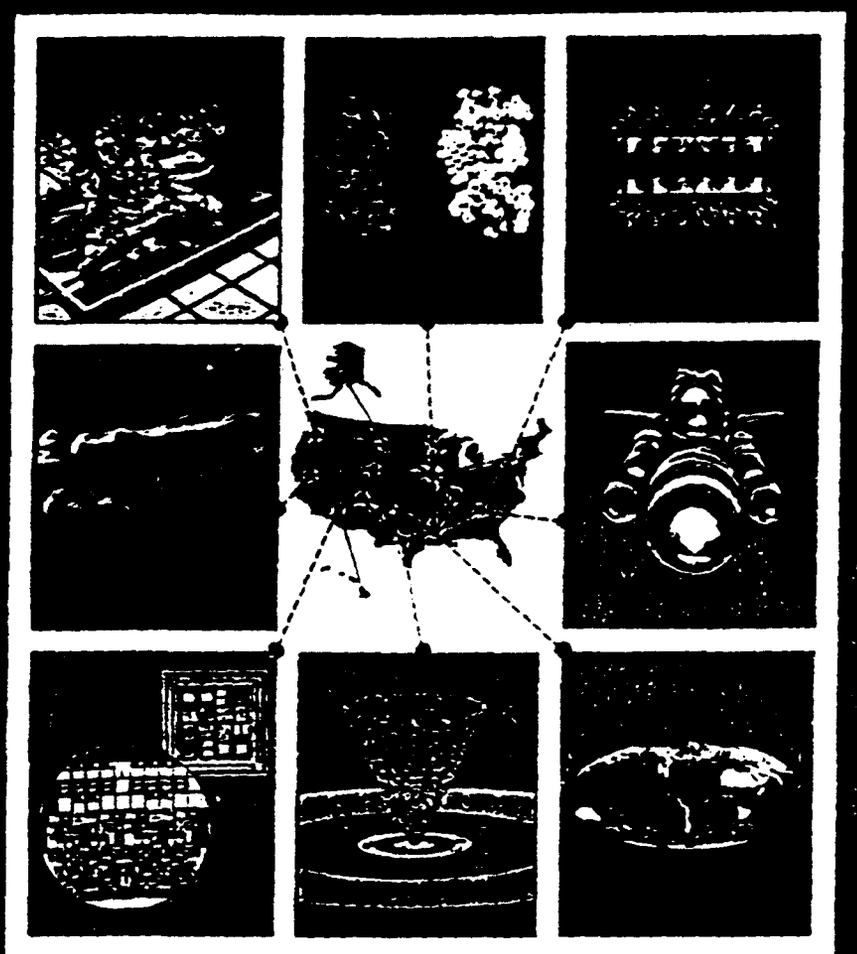
  Some HEP requirements are unique

How would SDC contribute to HEPLIB?

- Ideally, SDC (and SSCL) would develop a portion of the new software needed, getting some of the remainder from other HEPLIB Hcontributions
- SDC already working on data storage and access problem with help from US High Performance computing and Communications initiative (HPCC)
- One challenge is to coordinate distributed software development closely enough that detectors can reasonably count on software they plan to use from others.

Grand Challenges:
High Performance Computing and Communications

The FY 1992 U.S. Research and Development Program



A Report by the Committee on
Physical, Mathematical, and Engineering Sciences

To Supplement the President's Fiscal Year 1992 Budget

Figure 1 The High Performance Computing and Communications Program

Goals: Strategic Priorities

Extend U.S. technological leadership in high performance computing and computer communications.

Provide wide dissemination and application of the technologies both to speed the pace of innovation and to serve the national economy, national security, education, and the global environment.

Spur gains in U.S. productivity and industrial competitiveness by making high performance computing and networking technologies an integral part of the design and production process.

Strategy: Integrating Priorities

Support solutions to important scientific and technical challenges through a vigorous R&D effort.

Reduce the uncertainties to industry for R&D and use of this technology through increased cooperation between government, industry, and universities and by the continued use of government and government-funded facilities as a prototype user for early commercial HPCC products.

Support the underlying research, network, and computational infrastructures on which U.S. high performance computing technology is based.

Support the U.S. human resource base to meet the needs of industry, universities, and government.

Program Components

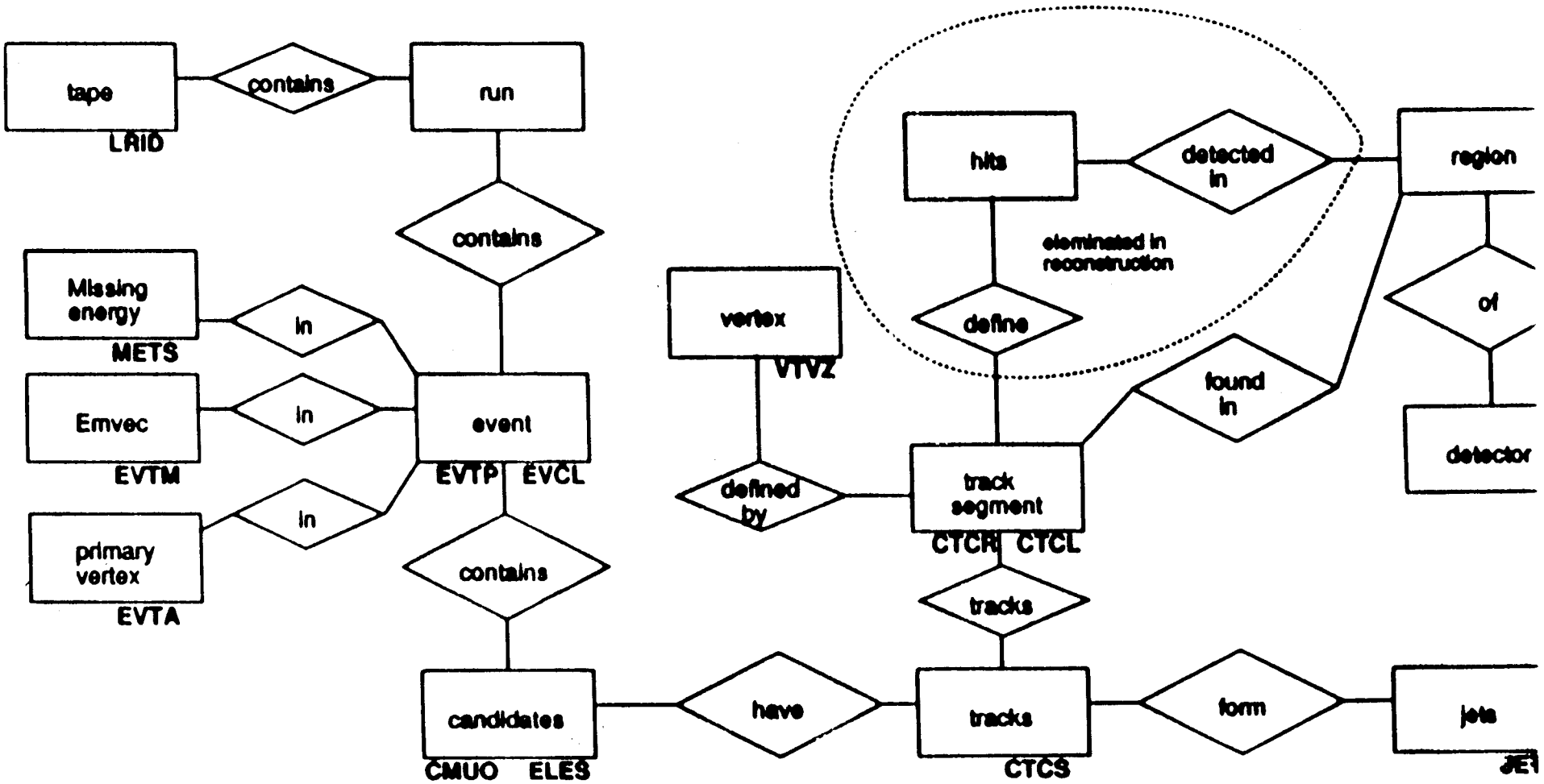
High Performance Computing Systems
- Research for Future Generations of Computing Systems
- System Design Tools
- Advanced Prototype Systems
- Evaluation of Early Systems

Advanced Software Technology and Algorithms
- Software Support for Grand Challenges
- Software Components and Tools
- Computational Techniques
- High Performance Computing Research Centers

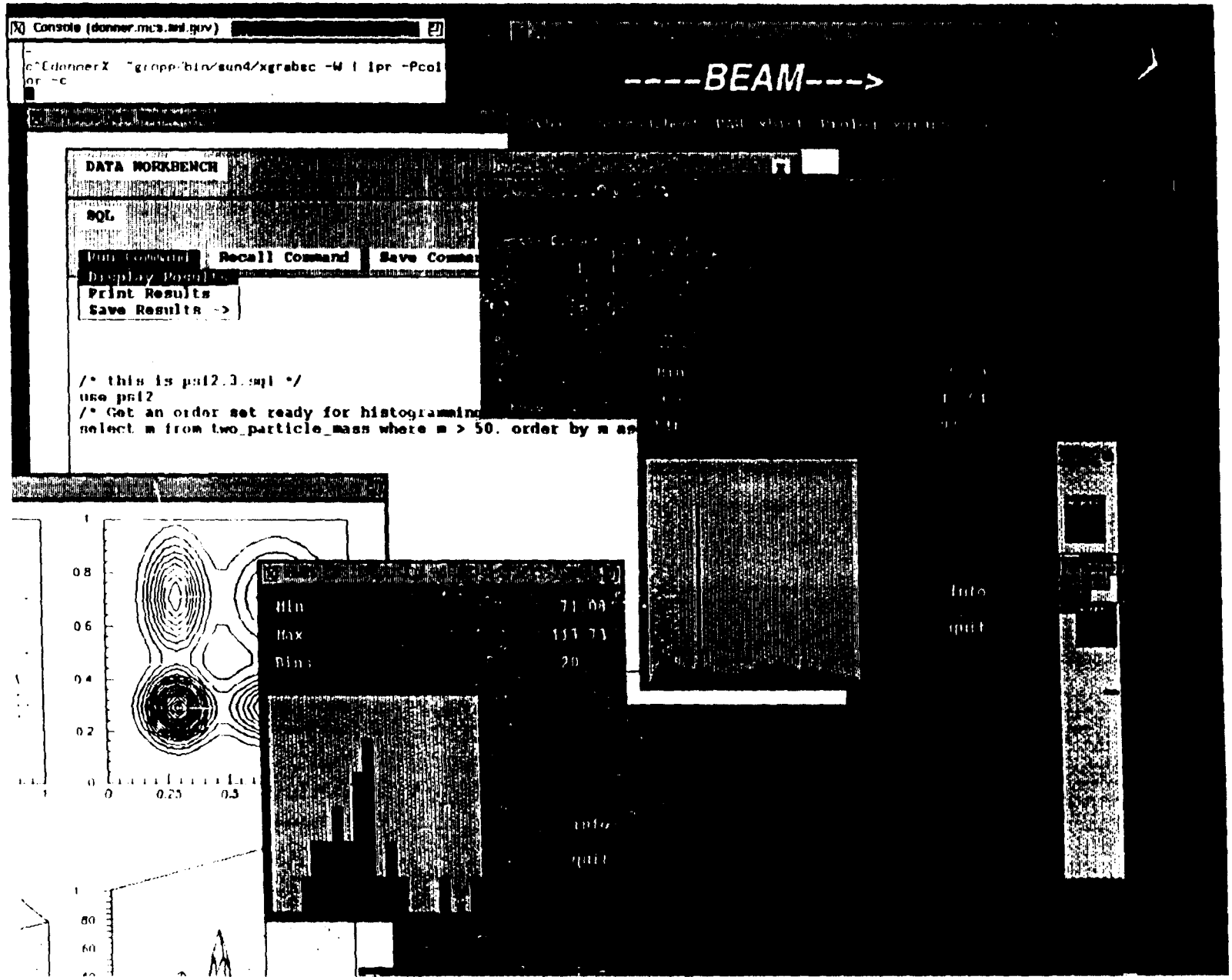
National Research and Education Network
- Interagency Interim NREN
- Gigabits Research and Development

Basic Research and Human Resources
- Basic Research
- Research Participation and Training
- Infrastructure
- Education, Training, and Curriculum

70


tape
LRID
contains
run
contains
Missing energy
METS
In
Emvec
EVTM
In
primary vertex
EVTA
In
event
EVTP EVCL
contains
candidates
CMUO ELES
hits
detected in
region
eleminated in reconstruction
define
vertex
VTVZ
found in
of
detector
defined by
track segment
CTCR CTCL
tracks
have
tracks
CTCS
form
jets

71


Console (donner.mcs.anl.gov)
----BEAM--->
DATA WORKBENCH
SQL
Recall Command
Print Results
Save Results ->
/* this is pai2.3.sql */
use pai2
/* Get an order set ready for histogramming
select m from two_particle_mass where m > 50. order by m as
Min
Max
Bins
Auto
quit
0.8
0.6
0.4
0.2
0
0.25
0.5
80
60

Principles

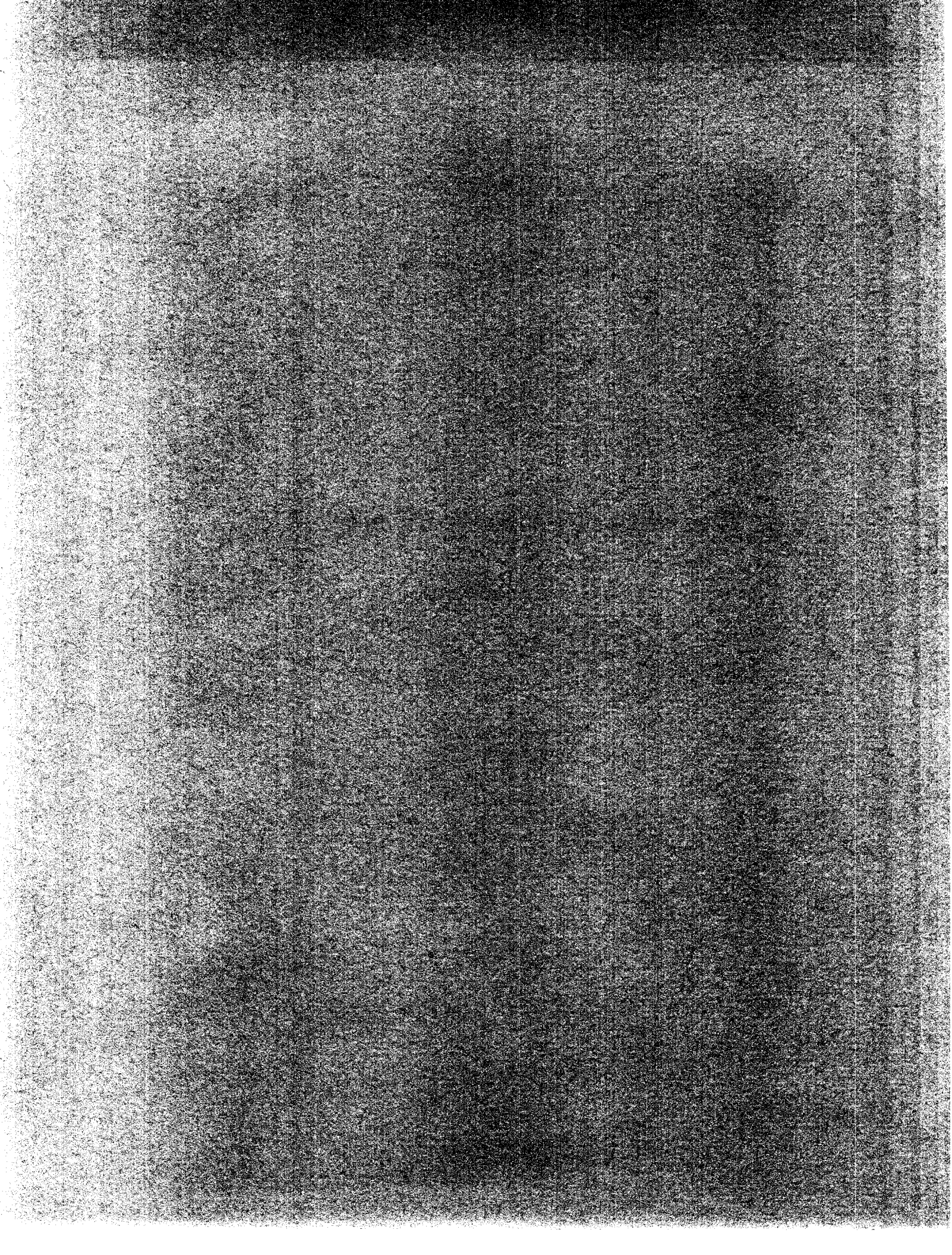
- Modularity (Interface specifications?)
- Portability (What platforms?)
- Documentation
- Proposals for planning use
- Review–suggest ways to make planned software more general
- Support and maintenance
- Core organized around large accelerator laboratories with programs to support

Getting going

- A multi-institution, multi-continent organization that will meet all requirements may not be possible
- But we should make every effort at this meeting to set up an organization that will try to make it work
- We have talked for several years. This meeting is the best opportunity we will have to bring HEPLIB into reality
- We should leave with some action plan

"Overview of Physics Software at KEK"

K. Amako



Overview of Physics Software at KEK

Katsuya Amako

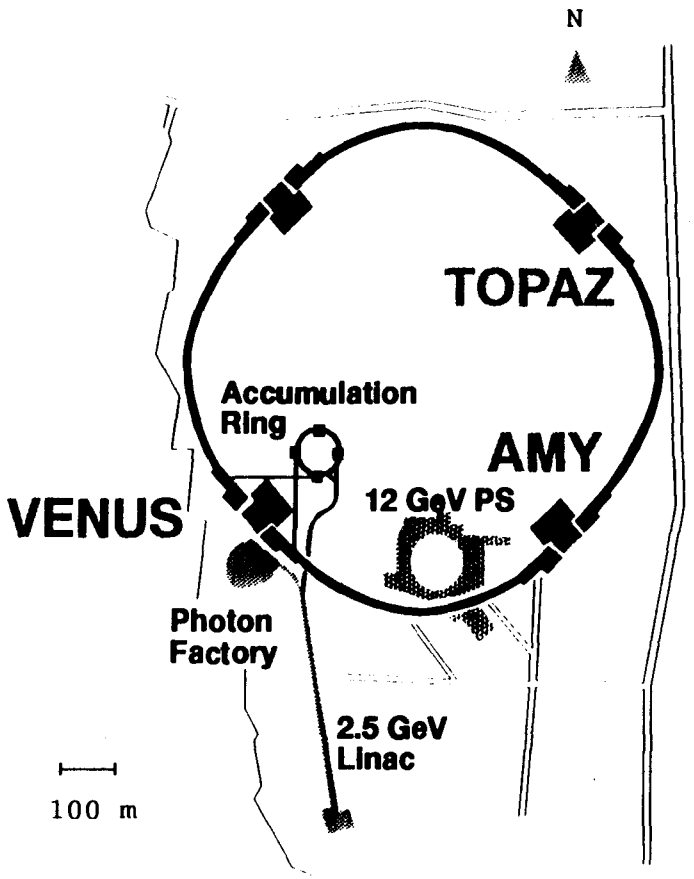
(Physics Dept., KEK)

HEPLIB '91

September 19, 1991

SSCL, Texas, U. S. A.

TRISTAN Experiments



TRISTAN : $e^+ e^-$ Collider

$E_{cm}^{max} = 64$ GeV

$L^{max} = 1.4 \times 10^{31}$ cm^{-2} s^{-1}

Date Experiments Started : Dec. 198(

Major Experiment Groups

VENUS (~75 Physicists, 18 Institutes)

A Magnetic Spectrometer with
Large Angular Coverage.

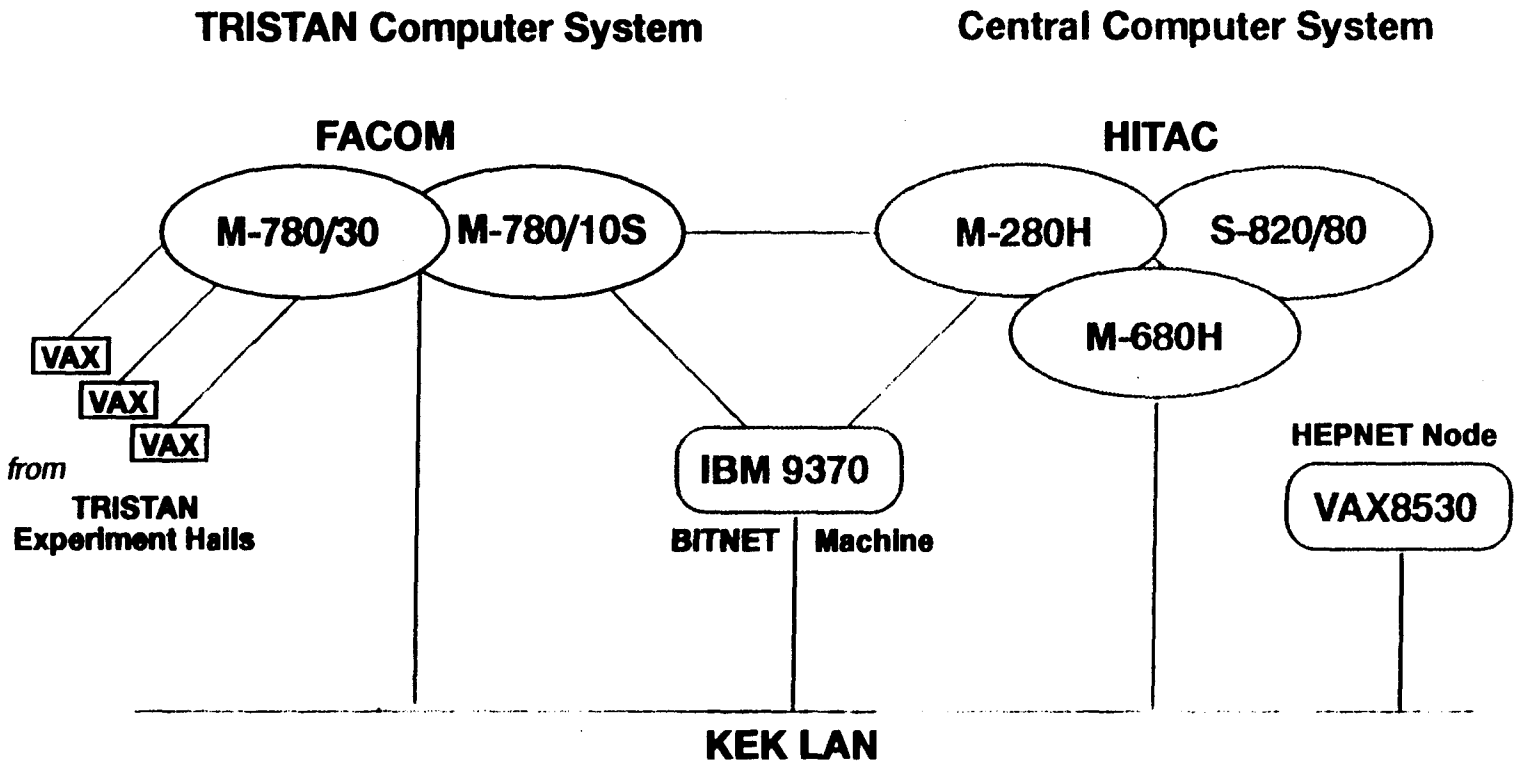
TOPAZ (~70 Physicists, 12 Institutes)

A Magnetic Spectrometer with
a Good Particle Identification.

AMY (~100 Physicists, 20 Institutes)

A Compact Magnetic Spectrometer
with Strong Magnetic Field.

KEK Data Handling Center



Programing Environment

	VENUS	TOPAZ	AMY
Major Machine	FACOM	FACOM, VAX	FACOM, VAX
Interactive Environment	NEWLIB	PFDE, NEWLIB VAX/VMS	PFDE, VAX/VMS
Programing Language	FORTRAN Assembler	FORTRAN Assembler	FORTRAN Assembler
Code Management	No Special Tool	No Special Tool	VAX/CMS
Coding Rule	Yes	Yes	Yes
Memory Management	ZBOOK	TBS (Tristan Bank Sys.)	TBS/AMY (Tristan Bank Sy.)
Graphic Tool	UGS2	UGS2	UGS2
Library	CERNLIB (Partially) SSL II	No CERNLIB SSL II	CERNLIB
Data Base	Coded by Group	Coded by Group	Coded by Group

Detector Simulator - Tool

	VENUS	TOPAZ	AMY
Framework	**Coded Originally** Basic techniques -> Adopted GEANT (ver.2, 1982)	**Coded Originally**	**Adopted the TOPAZ's frame**
Shower Handling			
Electromagnetic	**EGS4**	**EGS4** **Frozen Shower** (in Barrel Cal.)	**EGS4**
Hadron	**Parametrization** (based on test beam)	**GHEISHA**	**GRANT**
CPU Consumption			
Bhabha	**20 sec**	**2.8 sec**	**200 sec** (slow version)
Hadron	**3.5 sec**	**4 .0 sec**	**10~20 sec**
Muons	**0.4 sec**	**0.2 sec**	**0.2 sec**

Physics Analysis - Tool

	VENUS	TOPAZ	AMY
Histograming	HBOOK	HANDYPAK HBOOK	HBOOK
Plotting	GRHIST (Original package for VENUS usage)	DIS/TOPAZ TOP DRAWER	HPLOT PAW TOP DRAWER
Fitting/ Minimization	GRHIST MINUIT	MINUIT DIS/TOPAZ	MINUIT PAW
Documentation	LaTex PHYZZX MAC (ATF)	LaTex PHYZZX MAC (ATF)	LaTex PHYZZX MAC (Runoff)

HELPLIB in TRISTAN Age

1. In the age of TRISTAN, each experiment group in KEK developed and maintained its own HEPLIB. Therefore, the view point of a unified HEPLIB environment, the HEPLIB environment in Japan was chaotic.

2. Possible causes of this situation are
   - each group has its own history in software,
   - egoism of each group,
   - there was (is) no unified HEPLIB environment,
   - shortage of manpower in the KEK computer center,
   - e.t.c.

Major Japan HEP Projects in Future

Domestic Projects

- Asymmetric B-factory at KEK

$e^+ e^-$ Collider	8 GeV x 3.5 GeV
Luminosity	$1 \times 10^{34} cm^{-2} sec^{-1}$

- JLC (Linear Collider at KEK)

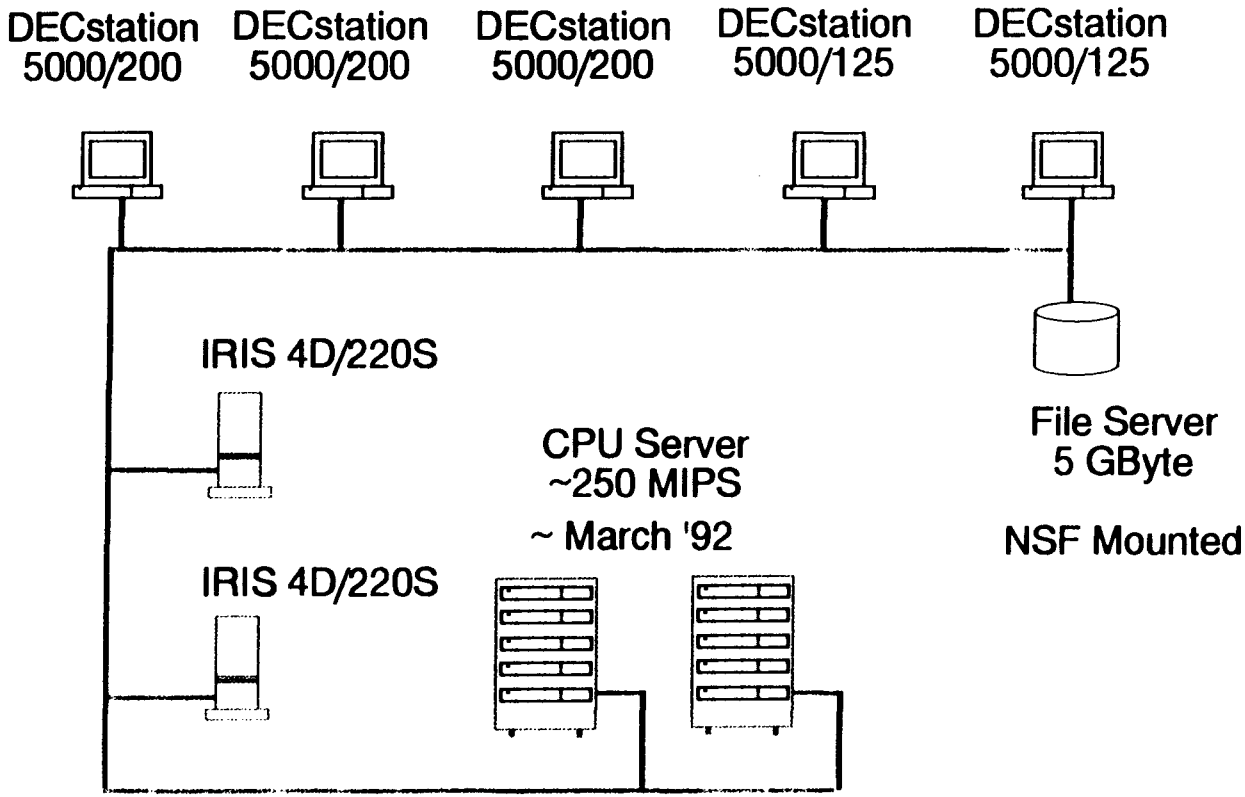
$e^+ e^-$ Collider	750 GeV x 750 GeV
Luminosity	$3 \times 10^{33} cm^{-2} sec^{-1}$

International Collaboration

- SSC/SDC

p p Collider	20 TeV x 20 TeV
Luminosity	$1 \times 10^{33} cm^{-2} sec^{-1}$

UNIX Environment for SDC/KEK



HEPLIB for SDC/Japan

Principle of Organization

Provide the same environment as in SSCL.

Present Structure of the Library

/usr1/machine_name/lib
- /cern
- /geant
- /init
- /lund
- /cmz
- /isajet
- /pgeant
- /cps
- /herwig
- /listserv
- /sdcshell

Management of HEPLIB for SDC/Japan

(Adopted the suggestion by H. Johnstad)

3 Separate Managements

1. Generic (Manager = Y. Morita : KEK Compter Center)
   - CERNLIB
   - CMZ
   - PATCHY
   - INIT
   - Documentations

2. Physics (Manager = Y. Takaiwa : KEK Physics Dept.)
   - ISAJET
   - LUND/PYTHIA
   - HERWIG

3. SDC Specific (Manager = Y. Takaiwa : KEK Physics Dept.)
   - SDCSHELL
   - PGEANT

(Contact Person = K. Amako : KEK Physics Dept.)

HELPLIB in Future HEP Projects in Japan

1. Related to the future HEP projects in Japan, there is a need among the HEP community to have a unified HEPLIB environment, because these projects are all international ones.

2. SDC group in Japan decided to utilize CERNLIB because the SDC software at the present stage is based on it.

3. The SDC group in Japan pushed the KEK computer center to support officially CERNLIB, and some amount of the manpower is available now.

4. During the discussions of how to support CERNLIB at KEK, one of the hottest issue was whether it should be done through CERN or SSCL.

Summary

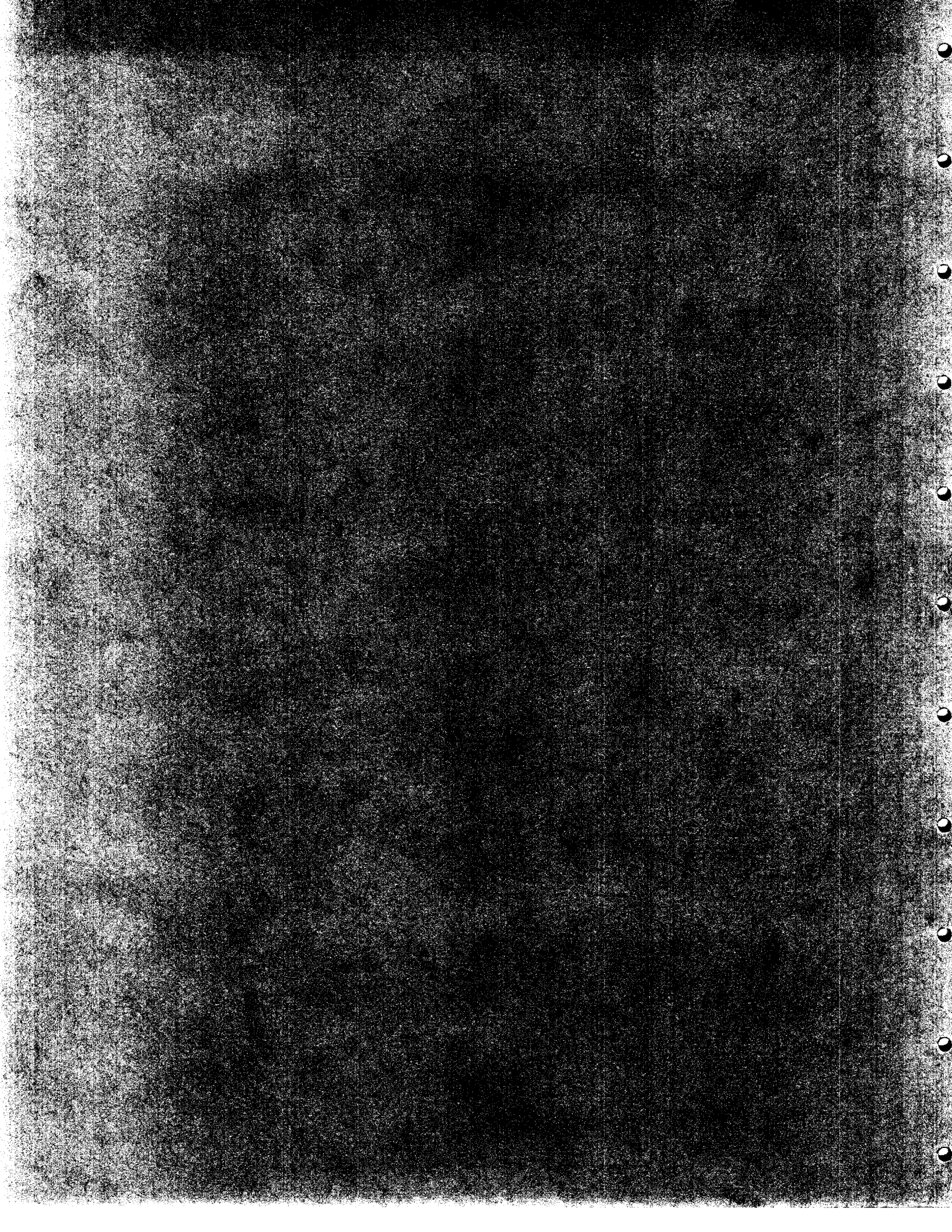
1. In the age of TRISTAN, each experiment group in KEK developed and maintained its own HEPLIB. Therefore, the view point of a unified HEPLIB environment, the HEPLIB environment in Japan was chaotic.

2. Related to the future HEP projects in Japan, there is a need among the HEP community to have a unified HEPLIB environment, because these projects are all international ones.

3. The HEP community in Japan will contribute and support the effort to establish a unified HEPLIB environment if it is organized in concrete way in the present meeting.

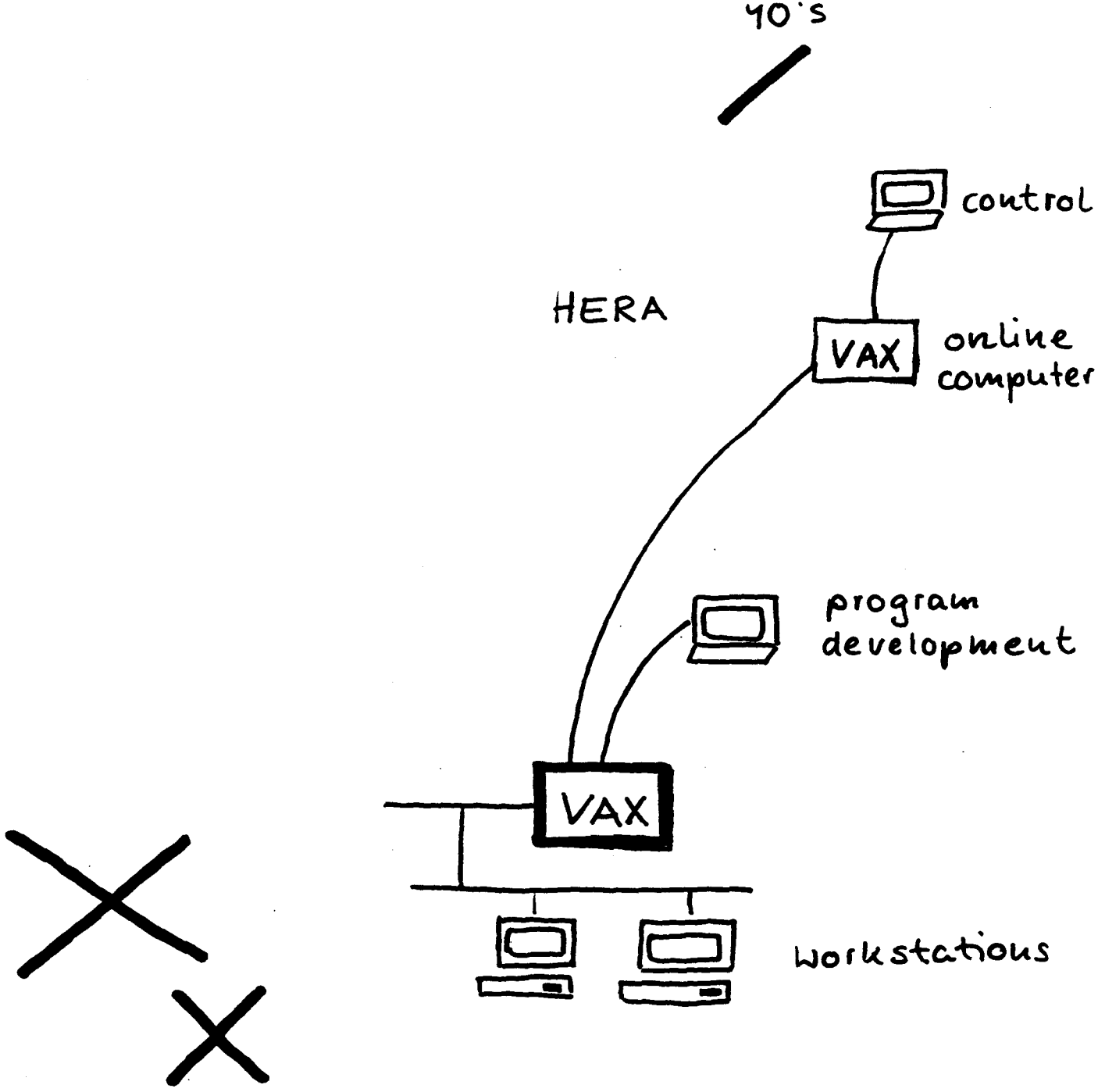
"Overview of Physics Software at DESY"

J.H. Peters



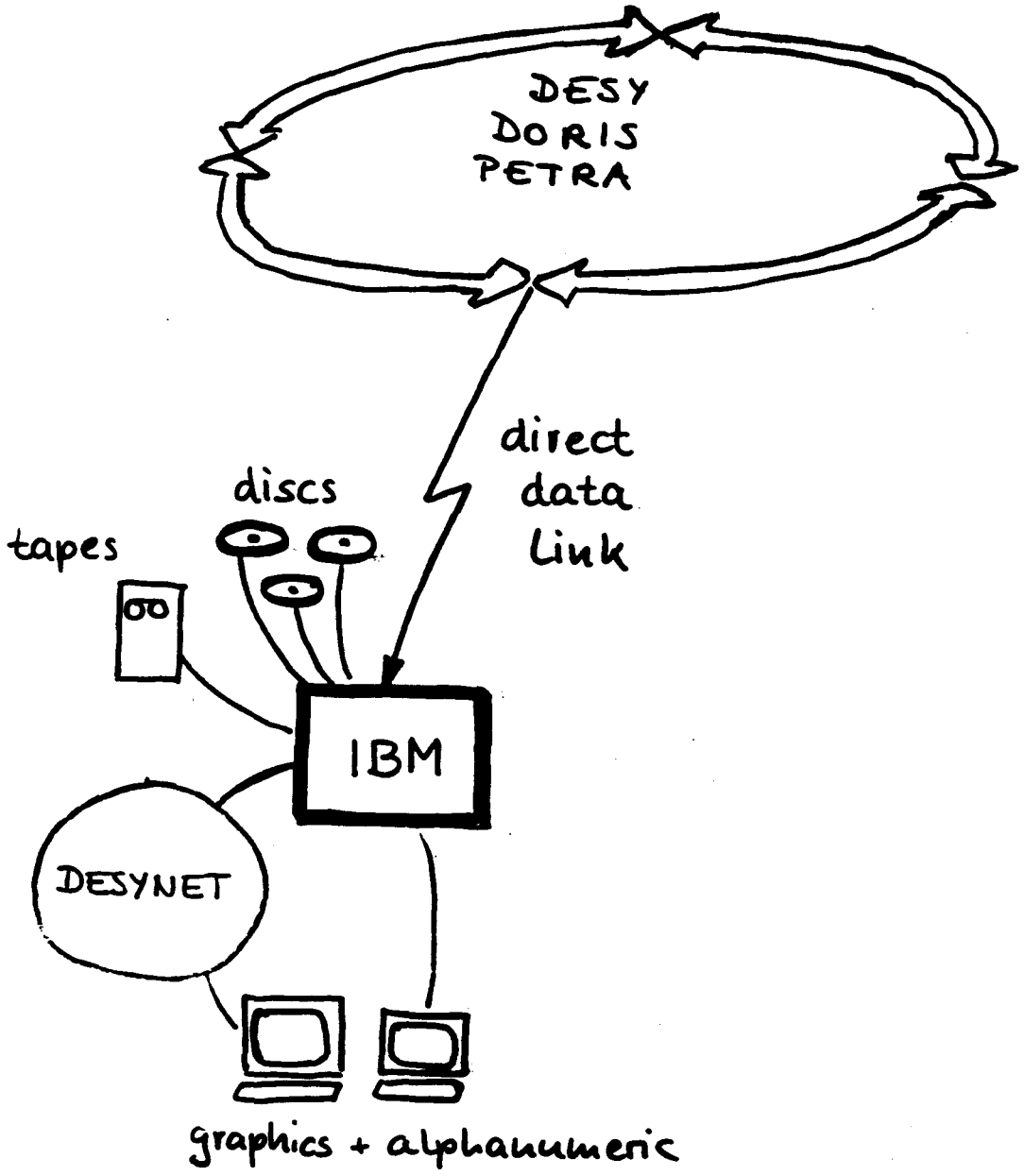
Jan Hendrik Peters
- SSC 19.9.91 -
HEPLIB 91

- **DESY** SOFTWARE
- SUPPORT OF **CERN** SOFTWARE



- program development → VMS
- CPU requirements → RISC
- graphics → Workstations

physics analysis (in the 80's)





hardware / software platforms

DESY / Hamburg

- IBM ES 9000/720 MVS, NENLIB
- VAX Cluster VMS
  - 2 VAX 6000
  - 1 VAX 8350
  - 1 VAX 11/750
  - 4 VAXstations 3100
  - + 1 **VAX 9000** (ordered)
- workstations Unix
  - 6 RS 6000/320
  - 20 HP 9000/425
  - 1 DN 10000
  - 6 DEC 5000
  - 4 Apollo 3000, 3500
  - ~15 VAX 3100
  - + 5 SGI 4D/460 (ordered)
  - + 4 HP 9000/730 (ordered)
  - 10 X-terminals
  - + 80 X-terminals (ordered)

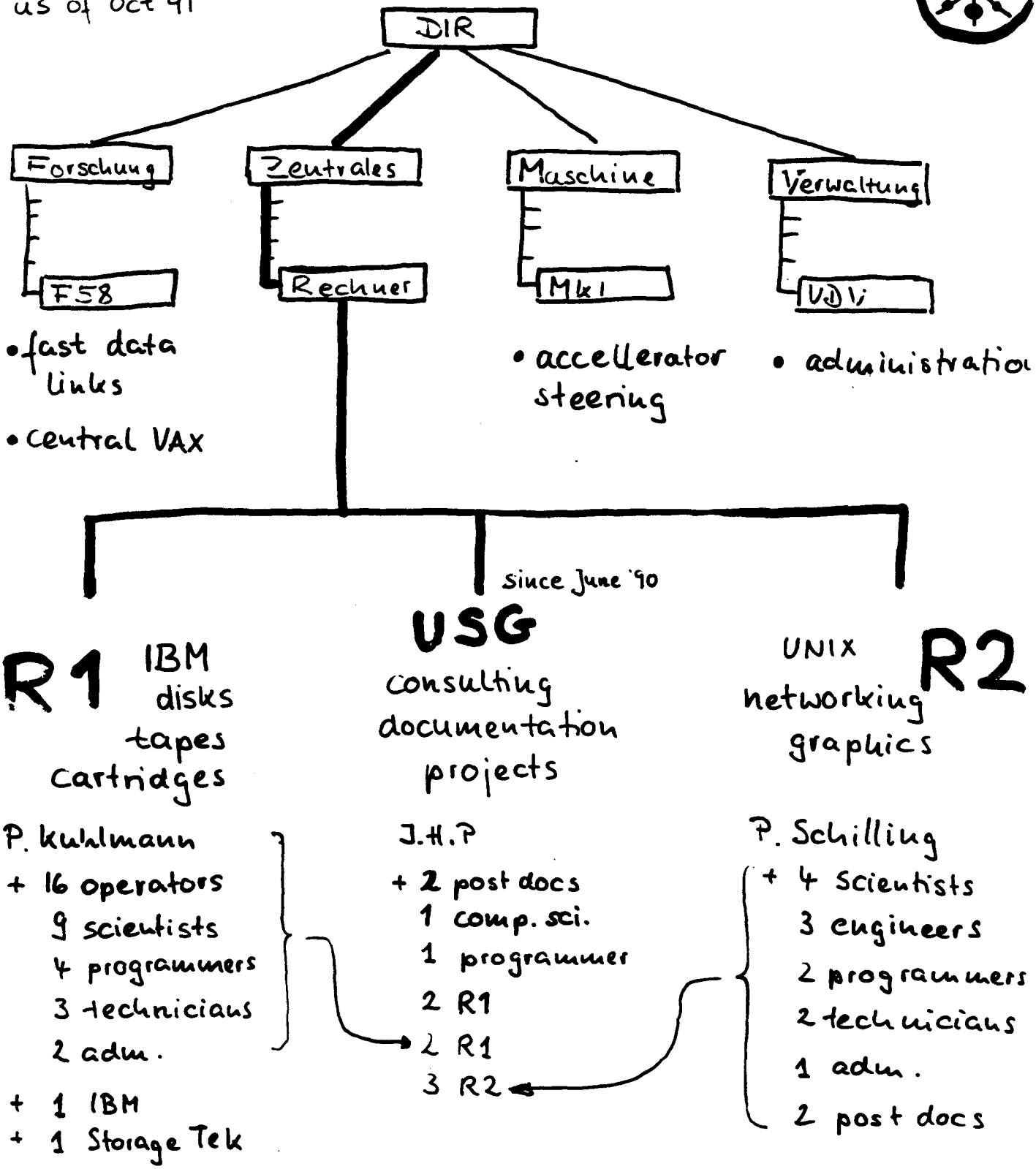
DESY / Zeuthen

- Convex 3210
- VAX Cluster
  - 1 VAX 3100
  - 2 μ-VAX II
- workstations
  - 2 RS 6000/320 + 1 HP 9000/720 + Sun Sparc3

Organisation

as of Oct '91





DESYLIB

- mostly assembler routines (IBM)
  - bit & byte handling
  - timing routines
  - dynamic allocation
  - system interface
  - etc
- some FORTRAN functions, packages
  - fitting function → FITLIB
  - smoothing algorithms
  - BOS → BOSLIB
  - 4-vector manipulation



- ▲ heavily used at DORIS, PETRA experiments
- ▲ not used for HERA software (CERN + expt)
- ▲ split → IBM specifics
  → FORTRAN packages ← expt. software

Software comparison for HERA experiments

	H1	ZEUS
memory manager	BOS	ZEBRA
data manager	DATMAN H1 - DDL	ADAMO ADAMO - DDL
code manager	CMZ	CODEBASE PATCHY
graphics	PAW LOOK	PAW, IDA LACE, GAZE
MC	H1SIM (Geant) H1FS (Superglot)	MOZART (Geant)
I/O package	FPACK	ADAMO / ZEBRA
reconstruction	H1REC	ZEPHYR
analysis	H1PHAN	ADAMO tools
data bases	H1DB (DB2) (FATHEN)	? (FATHEN)

Graphics

- mostly GKS based
  - GTS-Gral (CERN Version)
  - GTS-Gral (original on IBM)
  - DEC-GKS (on VXDESY)

products:

histogramming: GEP, PAW

event display:

- H1: LOOK (all platforms)
- ZEUS: LACE (2d on Unix, Vax)
  GAZE (3d on Apollo, Dec)

- some PHIGS applications
  - speed up
  - 3d hardware rotations
  - → RAL GKS-PHIGS interface

DESYLIB

The DESY program library consists of the following libraries:

FORTRAN dialect	Language	Source library	Load library
IBM VS FORTRAN 77	FVS	R01UTL.DESY.S	SYS1.DESY.LFVS
Siemens FORTRAN 77	F77	R01UTL.DESY.S	SYS1.DESY.LF77
FORTRAN IV	HXE	R01UTL.SOURCE(0)	SYS1.DESY.LHXE

ACCTNR	Get account number from JOB statement
BCDSN	Is DSN a backup data set name ?
BLKLET	Generate block letter text
CALDAT	Calendar date conversion package. Please see
CLEAR	Clear screen
CTOUP (and CTOLO)	Convert text to upper (lower) case
DECODE	Fast unpacking of integers from 16 bit words
DELAY	Delay executing program
DMPOT	Raise a matrix to an integer power
DORGTB	Convert DSORG bits into character format
DSATTR	Provide disk data set attributes
DSNABS	Construct absolute name of a GDG data set
DSNFDD	Find data set name from DDname
DVECSB	Vector manipulation package
DYNAM	Dynamic allocation of files
FNORM	Gauss distributed random numbers
FRECOR, GETCOR	Storage management
FRFONT	Decode text for special font types
FSBEG	Full screen package
GETPAR	Get PARM field data from EXEC statement
HOLID	Calendar with holiday information
HSQRT	SQRT(1 + A) - 1
IDSTL	Addressing of storage obtained with GETCOR
INITFORT	Initialize FORTRAN environment from Pascal
ITRACE	Traceback of routines calling this one
JBWTOR	Stop a job by operator intervention
JOBNAM	Obtain job name and step name in batch job
JUHR	Guard against CPU time overflow
KONTF	Inform about present Quota ("Kontingent")
KVR4CH	Number conversion package
MEDIAN	Mean value and variance of sample
MODELL	Which machine is this job running on ?
MEP	Mean value program
NAPACK	Communication NEWLIB Clist - called program

	NEUTON	Simultaneous non-linear equations
→	NEWLIB	Recursive call on NEWLIB
	QINT	Multiple numerical integration
	QTIMER	Timing information including vectortime
	RD27E	Fast reading of V, VB, or VBS data sets.
	RAN	Two new pseudo random number generators
	RECFTB	Convert RECFM bits into character format
	RGAUSS	Grid points & weights for Gauss integration
	RN	Random number generator
	SOLVE	Least-squares fitting
→	SPFT	Smoothing of noisy data
→	SPLCOF	Determination of spline coefficients
→	SPLDER	Diff./Int. of spline functions
	TIMFAC	CPU time factors for DESY central computers
	TMLOG	Time usage analysis
	UNIFIT	Fitting program
→	VARGET, VARPUT	Communic. NEWLIB Clist - called program
	VECSUB	Four-Vector manipulation package
→	VECTOP	Four-Vector manipulation package
	WHICH	Which machine is this job running on ?
	ZTOBL	Convert hex zero into blank character

+ FIT routines (V. Blobel)

+ GEP graphics editor package (E. Bassler)

+ BOS memory manager (V. Blobel)

+ PCM portable code manager (V. Blobel)

(+ experiment independent code) ?

FPACK I/O package with FP-conversion
self descriptive data (V. Blobel)

Look event display + histogram package
(V. Blobel)

⋮

Look – display program. Font –3 Date 17/12/1990

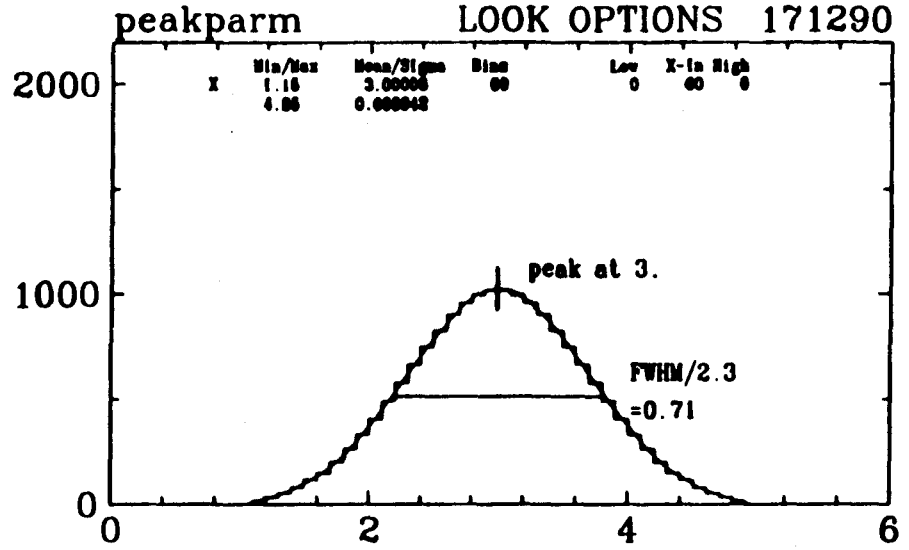


Figure 11 X–projection

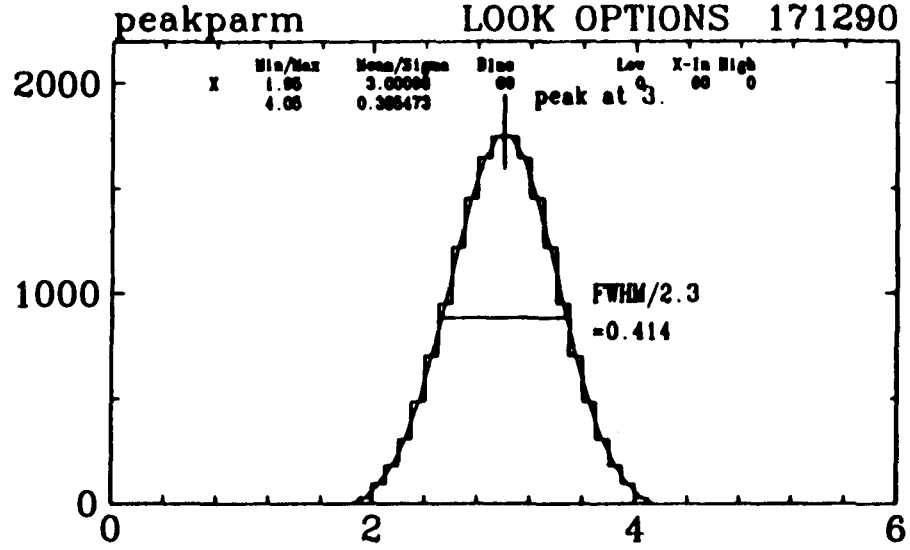


Figure 12 Y–projection

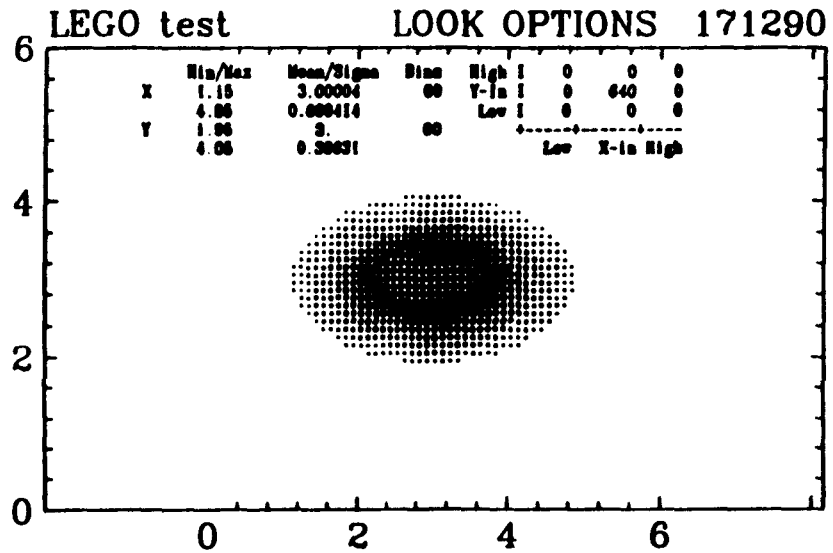


Figure 5 lego 0 arp

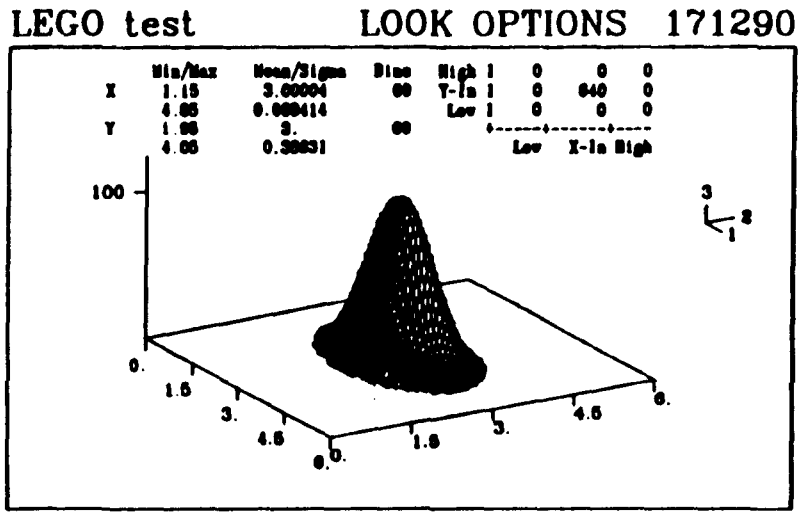


Figure 5 lego 1 4 nogrid

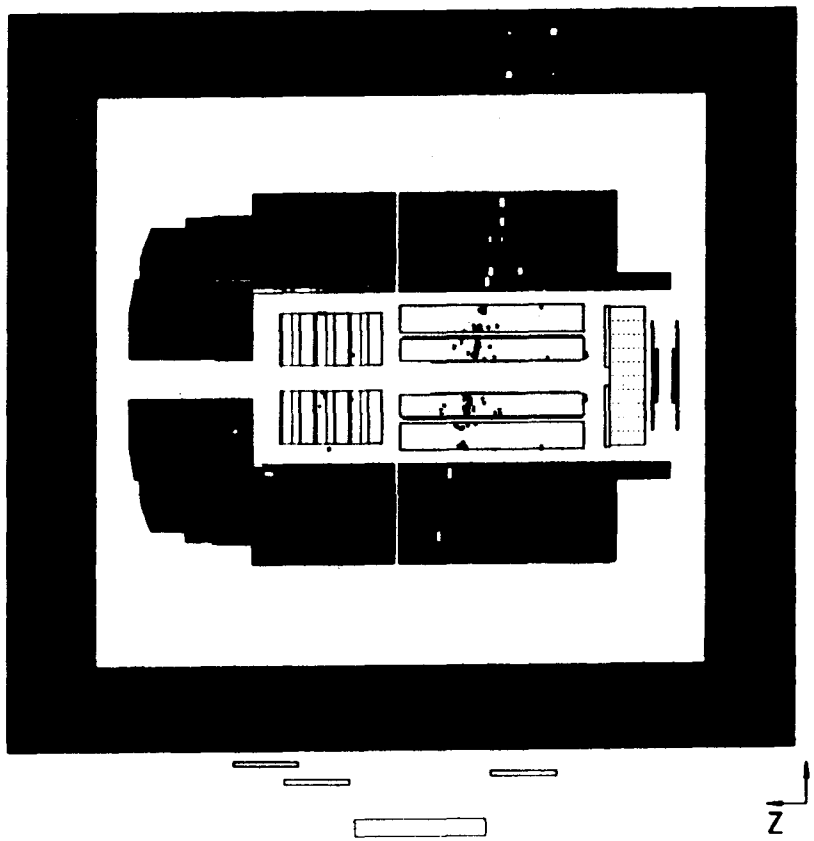
96

100

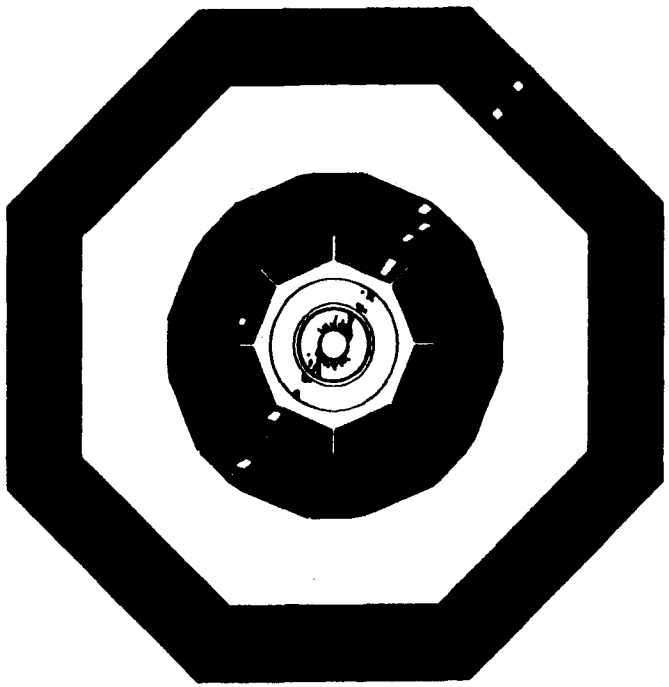
Look – Run number 3648 Event number 112 Date 15/08/1991

H1 Event Display 1.09/06 910416
DSN=HERA03.H1COSM91.RUN03648
Trigger word = 9

E= 0. x 0. GeV H=11.4 kG
Run date 91/ 4/13 1:49



LAr wheels 1 2 3



A4

Fit Programs

R01UTL.FIT.S	source	
R01UTL.FIT.LFVS	load	IBM VS FORTRAN 77
R01UTL.FIT.LF77	load	Siemens FORTRAN 77
R01UTL.FIT.L4	load	FORTRAN IV (HXE)
R01UTL.FIT.DOC	documentation	

Overview on FIT programs

Minimizer

more general:

MINUIT	powerful, many options	MINUIT
VALLEY	simple to use	VALLEY
MLFIT	derivatives not required	MLFIT

special, but faster:

SOLVE	needs derivatives	SOLVE

function fitting

UNIFIT	no user program necessary	UNIFIT

Constrained least squares fitting

CONLES	general package, uses Lagrangian multipliers	CONLES

List of fit programs :

---

1) fit program : MINUIT
what is it? : It is a minimizer, which can be used for least square fit, or maximum likelihood fit. It is very powerful, very sophisticated, has many options, produces lots of printout. You can plot contours of likelihood. You DON'T have to calculate derivatives, (but you may do so - in order to gain speed). Also extended maximum likelihood method can be applied.
documentation : R01UTL.FIT.DOC(MINDOC)
more documentation: R01UTL.FIT.DOC(MINGUIDE)
examples : #MINUIT1, #MINUIT2, #MINUIT3, #MINUIT4 on R01UTL.FIT.DOC

---

2) fit program : VALLEY
what is it? : It is a minimizer, which can be used for least square fit, or maximum likelihood fit. You DON'T have to calculate derivatives and in this respect it is similar to MINUIT. It is simple to use, has however not the many options of MINUIT.



documentation : writeup by V.Blobel. See FIT folder
examples : #VALLEY1 #VALLEY2 on R01UTL.FIT.DOC

---

3) fit program : MLFIT
what is it? : It is a minimizer, which can be used for least square fit, or maximum likelihood fit. In case of maximum likelihood fit it uses poisson statistics. The calculation of derivatives with respect to the parameters is not necessary, but possible, to gain speed.



documentation : writeup by V.Blobel. See FIT folder
examples : #MLFIT1, #MLFIT2 on R01UTL.FIT.DOC
NOTE : GEP is using MLFIT for interactive fitting.

---

4) fit program : SOLVE
what is it? : It is a minimizer, which can be used for least squares fit.
You have to calculate derivatives with respect to the parameters.
documentation : see YELLOW R1-INFO (part C,page 17)

---

5) fit program : UNIFIT
what is it? UNIFIT is doing least square fits , using SOLVE.
This program was written for the occasional fitter who wants to fit simple functions like polynomials, or exponentials, or gaussians.
You can choose among a few functions by giving an index.
You don't have to write the FORTRAN coding of the function you want; you dont't have to calculate derivatives.
UNIFIT also prints a small drawing.
documentation : writeup by D.Lueke,
see Fit folder
or YELLOW R1-INFO (part C,E201,page 19)
examples : #UNIFIT1 on R01UTL.FIT.DOC

---

6) fit program : CONLES
what is it? : Package of programs for constrained least squares fits. It allows up to 20 constraints to a set of up to 50 parameters. The standard technique with Lagrangian multipliers is used, derivatives can be calculated numerically.
Typical applications are kinematical fits.
documentation : writeup by V.Blobel. See FIT folder

CERNLIB activities

- implementation on workstations
  copy tar files CERN → DESY H. Keller
  Apollo, DN10k, HP, SGI Th. Finnern
  DEC 5000, RS6000, Sun system managers
- implementation on VAX
  + ZEUS modifications T. Poser → H. Keller
- development of MVS versions
  PAW, CMZ H. Butenschön
  Fatmen J.H.P.
- coordination with MVS installation at GSI J.H.P.
- implementation of MVS version H. Keller
- documentation J.H.P.

CERNLIB

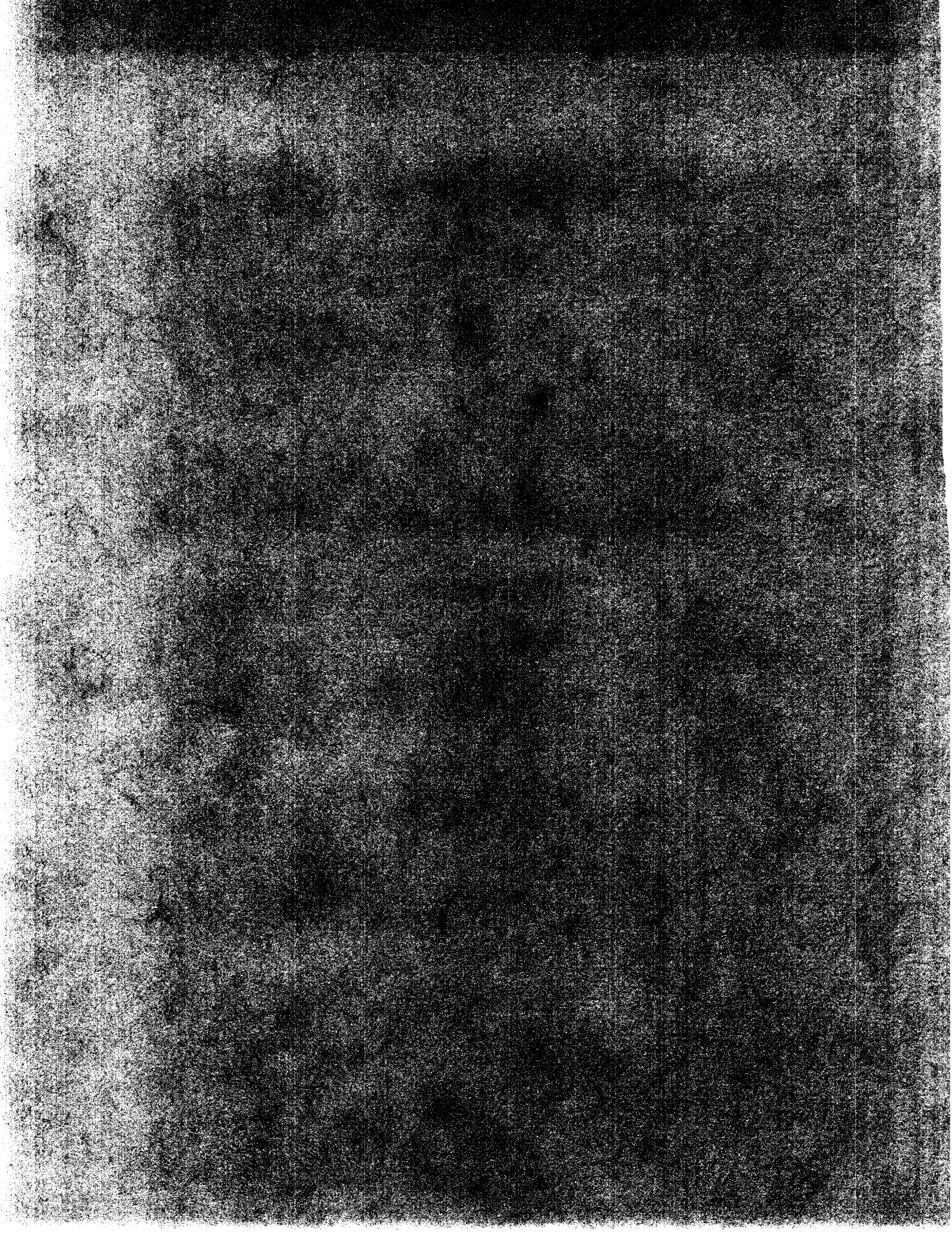
CNL Line	OLD-PRO-NEW Line
Every library corresponds to a CNL version.	From the CNL Line libraries there are copies for 3 generations. Using the following names the user does not need to change his JCL.
"-->" indicates the relation	
RO1UTL.CERN.suffix *)	--> RO1UTL.CERN.OLD.suffix
RO1UTL.CERN.CNL200.LIB.suffix	--> RO1UTL.CERN.PRO.suffix
RO1UTL.CERN.CNL201.LIB.suffix	--> RO1UTL.CERN.NEW.suffix

Versions

- IBM/MVS — modification and adaption + development
- VAX — copy from CERN
- Unix — ftp from CERN, binary files
- [illegible]-DOS — development

"Overview of Physics Software at LBL"

W. Koellner



OVERVIEW OF PHYSICS SOFTWARE AT LBL

Werner Koellner
Physics Division
Lawrence Berkeley Laboratory

HEPLIB'91
SSC Laboratory
Dallas, TX Sept. 1991

Introduction to the LBL Environment

A non-single-purpose Lab

General Computing Support: Available primarily for DEC and SUN platforms

Project-specific Support: Available at desired or required levels only via recharging of costs.

Software Support:

General Software

Graphics: GKS (ATC)
DI3000
DISSPLA
TELLAGRAPH
NCAR

Math: IMSL
NAG
MATHEMATICA

Visual: AVS

HEPLIB'91
SSC Laboratory
Dallas, TX Sept. 1991

Software Support: (cont'd)

Research-oriented Software

provided by individual divisions and/or groups,

e.g. CERN Libraries

Genome Research

Physics Division Computing Management

semi-independent

loosely coordinated with the Computing Science Division

own software management and support

own software and software tools development

own User Account management

own disk management, own archival backup

renting 4 RA 90's @ 1.2 GBytes

5 SI's @ 1.7 GBytes

(total 13.4 GBytes)

shared own workstation management

25 VAXStations

75 Macintoshes

16 IBM PC's

17 SUN Workstations

some others

Physics Software at LBL

manpower: 1 1/2 FTE management/programming

occasional temp. help

Research Staff programming support

"imported": CERN Libraries (VAX/SUN)

pre-releases (no pro vers)

feedback to CERN

support for ATC GKS

PATCHY, CMZ (VAX/SUN)

CERN-maintained/distributed

HEP Software (VAX/SUN)

other utilities: UGS

TOPDRAWER

EGS

FOR_STRUCT

SWING

MORTRAN

HEPLIB'91
SSC Laboratory
Dallas, TX Sept. 1991

locally developed (and exported)

DISPLAY

DISPLAY5 (with D0 support)

MMS/CMS custom interface

CMS action-triggered

rebuild of Obj Libraries

Coupled with IRIS

platform-independent MAKE

IRIS (Information Retrieval)

JY411

CALC

project-specific: customized EDFOR

TPC Libraries

D0 Libraries

CDF Libraries

SDC Libraries (VAX & SUN)

NEWS facilities (D0, TPC, SDC)

HELP facilities (VAX HELP)
(SUN man entries)

current/new development:

B-Factory software and software tools development

DISPLAY5 port to UNIX
use HIGZ

Modular and machine-idependent Monte Carlo

NEWS OF PHYSICS LIBRARIES

Werner Koellner

● GENERAL INFORMATION

Object libraries, source files, and other files and procedures thought to be useful are being maintained at varying levels. Various CERN "libraries" make up the major part of this collection. In general, the newest releases or pre-releases are offered as default versions for general use. Please let me know if some package, which may be of substantial interest, is not available.

● CERN LIBRARY PROBLEM HANDLING

Users are encouraged to report problems or questions regarding CERN libraries, by writing to one of the following discussion lists, or to me (WOKoellner@lbl.gov):

LPAW@CERNVM.BITNET .....(about PAW)

LGEANT@CERNVM.BITNET .....(about GEANT)

HEPLIB@CERNVM.BITNET .....
.....(about CERN Library codes)

You may also subscribe to any of these discussion lists by sending an electronic mail message containing the single line

```
SUBSCRIBE <list> <your full name>
```

(list being one of the above) to

```
LISTSERV@cernvm.cern.ch
```

Copies of the various discussion mails are available in CERN$INFORM:

LGEANT.LOGyymm,
LPAW.LOGyymm,
LGEANTyymmdd*.NOTE,
LPAWyymmdd*.NOTE, and
HEPLIByymmdd*.NOTE.

Your problem may be among these discussed.

● CERN LIBRARY USER LISTS

Users who wish to be alerted whenever I rebuild the default GEANT or PAW Libraries or update other CERN Libraries may register by sending me a request.

● WHAT'S AVAILABLE

Some or all of the following packages are available on supported platforms. Specific information may be displayed online by typing

HELP <package name>

(on CSA) or via available man entries (on Sun or STARDENT), or by typing

HELP @PHYSICS_UTILITIES

(on CSA) and choosing the desired subtopic. Additional information, particularly regarding CERN packages, may be found in the CERN$INFORM directory area.

CERN LIBRARIES:

CMZ	Code Maintenance
COJETS	pbar-p Monte Carlo
DZEDIT	Zebra Bank Doc./Display System
GARFIELD	Drift Chamber Simulation
GEANT	Detector Design
EURODEC/JET	pbar-p Monte Carlo
GENLIB	General Library
GRAFLIB	Graphics Interface Package
HBOOK	Histogram Package (in PACKLIB)
HERWIG	hadron Monte Carlo
HPLOT	Plotting Package (in GRAFLIB)
ISAJET	pbar-p Monte Carlo
JETSET	Lund Monte Carlo
KERNLIB	General Library
LUCIFER	Lund Monte Carlo
MINUIT	Fitting (in PACKLIB)
PACKLIB	General Library
PATCHY	Code Maintenance
PAWLIB	Physics Analysis
PDFLIB	Parton Density Functions
TWISTER	Lund Monte Carlo
ZEBRA	I/O & Memory Mgt. (in PACKLIB)

FILE TRANSFER:

ZFTP	Transfer between Sun, VAX, IBM
TELNETG	HIGZ Graphics on remote hosts

OTHER PACKAGES:

CALCULATOR	Fancy HP Calculator
DISPLAY(5)	HBOOK/HPLOT Histogr. Manipulation
FOR_STRUCT	Source Code Structuring
EGS	e+e- Monte Carlo
JY411	CAMAC Drivers
MORTRAN	FORTRAN Preprocessing
PROBE	Examine Object Libraries
SWING	Directory Management
TOPDRAWER	Plot Processing
UGS	Unified Graphics Package

ON UX5 and STARDENT

• NEWS

DZEDIT, a new ZEBRA Bank Documentation and Display System, has been installed. See **man dzedit**; for a user manual see Ed Sheena (x5176).

New versions of the ZEBRA manuals "FZ" and "DIA" have been received. Please see Ed Sheena (x5176) for a copy.

The effort to support GEANT (and PAW), with X11 graphics on STARDENT is still in progress.

• LAST MONTH

✔ GEANT and CSPACK Manuals in LaTeX format
✔ Parton Density Function Library (see **man pdflib**)
✔ PATCHY, etc., installed (see **man patchy**)
✔ ZEBRA File Transfer Utility, ZFTP (see **man zftp**)

• GENERAL INFORMATION

More detailed information about the maintenance status of all libraries can be found in the CSA section.

Selected CERN Libraries are available on UX5 and on the STARDENT computer, and may be copied to Sun SPARCStations.

Currently, the following man entries serve to give details about status and use of available packages:

man cernlib
man dzedit
man geant
man herwig
man isajet
man jetset
man minuit
man patchy
man paw
man pdflib and
man zftp

The path to the CERN library area on UX5 and STARDENT begins with

/home/ux5/ux5c/phyd/cern

All files that are common to both UX5 and STARDENT are downstream from subdirectory "sun" while those unique for STARDENT are downstream from subdirectory "stardent". I suggest that you set the environment variable CERN_ROOT:

setenv CERN_ROOT /home/ux5/ux5c/phyd/cern

Then you find all files in

$CERN_ROOT/sun/*.

Versions of files that are different for the STARDENT are in

$CERN_ROOT/stardent/*.

In linking with any library, just specify

-l<library>

where **<library>** is one of the strings listed under "LIBRARY" below.

On Sun machines, ATC-GKS and X11 are the supported graphics interfaces for CERN programs with graphics. Information about linking with these graphics packages can be gleaned from

man paw or
man geant.

The following libraries are available in

$(CERN_ROOT)/sun/lib.

Most of these are also available in

$(CERN_ROOT)/stardent/lib.

LIBRARY	PACKAGES
dzdoc	dzdoc (for dzedit)
geantlib	geane, geang, geanh, geant, geanx
genlib	gen
graflib	hplot5, higz, gkspack
herwig	webber LUND Monte Carlo
isajet	p-p, pbar-p Monte Carlo
jetset73	LUND Monte Carlo (Jetset73 + Pythia55)
kernlib	kerngen, kernnum
minuit	minuit
packlib	cspack, epio, ffread, hbook4, iopack, kapack,kuip, minuit, zbook, zebra, zcedex
pdflib	pdf
pawlib	paw, comis, sigma

ON CSA

• NEWS

DZEDIT, a new ZEBRA Bank Documentation and Display System, has been installed for both GKS or X11 graphics. For a user manual see Ed Sheena (x5176).

New versions of the ZEBRA manuals "FZ" and "DIA" have been received. Please see Ed Sheena (x5176) for a copy.

Recently installed:

CSPACK Version 1.19/05 ....(in CERN$PACK_LIB)
COMIS Version 1.11/04 ....(in CERN$PAW_LIB)
FFREAD Version 3.10/02 ....(in CERN$PACK_LIB)
HBOOK4 Version 4.12 ....(in CERN$PACK_LIB)
HERWIG Version 5.03 ....(in CERN$HERWIG_LIB
HIGZ Version 1.12/16 ....(in CERN$GRAF_LIB)
HPLOT5 Version 5.10/13 ....(in CERN$GRAF_LIB)
KERNBIT Version 1.06/06 ....(in CERN$KERN_LIB)
KERNFOR Version 4.30 ....(in CERN$KERN_LIB)
KERNNUM Version 3.06/01 ....(in CERN$KERN_LIB)
KUIP Version 1.66/06 ....(in CERN$PACK_LIB)
MINUIT Version 2.05/01 ....(in CERN$PACK_LIB)
PAW Version 1.11/04 ....(in CERN$PAW_LIB)
SIGMA Version 1.06 ....(in CERN$PAW_LIB)
UMON Version 1.01/01 ....(in CERN$KERN_LIB)
ZBOOK Version 2.23/01 ....(in CERN$PACK_LIB)
ZEBPACK Version 1.00/04 ....(in CERN$GRAF_LIB)
ZEBRA Version 3.67 ....(in CERN$PACK_LIB)

• LAST MONTH

✔ Library release CNL201 or newer
✔ GEANT and CSPACK Manuals in LaTeX format
✔ Parton Density Function Library
✔ PAW in "no-graphics" mode
✔ PATCHY, etc., installed
✔ ZEBRA File Transfer Utility, ZFTP

• GENERAL INFORMATION

On the CSA Cluster and on LAN workstations, if the required disks are mounted you can access the Physics Utilities, including the CERN Library, the PAW (Physics Analysis Workstation) Library, and the various Physics Utilities HELP Libraries by executing the DCL command

```
@Physics$Manager:Setup_Phys
```

We recommend that you include this line in your Login.Com file.

CERN libraries are updated at unpredictable times. Changes are documented in the "Program Library News" section of the CERN Computer Newsletter. Past, current, and sometimes future issues can be found in **Cern$Inform:PROGLIB.CNLxxx**. Of particular interest are news regarding the status of obsolete routines. In some cases a previous version of an object library is saved as **xxxxx.OLD**.

The recommended method to access the latest standard object libraries is to use logical names **CERN$*_LIB**. For these names and other details please see the help text in

```
Help Cern
```

On the CSA cluster, the supported graphics interfaces for CERN programs with graphics are ATC-GKS, X11, and DI3000 (on CSA2 only).

• GEANT

GEANT 3.14 was released in November 1990. The default object library is always built by using newly released PATCHY correction cradles. The unmodified library is **Cern$Library:Geant314.Olb**.
The latest GEANT changes are noted in **CERN$INFORM:GEANT_CORR.HISTORY**.

• X11 GRAPHICS

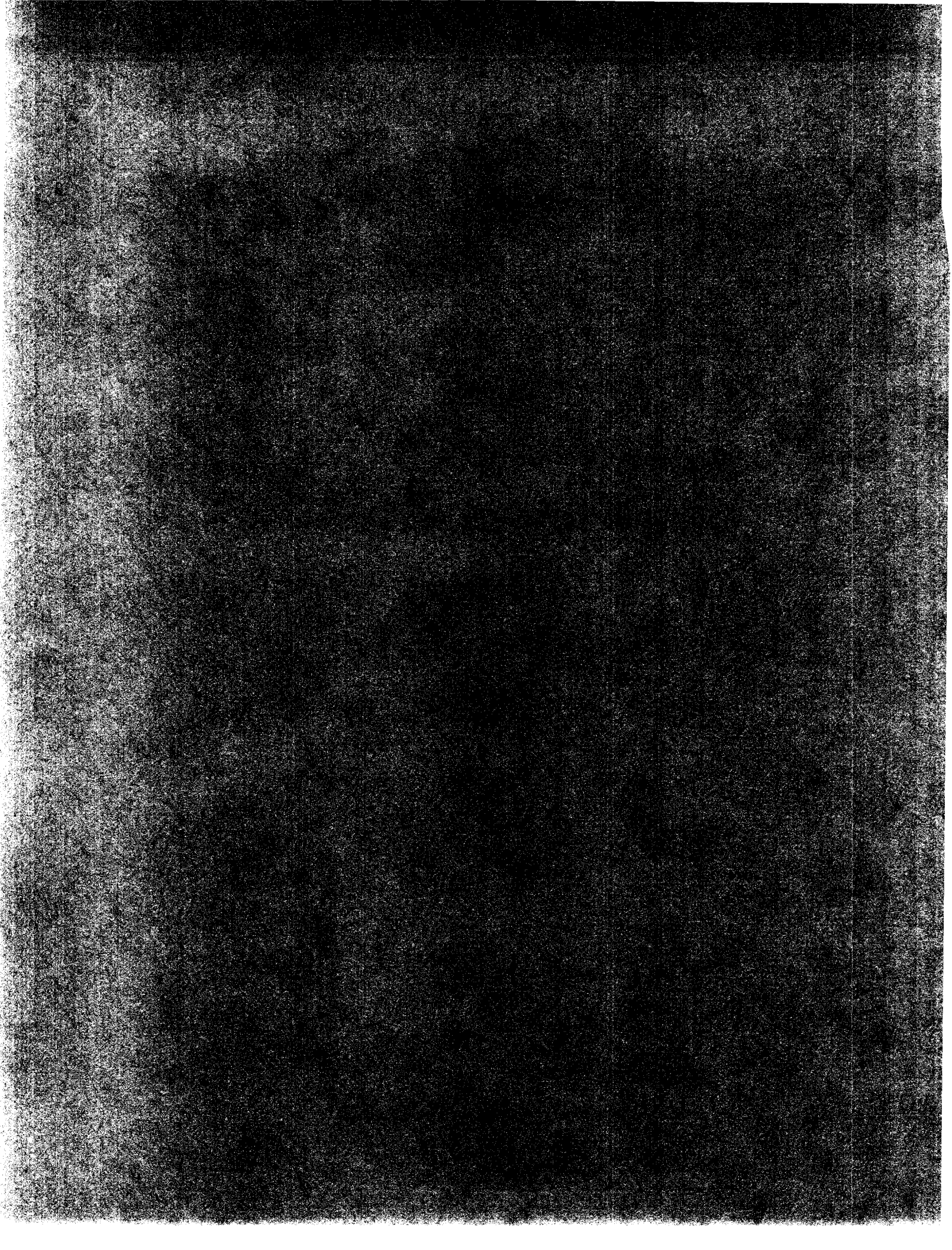
Instead of using ATC-GKS drivers for GEANT and PAW graphics displays, you can use X11 drivers, which may result in faster display tasks. You may run **PAW$LIBRARY:PAW_X11**, or may link your **GEANT.EXE** with X11 drivers. If you use **Cern$Library:Geant.Lnk** to link your Geant, the first question will let you specify X11. Note that prior to running an X11 program from a non-X11 device, e.g., from a VT240 terminal, you may need to SET DISPLAY ... Please see the LBL Computing Newsletter, Aug. 1990, Pg.4, for details.

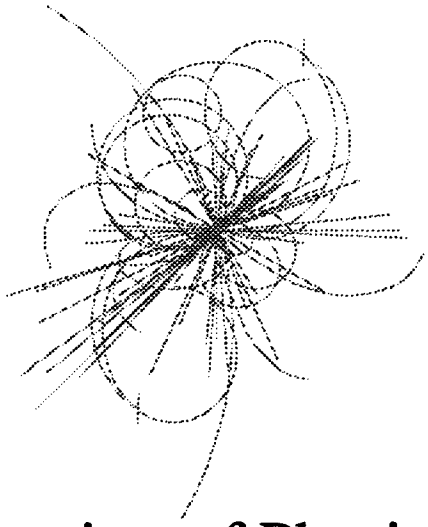
Forward comments and questions to me at x4398, or

UNIX or Software Tools Mail: WOKoellner@lbl.gov
VMS Mail: lbl::WOKoellner

"Overview of Physics Software at SSC"

H. Johnstad





Overview of Physics Software at SSCL

presented at

The HEPLIB'91 International Users Meeting on the Support and Environments of High Energy Physics Computing

SSC Laboratory, Dallas, Texas
September 19-20, 1991

Harald Johnstad
Physics Research Division
Superconducting Super Collider Laboratory*
2550 Beckleymeade Avenue
Dallas, Texas 75237

*Operated by the Universities Research Association Inc., for the U.S. Department of Energy under Contract No. DE-AC35-89ER40486

Overview of Physics Software at SSCL

Harald Johnstad

1 COMPUTERS

> PDSF (SUN/SGI)

> VAX

> Other Mainframes

> Intel i860

> Workstations:

Apollo DN1000

DECstation

HP720

IBMRS6000

SGI

SUN

2 WHO NEEDS SUPPORT

- SDC Collaboartion
- GEM Collaboration
- Non-SSC experiments (CDF,...)
- Base program physics support
- SSC Library

3 SOFTWARE

> CERN Program Library

> Code management tools (PATCHY, CMZ)

> Physics generators (LUND, ISAJET,...)

> Accelerator software (MAD, TRANSPORT,...)

> Mathematical libraries (IMSL, NAG,...)

> Graphics (X11, GKS, GL,...)

> SCRI Software (TYPES)

> Detector simulation (GEANT, other?)

> GEM GEANT Applications

> SDC SHELL

> SDC Parallel SHELL and User Interface

4 DIRECTORY STRUCTURES

> Similar procedures on different systmes: VAX, UNIX,...

> Multi-version support (Old, Current, New, Debug,...)

> Subdirectories (Source, Object, Libraries, Executables, Manager, Documentation, Help, etc...)

5 INITIALIZATION

> VAX/VMS - SETUP CERN/CNL203

> UNIX - initcern.csh

```
cd /ssclib
/home/sscux1/physics/apo10k/lib
/home/sscux1/physics/decs/lib
/home/sscux1/physics/ibmrt/lib
/home/sscux1/physics/sgi/lib
/home/sscux1/physics/sun/lib
```

6 MAINTENANCE

> Multi version maintenance

> Multi version maiantenance

> Pre-release testing

> Compare all local modifications with original with automztic tolls, create correction PAM files, submit to CERN

7 DOCUMENTATION

> CERN documentation printed at CERN

> Special agreement for transportation

> Minimal reproduction at SSCL as needed

> UNIX man pages

> Maintenance of LISTSERV discussion Lists

> Hardcopy of LISTSERV discussion Lists

8 DISTRIBUTION, SUBSCRIBERS

> SSCL Computer Science Library

> Distribution of software libraries

> Distribution of documentation

9 USE OF LISTSERV AT CERNVM

> All CERNVN Lists

> Automatic file distribution (when available)

> Automatic file update information

> Maintain NOTEBOOKs in public areas

> Local access procedures

10 USE OF LISTSERV AT UTDALLAS

> Information to SSC related experiments

> Distribution of SSCL software

> HEPLIB related information

11 MANPOWER

> Basic support

> Coordination

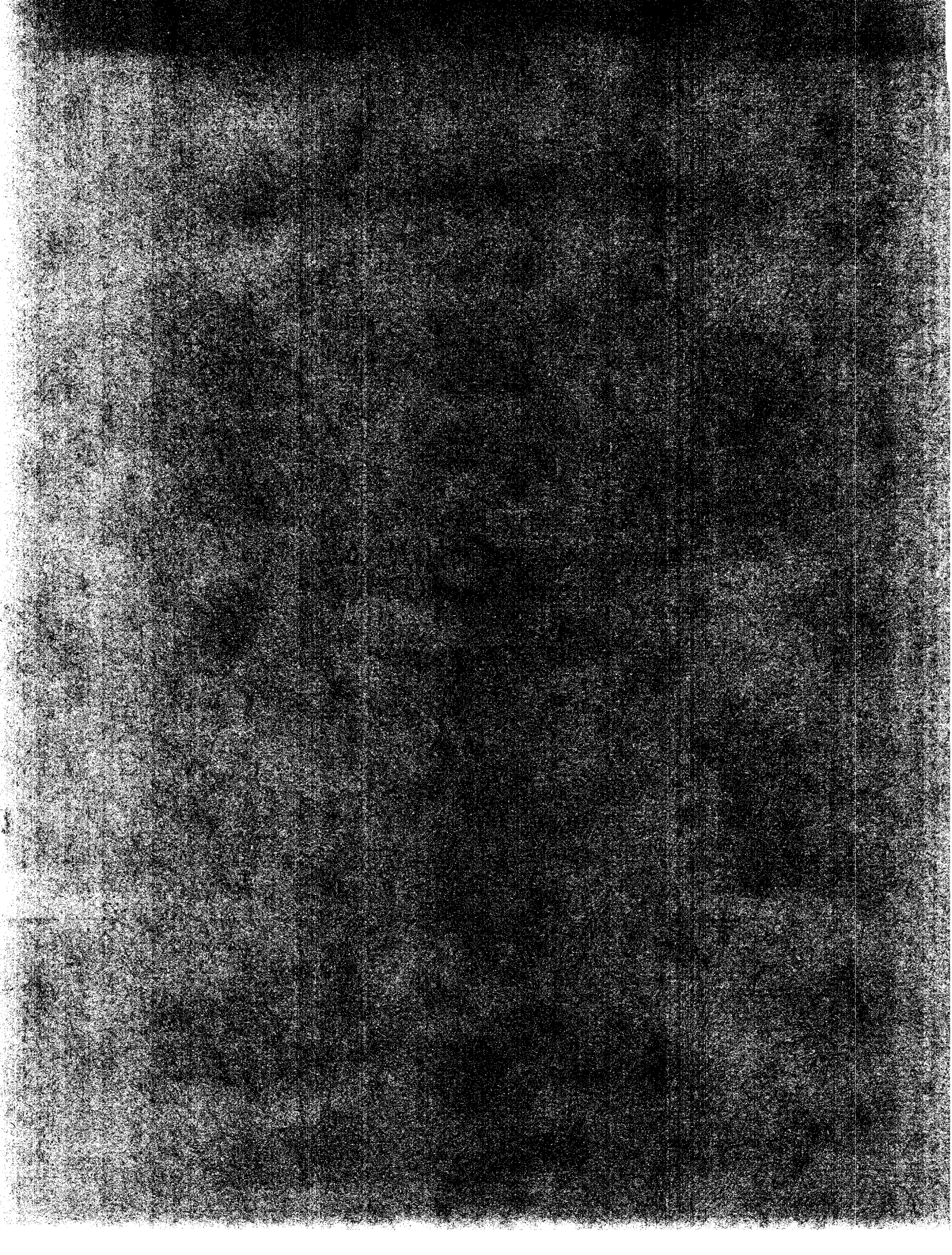
> Physicists

> Computer Scientists

> Computing Division

"SLD Interactive Physics Analysis"
"Initial Experience with XMotif"
"Comments on GEM"

T. Johnson



SLD Interactive Physics Analysis

Motif Interactive Data Analysis Shell
(MIDAS)

Tony Johnson

Boston University/SLAC/SLD/GEM

19 September 1991

Topics to be discussed:

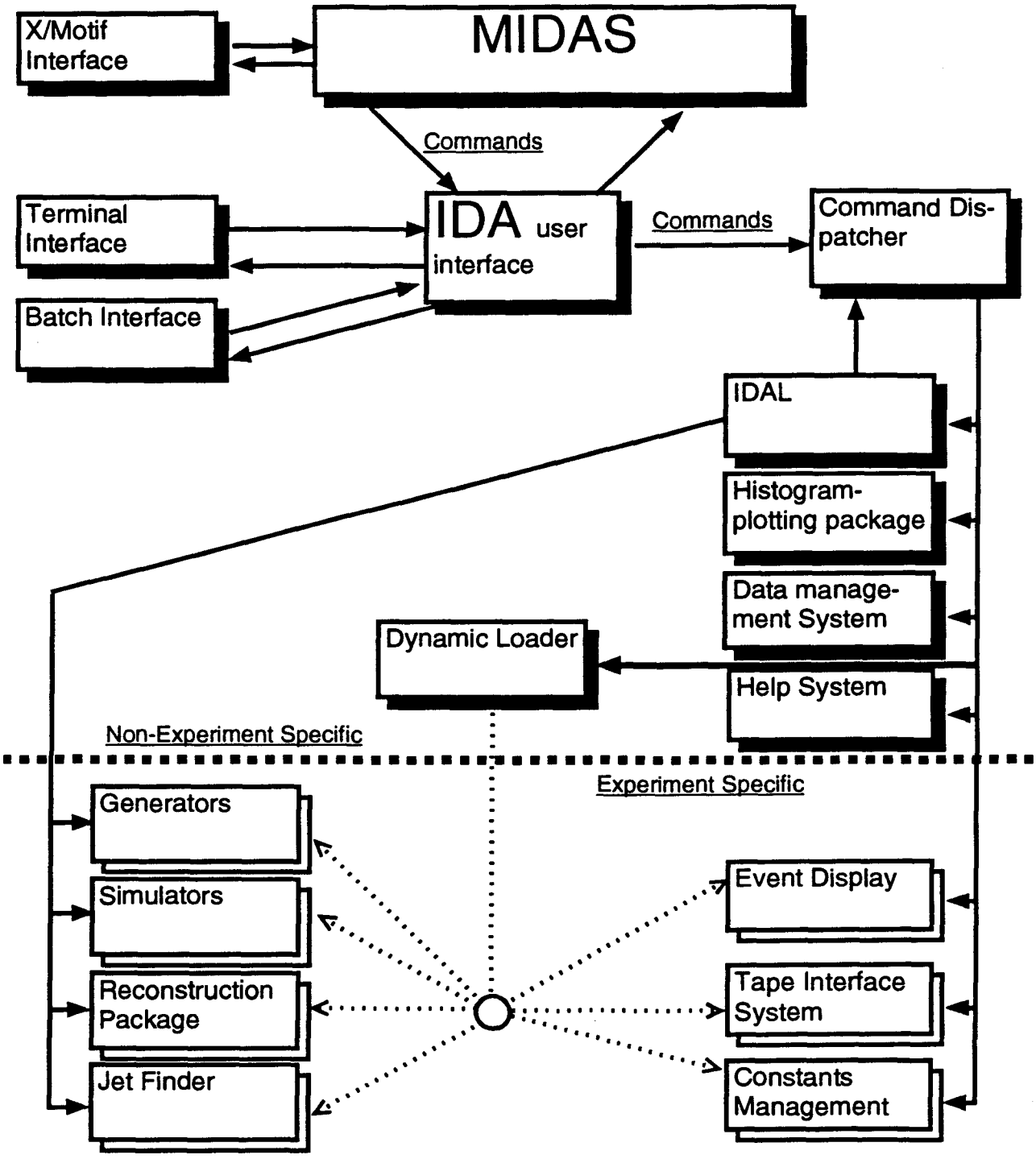
1. What is Interactive Data Analysis.
2. History of Interactive Data Analysis (IDA) at SLAC.
3. Why use an Interactive Data Analysis Shell for entire MC-Reconstruction-Analysis Environment.
4. Design and Development of a Data Management System for use in a fully Interactive Environment (Jazelle).
5. The IDAL (interactive physics analysis language).
6. User experience with SLD IDA.
7. Integration of 3-D Event display into interactive environment.
8. Extension of IDA environment to use Motif/X (MIDAS).

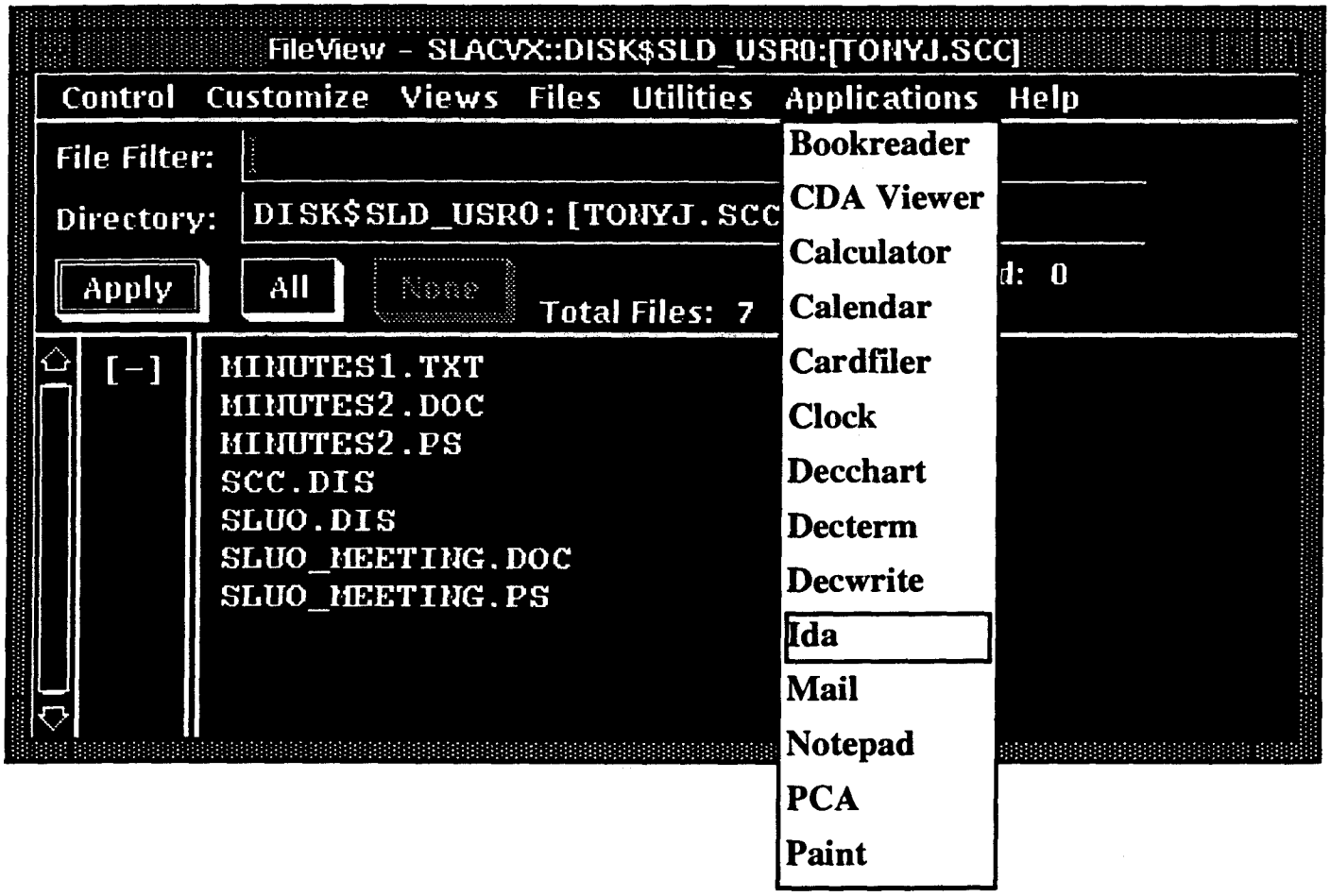
.....if I had 2 hours.

What will <u>really</u> be discussed:

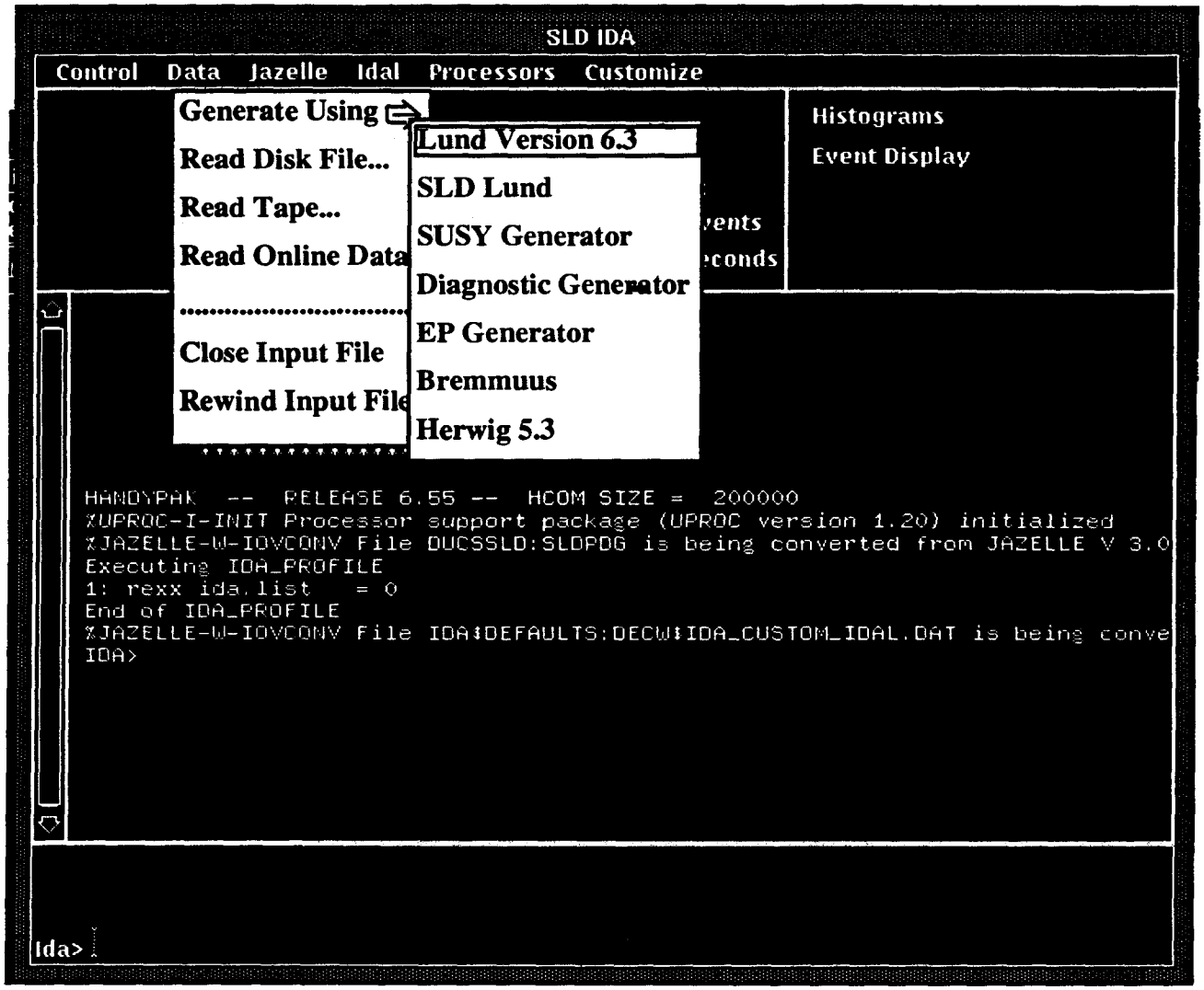
1. Very Brief Introduction to IDA/MIDAS.
2. Run through a "typical" MIDAS session.
3. A few comments on implementation issues.
4. Conclusion - Why discuss this at a HEPLIB meeting?

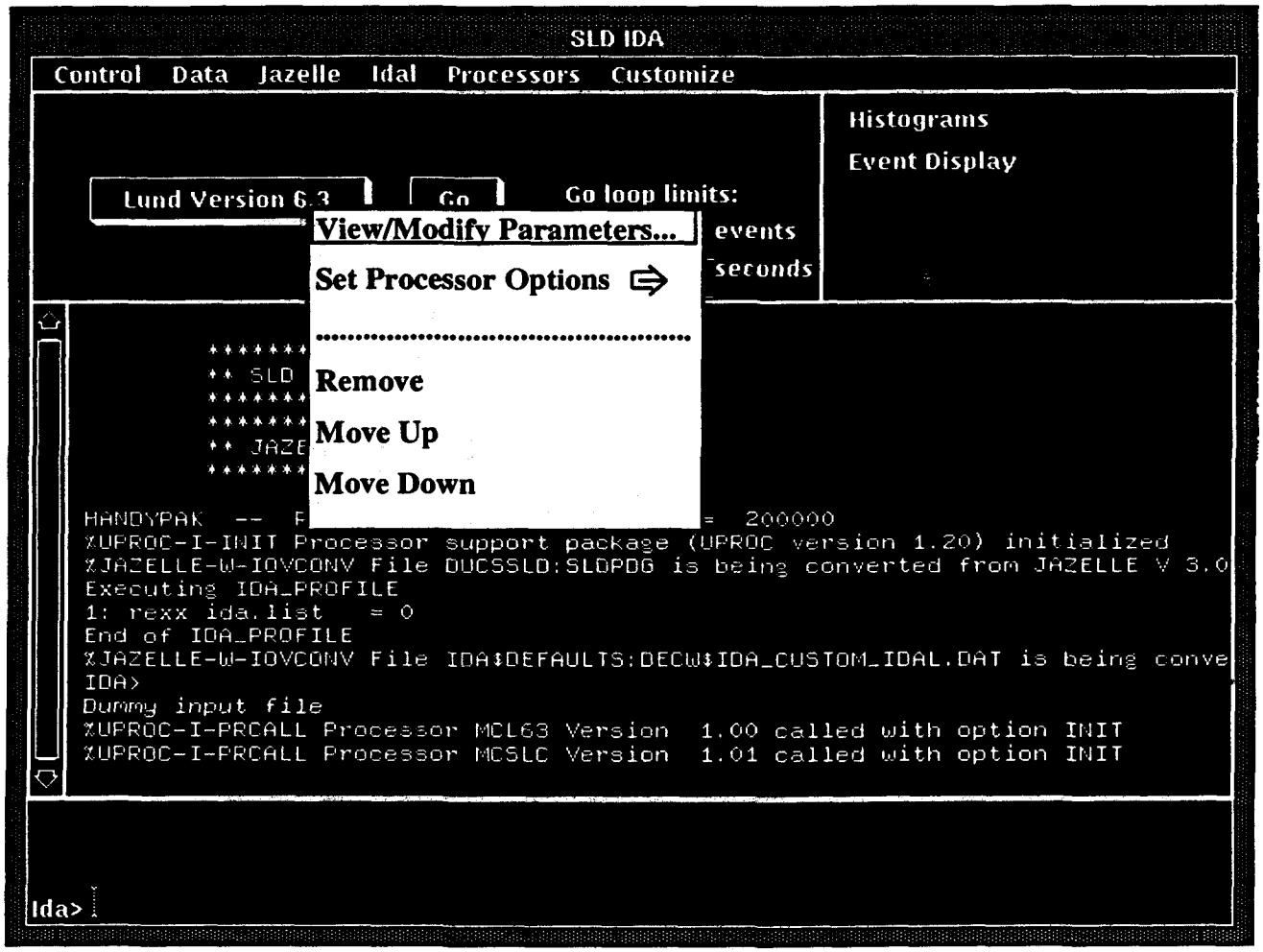
System Overview:

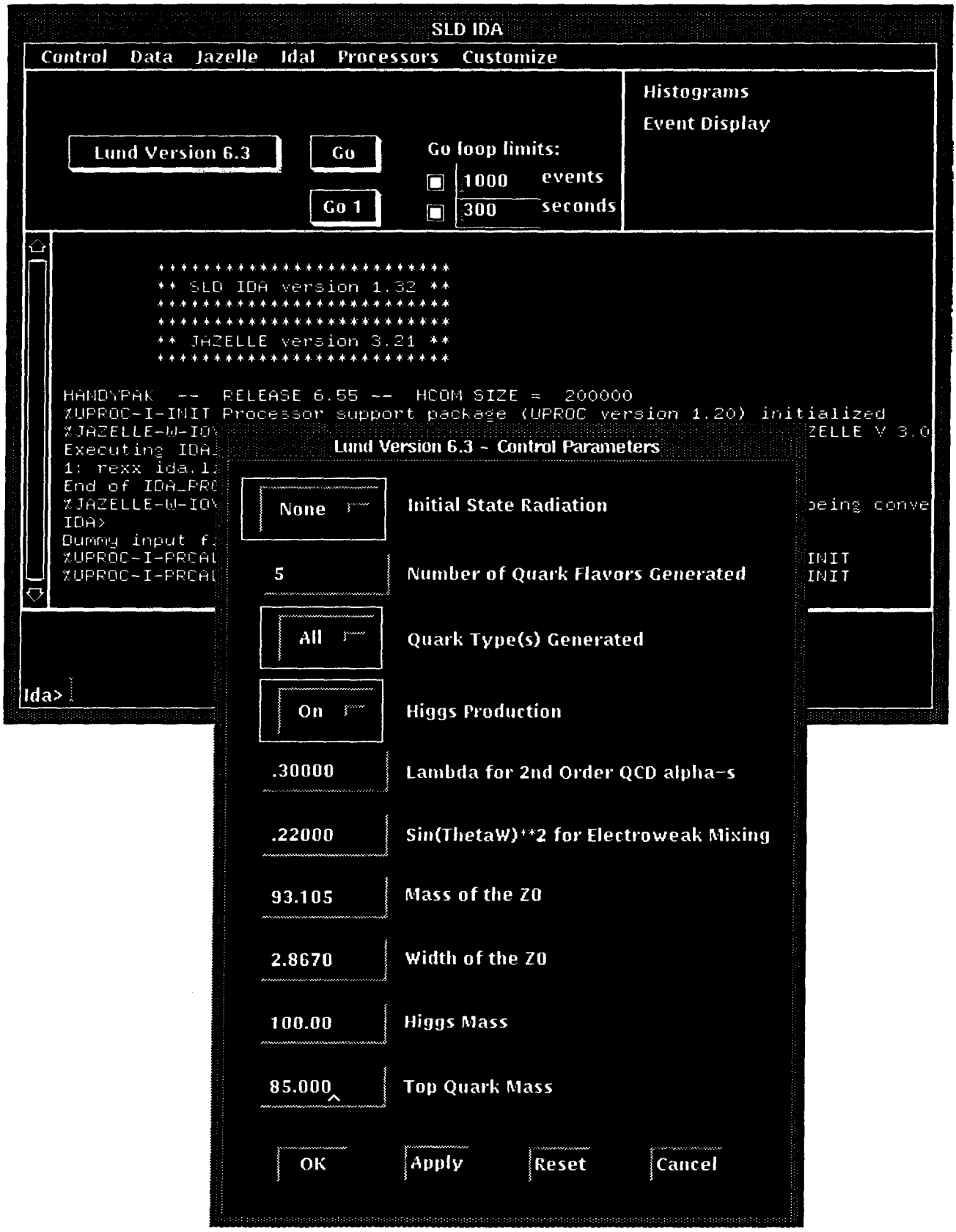


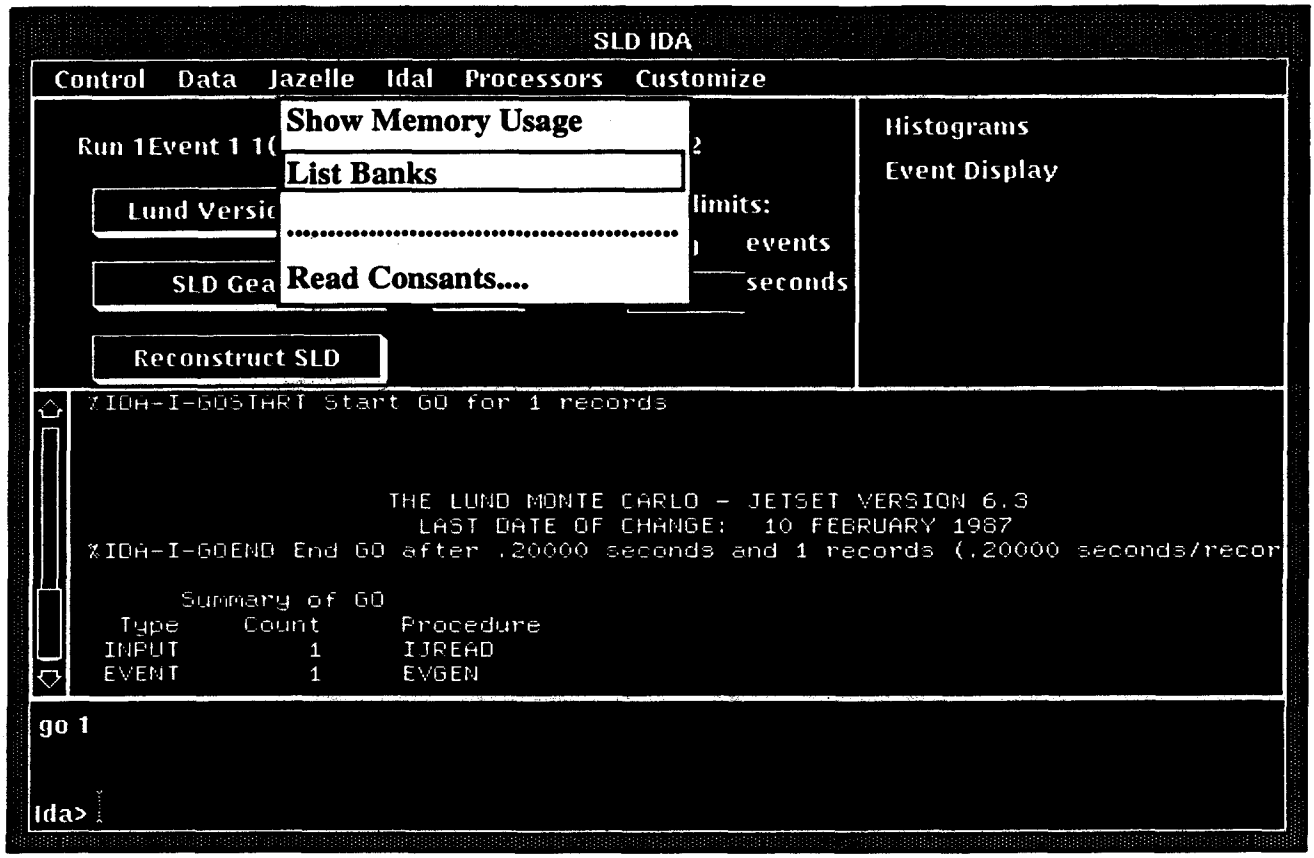

FileView - SLACVX::DISK$SLD_USR0:[TONYJ.SCC]
Control Customize Views Files Utilities Applications Help
File Filter:
Directory: DISK$SLD_USR0:[TONYJ.SCC
Apply
All
None
Total Files: 7
d: 0
[-]
MINUTES1.TXT
MINUTES2.DOC
MINUTES2.PS
SCC.DIS
SLUO.DIS
SLUO_MEETING.DOC
SLUO_MEETING.PS
Bookreader
CDA Viewer
Calculator
Calendar
Cardfiler
Clock
Decchart
Decterm
Decwrite
Ida
Mail
Notepad
PCA
Paint

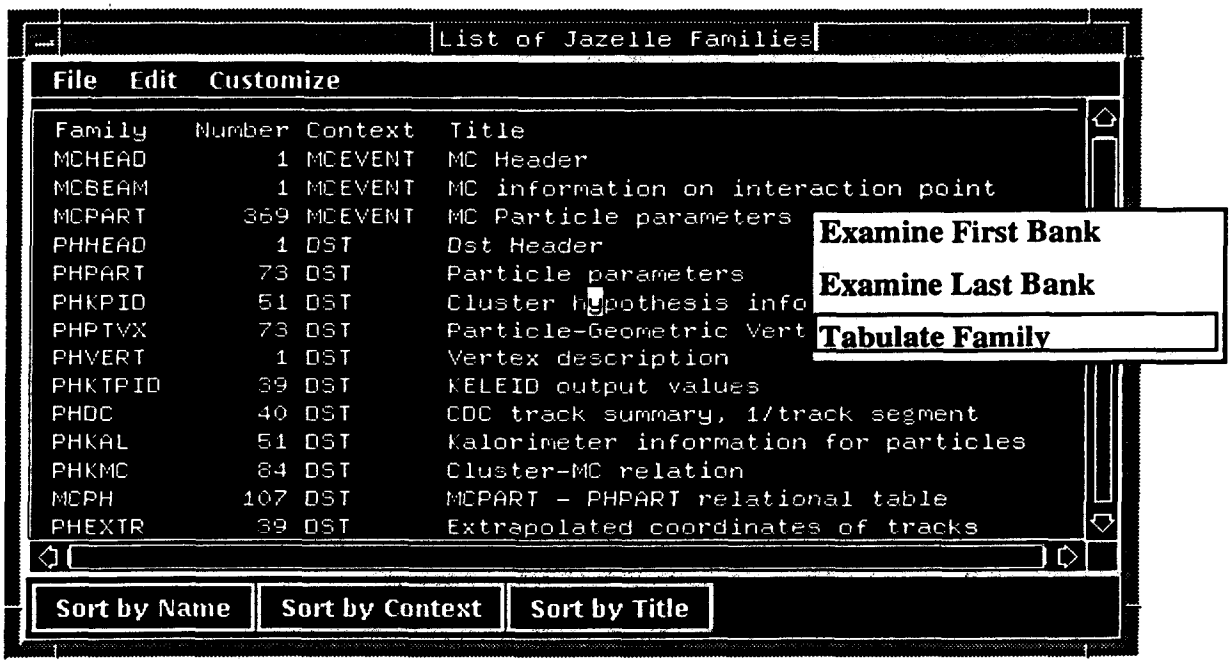

SLD IDA
Control Data Jazelle Idal Processors Customize
Histograms
Event Display
Go
Go 1
Go loop limits:
1000 events
300 seconds
++++++++++++++++++++++++++
** SLD IDA version 1.32 **
++++++++++++++++++++++++++
++++++++++++++++++++++++++
** JAZELLE version 3.21 **
++++++++++++++++++++++++++
HANDYPAK -- RELEASE 6.55 -- HCOM SIZE = 200000
%UPROC-I-INIT Processor support package (UPROC version 1.20) initialized
%JAZELLE-W-IOVCONV File DUCSSLD:SLDPDG is being converted from JAZELLE V 3.0
Executing IDA_PROFILE
1: rexx ida.list = 0
End of IDA_PROFILE
%JAZELLE-W-IOVCONV File IDA$DEFAULTS:DECW$IDA_CUSTOM_IDAL.DAT is being conve
IDA>
Ida>

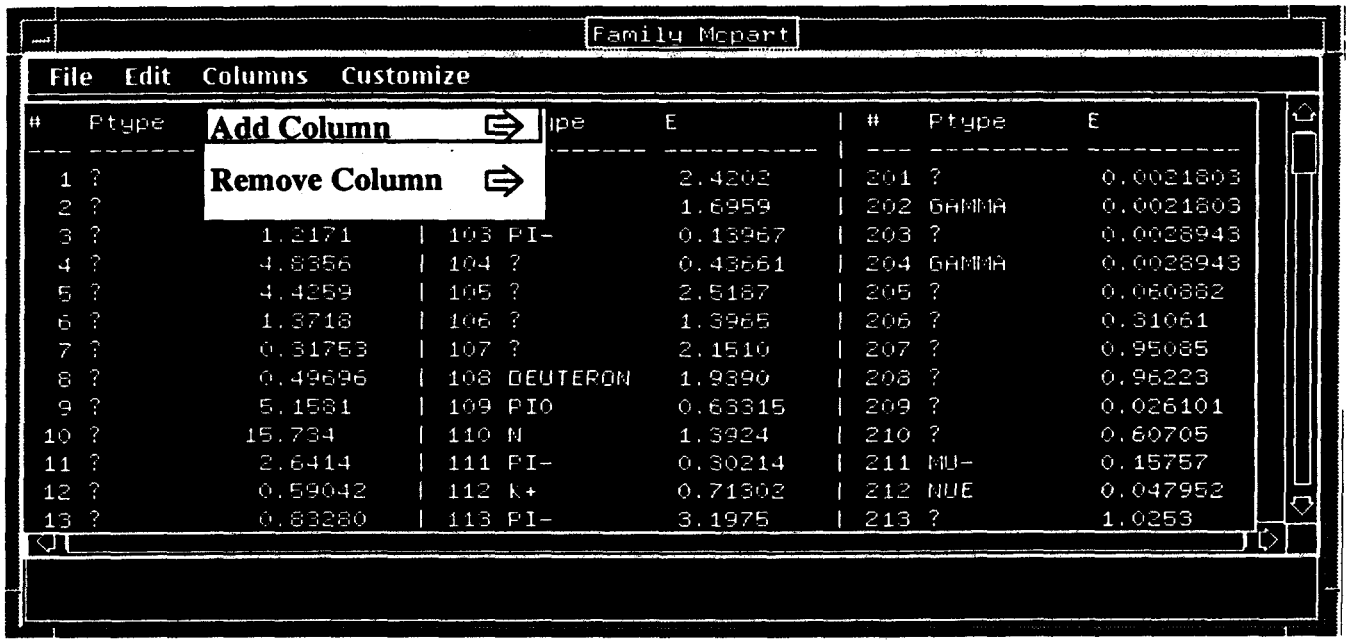

SLD IDA
Control
Data
Jazelle
Idal
Processors
Customize
Generate Using
Read Disk File...
Read Tape...
Read Online Data
Close Input File
Rewind Input File
Lund Version 6.3
SLD Lund
SUSY Generator
Diagnostic Generator
EP Generator
Bremmuus
Herwig 5.3
Histograms
Event Display
HANDYPAK -- RELEASE 6.55 -- HCOM SIZE = 200000
%UPROC-I-INIT Processor support package (UPROC version 1.20) initialized
%JAZELLE-W-IOVCONV File DUCSSLD:SLDPDG is being converted from JAZELLE V 3.0
Executing IDA_PROFILE
1: rexx ida.list = 0
End of IDA_PROFILE
%JAZELLE-W-IOVCONV File IDA$DEFAULTS:DECW$IDA_CUSTOM_IDAL.DAT is being conve
IDA>
Ida>

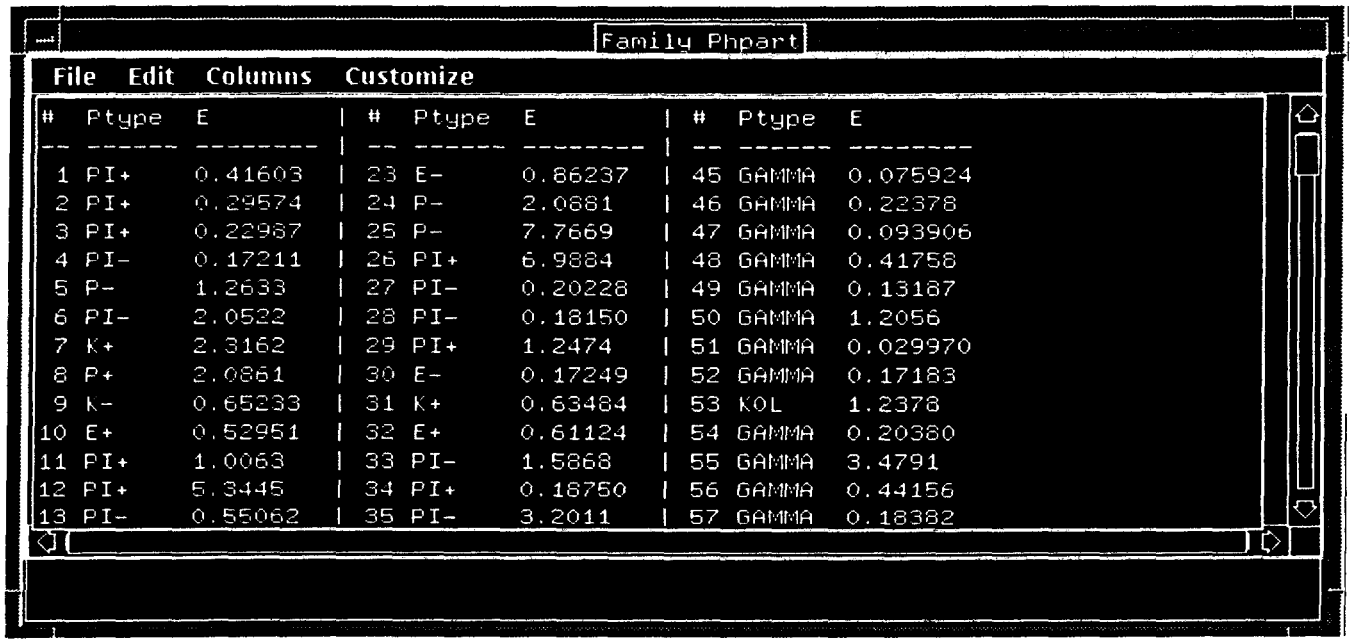

SLD IDA
Control Data Jazelle Idal Processors Customize
Histograms
Event Display
Lund Version 6.3
Go
Go loop limits:
View/Modify Parameters...
events
Set Processor Options
seconds
Remove
Move Up
Move Down
HANDYPAK -- P = 200000
%UPROC-I-INIT Processor support package (UPROC version 1.20) initialized
%JAZELLE-W-IOVCONV File DUCSSLD:SLDPDG is being converted from JAZELLE V 3.0
Executing IDA_PROFILE
1: rexx ida.list = 0
End of IDA_PROFILE
%JAZELLE-W-IOVCONV File IDA$DEFAULTS:DECW$IDA_CUSTOM_IDAL.DAT is being conve
IDA>
Dummy input file
%UPROC-I-PRCALL Processor MCL63 Version 1.00 called with option INIT
%UPROC-I-PRCALL Processor MCSLC Version 1.01 called with option INIT
Ida>

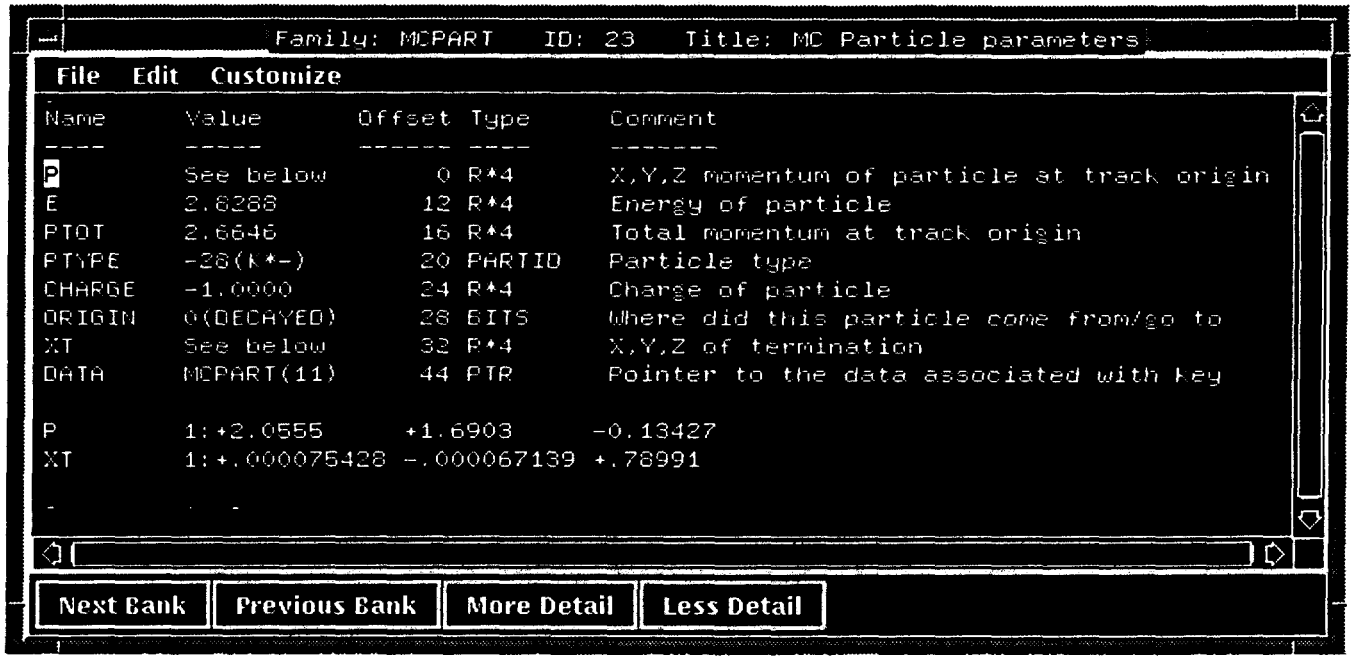

SLD IDA
Control Data Jazelle Idal Processors Customize
Histograms
Event Display
Lund Version 6.3
Go
Go 1
Go loop limits:
1000 events
300 seconds
** SLD IDA version 1.32 **
** JAZELLE version 3.21 **
HANDYPAK -- RELEASE 6.55 -- HCOM SIZE = 200000
%UPROC-I-INIT Processor support package (UPROC version 1.20) initialized
Executing IDA_
1: rexx ida.l
End of IDA_PRO
IDA>
Dummy input f
%UPROC-I-PRCAL
%UPROC-I-PRCAL
Ida>
Lund Version 6.3 - Control Parameters
None
Initial State Radiation
5
Number of Quark Flavors Generated
All
Quark Type(s) Generated
On
Higgs Production
.30000
Lambda for 2nd Order QCD alpha-s
.22000
Sin(ThetaW)**2 for Electroweak Mixing
93.105
Mass of the Z0
2.8670
Width of the Z0
100.00
Higgs Mass
85.000
Top Quark Mass
OK
Apply
Reset
Cancel

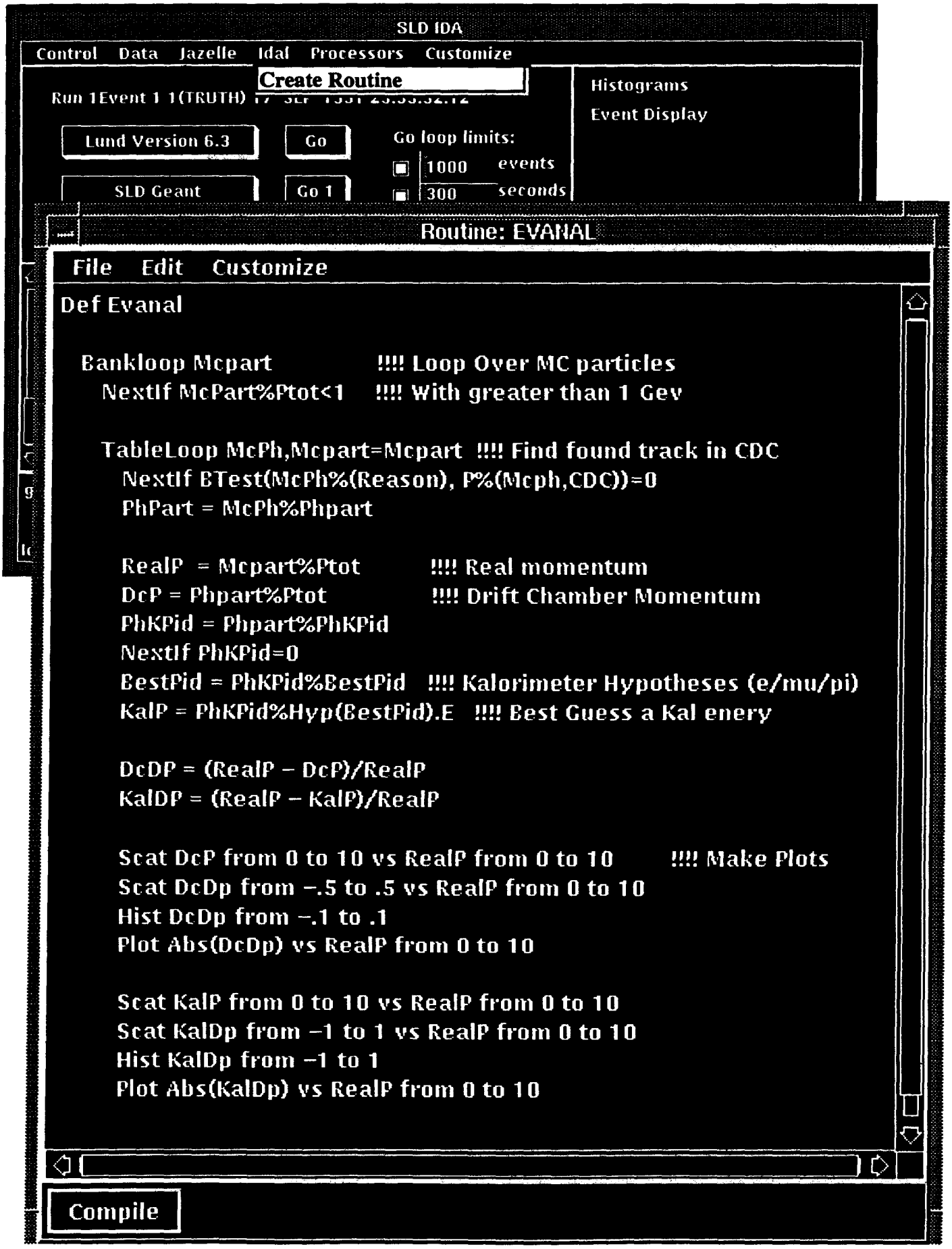

SLD IDA
Control Data Jazelle Idal Processors Customize
Show Memory Usage
List Banks
Read Consants....
Run 1Event 1 1(
Lund Versic
SLD Gea
limits:
events
seconds
Histograms
Event Display
Reconstruct SLD
%IDA-I-GOSTART Start GO for 1 records
THE LUND MONTE CARLO - JETSET VERSION 6.3
LAST DATE OF CHANGE: 10 FEBRUARY 1987
%IDA-I-GOEND End GO after .20000 seconds and 1 records (.20000 seconds/recor
Summary of GO
Type Count Procedure
INPUT 1 IJREAD
EVENT 1 EVGEN
go 1
Ida>

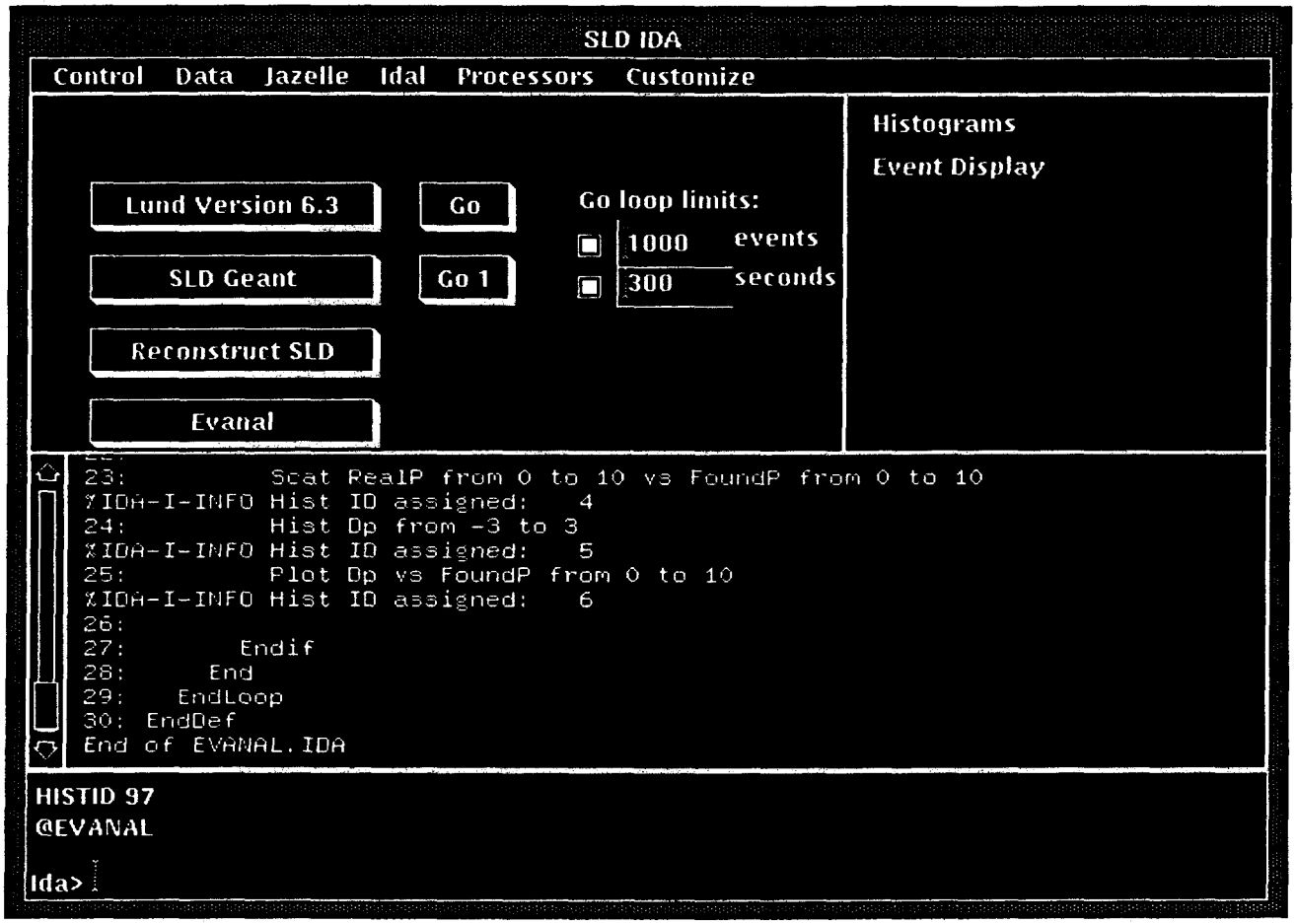

List of Jazelle Families
File Edit Customize
Family Number Context Title
MCHEAD 1 MCEVENT MC Header
MCBEAM 1 MCEVENT MC information on interaction point
MCPART 369 MCEVENT MC Particle parameters
PHHEAD 1 DST Dst Header
PHPART 73 DST Particle parameters
PHKPID 51 DST Cluster hypothesis info
PHPTVX 73 DST Particle-Geometric Vert
PHVERT 1 DST Vertex description
PHKTPID 39 DST KELEID output values
PHDC 40 DST CDC track summary, 1/track segment
PHKAL 51 DST Kalorimeter information for particles
PHKMC 84 DST Cluster-MC relation
MCPH 107 DST MCPART - PHPART relational table
PHEXTR 39 DST Extrapolated coordinates of tracks
Examine First Bank
Examine Last Bank
Tabulate Family
Sort by Name
Sort by Context
Sort by Title

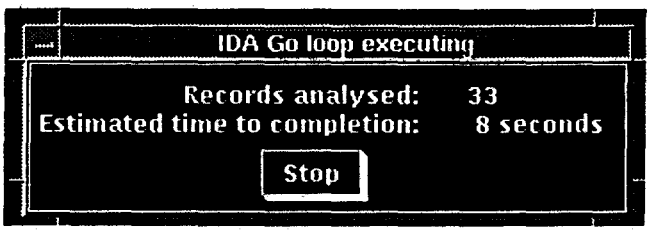

Family Mcpart
File Edit Columns Customize
Add Column
Remove Column
Ptype E # Ptype E
1 ? 2.4202 201 ? 0.0021803
2 ? 1.6959 202 GAMMA 0.0021803
3 ? 1.2171 103 PI- 0.13967 203 ? 0.0028943
4 ? 4.8356 104 ? 0.43661 204 GAMMA 0.0028943
5 ? 4.4259 105 ? 2.5167 205 ? 0.060882
6 ? 1.3718 106 ? 1.3965 206 ? 0.31061
7 ? 0.31753 107 ? 2.1510 207 ? 0.95085
8 ? 0.49696 108 DEUTERON 1.9390 208 ? 0.96223
9 ? 5.1581 109 PIO 0.63315 209 ? 0.026101
10 ? 15.734 110 N 1.3924 210 ? 0.60705
11 ? 2.6414 111 PI- 0.30214 211 MU- 0.15757
12 ? 0.59042 112 K+ 0.71302 212 NUE 0.047952
13 ? 0.83280 113 PI- 3.1975 213 ? 1.0253

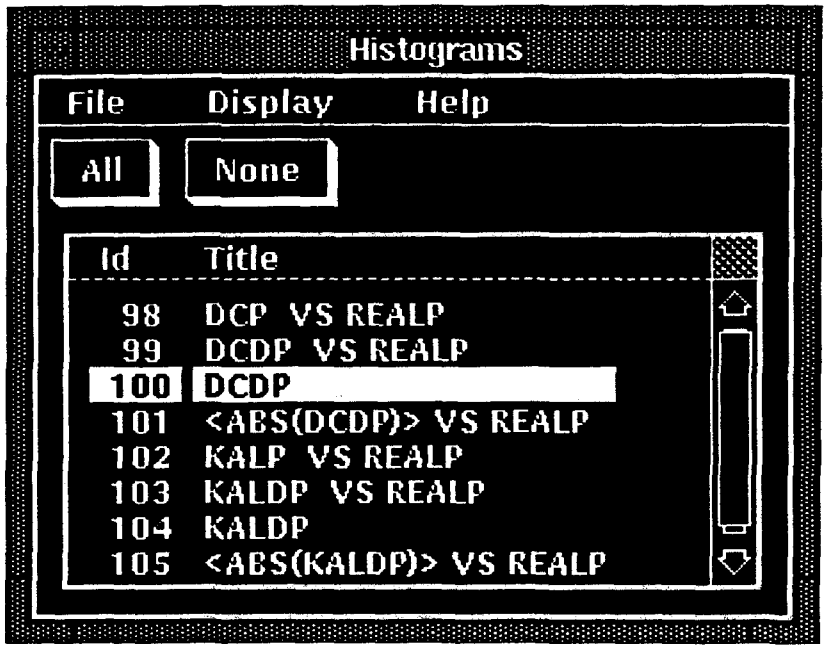

Family Phpart
File Edit Columns Customize
Ptype E # Ptype E # Ptype E
1 PI+ 0.41603 23 E- 0.86237 45 GAMMA 0.075924
2 PI+ 0.29574 24 P- 2.0681 46 GAMMA 0.22378
3 PI+ 0.22987 25 P- 7.7669 47 GAMMA 0.093906
4 PI- 0.17211 26 PI+ 6.9884 48 GAMMA 0.41758
5 P- 1.2633 27 PI- 0.20228 49 GAMMA 0.13187
6 PI- 2.0522 28 PI- 0.18150 50 GAMMA 1.2056
7 K+ 2.3162 29 PI+ 1.2474 51 GAMMA 0.029970
8 P+ 2.0861 30 E- 0.17249 52 GAMMA 0.17183
9 K- 0.65233 31 K+ 0.63484 53 KOL 1.2378
10 E+ 0.52951 32 E+ 0.61124 54 GAMMA 0.20380
11 PI+ 1.0063 33 PI- 1.5868 55 GAMMA 3.4791
12 PI+ 5.3445 34 PI+ 0.18750 56 GAMMA 0.44156
13 PI- 0.55062 35 PI- 3.2011 57 GAMMA 0.18382

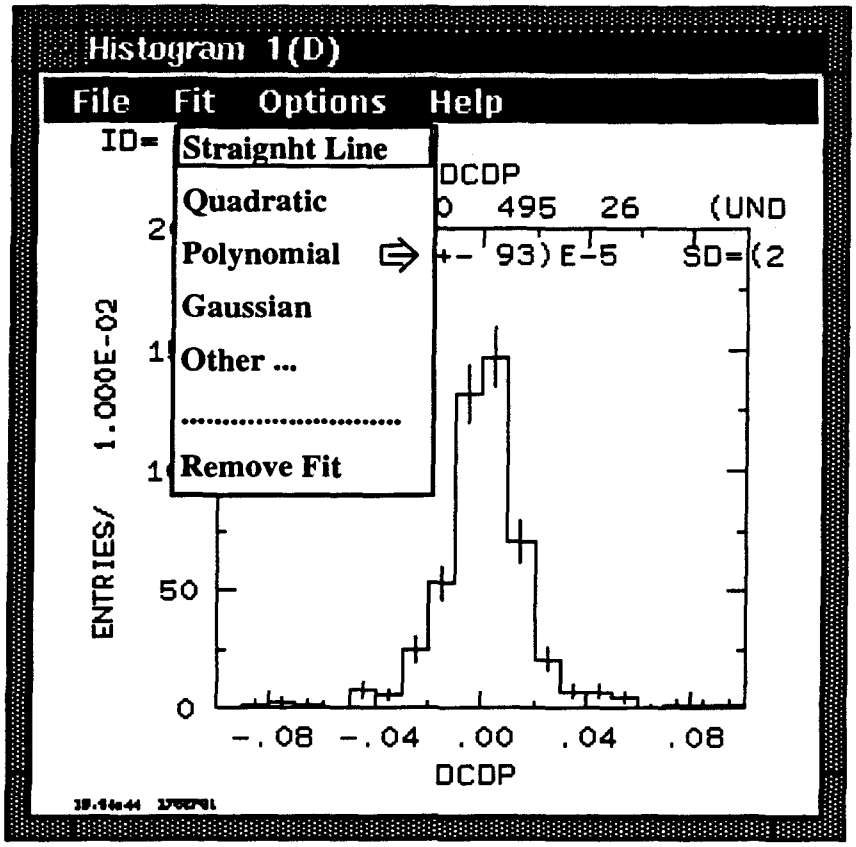

Family: MCPART ID: 23 Title: MC Particle parameters
File Edit Customize
Name Value Offset Type Comment
P See below 0 R*4 X,Y,Z momentum of particle at track origin
E 2.8288 12 R*4 Energy of particle
PTOT 2.6646 16 R*4 Total momentum at track origin
PTYPE -28(K*-) 20 PARTID Particle type
CHARGE -1.0000 24 R*4 Charge of particle
ORIGIN 0(DECAYED) 28 BITS Where did this particle come from/go to
XT See below 32 R*4 X,Y,Z of termination
DATA MCPART(11) 44 PTR Pointer to the data associated with key
P 1: +2.0555 +1.6903 -0.13427
XT 1: +.000075428 -.000067139 +.78991
Next Bank
Previous Bank
More Detail
Less Detail

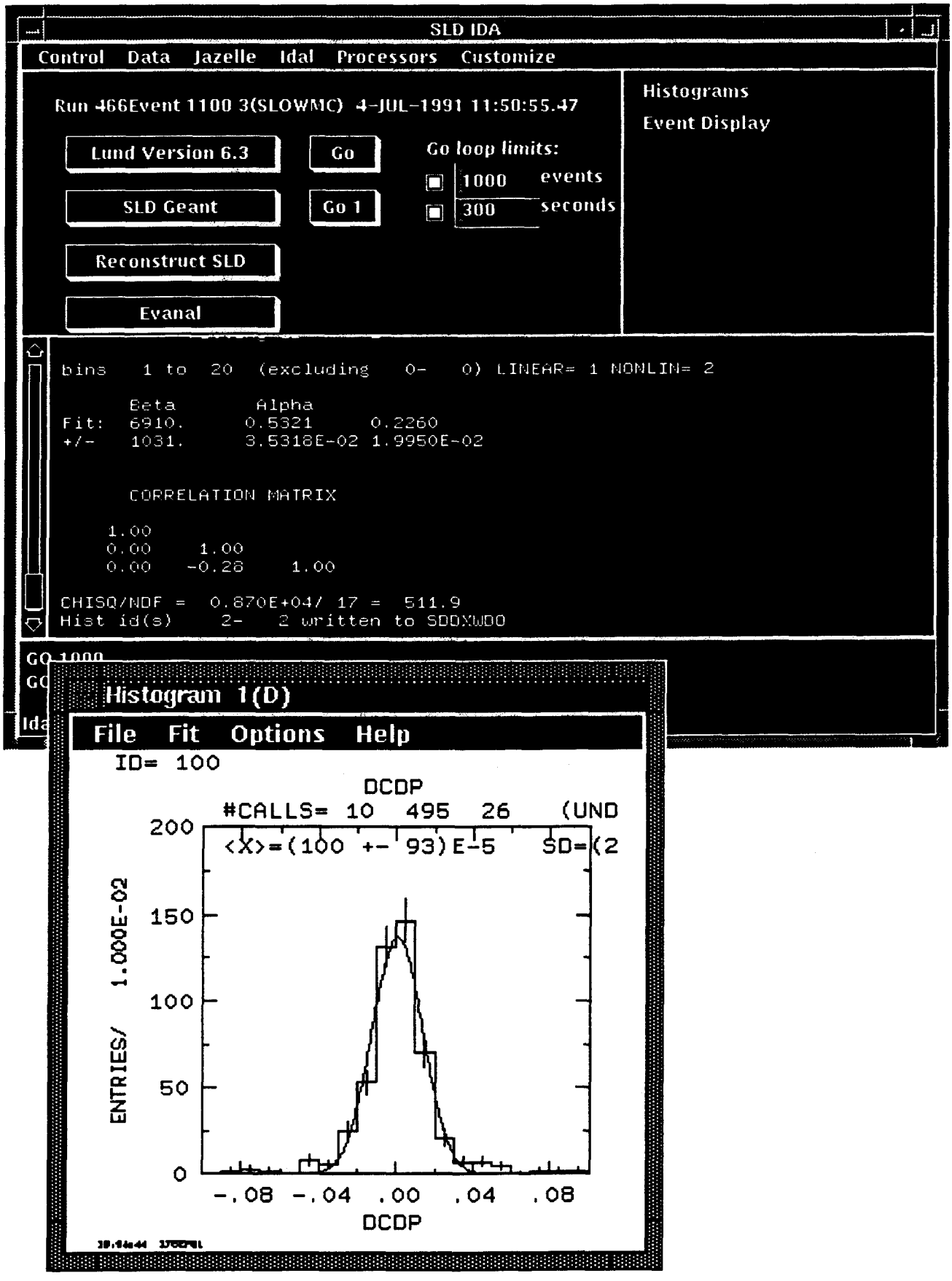

SLD IDA
Control Data Jazelle Idal Processors Customize
Create Routine
Run 1Event 1 1(TRUTH)
Histograms
Event Display
Lund Version 6.3
Go
Go loop limits:
1000 events
SLD Geant
Go 1
300 seconds
Routine: EVANAL
File Edit Customize
Def Evanal
Bankloop Mcpart !!!! Loop Over MC particles
NextIf McPart%Ptot<1 !!!! With greater than 1 Gev
TableLoop McPh,Mcpart=Mcpart !!!! Find found track in CDC
NextIf BTest(McPh%(Reason), P%(Mcph,CDC))=0
PhPart = McPh%Phpart
RealP = Mcpart%Ptot !!!! Real momentum
DcP = Phpart%Ptot !!!! Drift Chamber Momentum
PhKPid = Phpart%PhKPid
NextIf PhKPid=0
BestPid = PhKPid%BestPid !!!! Kalorimeter Hypotheses (e/mu/pi)
KalP = PhKPid%Hyp(BestPid).E !!!! Best Guess a Kal enery
DcDP = (RealP – DcP)/RealP
KalDP = (RealP – KalP)/RealP
Scat DcP from 0 to 10 vs RealP from 0 to 10 !!!! Make Plots
Scat DcDp from –.5 to .5 vs RealP from 0 to 10
Hist DcDp from –.1 to .1
Plot Abs(DcDp) vs RealP from 0 to 10
Scat KalP from 0 to 10 vs RealP from 0 to 10
Scat KalDp from –1 to 1 vs RealP from 0 to 10
Hist KalDp from –1 to 1
Plot Abs(KalDp) vs RealP from 0 to 10
Compile

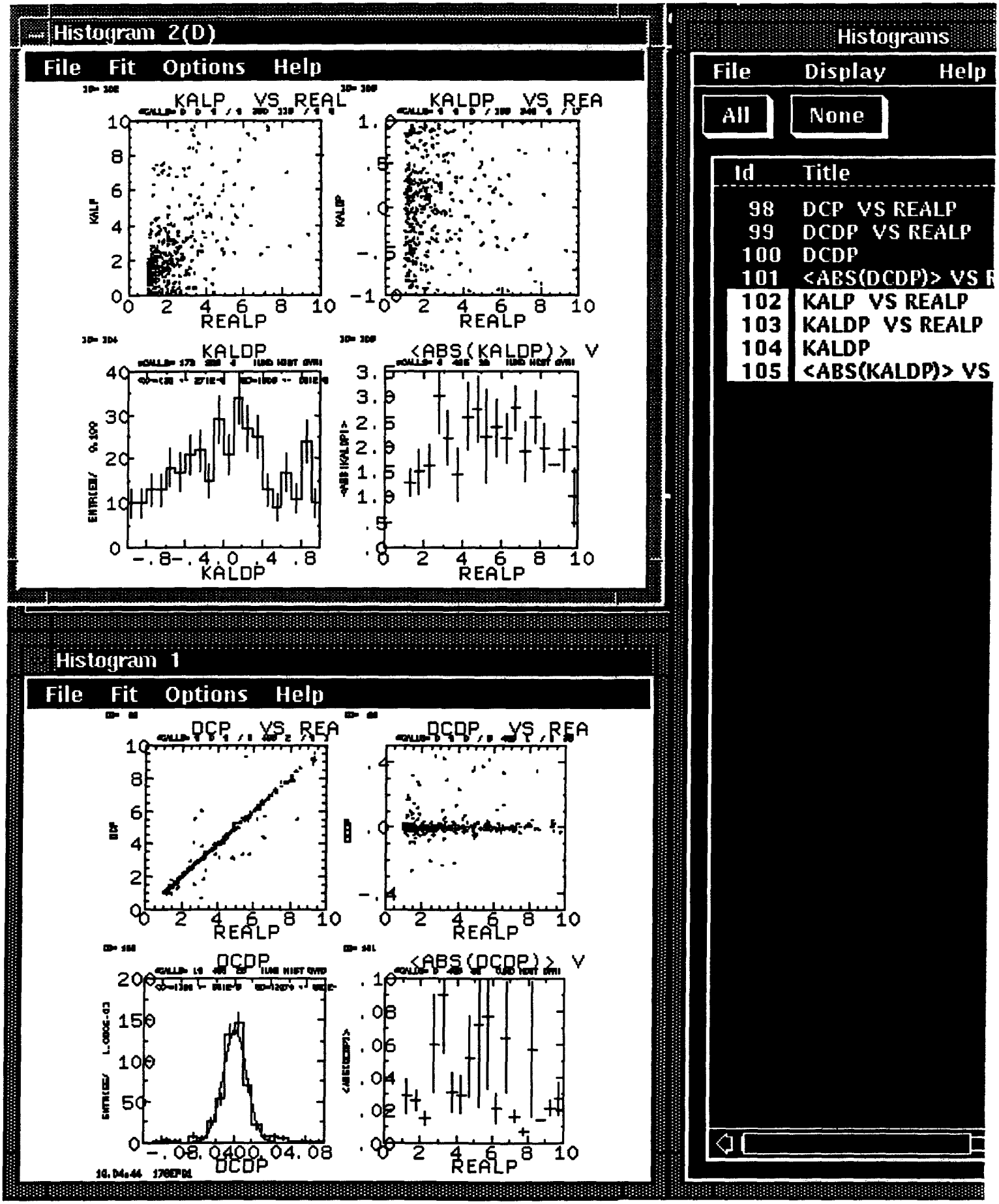

SLD IDA
Control Data Jazelle Idal Processors Customize
Histograms
Event Display
Lund Version 6.3
Go
Go loop limits:
1000 events
SLD Geant
Go 1
300 seconds
Reconstruct SLD
Evanal
23: Scat RealP from 0 to 10 vs FoundP from 0 to 10
%IDA-I-INFO Hist ID assigned: 4
24: Hist Dp from -3 to 3
%IDA-I-INFO Hist ID assigned: 5
25: Plot Dp vs FoundP from 0 to 10
%IDA-I-INFO Hist ID assigned: 6
26:
27: Endif
28: End
29: EndLoop
30: EndDef
End of EVANAL.IDA
HISTID 97
@EVANAL
Ida>

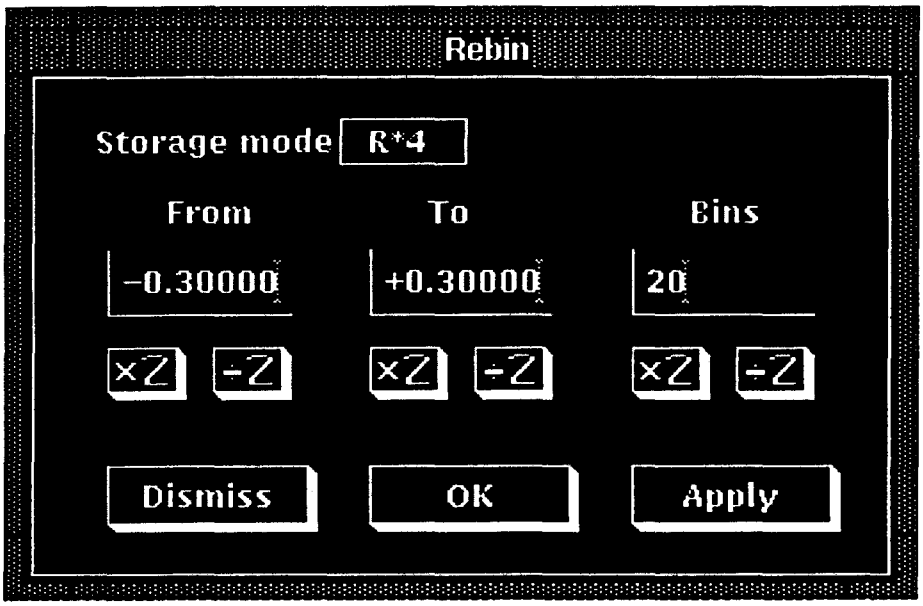

IDA Go loop executing
Records analysed: 33
Estimated time to completion: 8 seconds
Stop

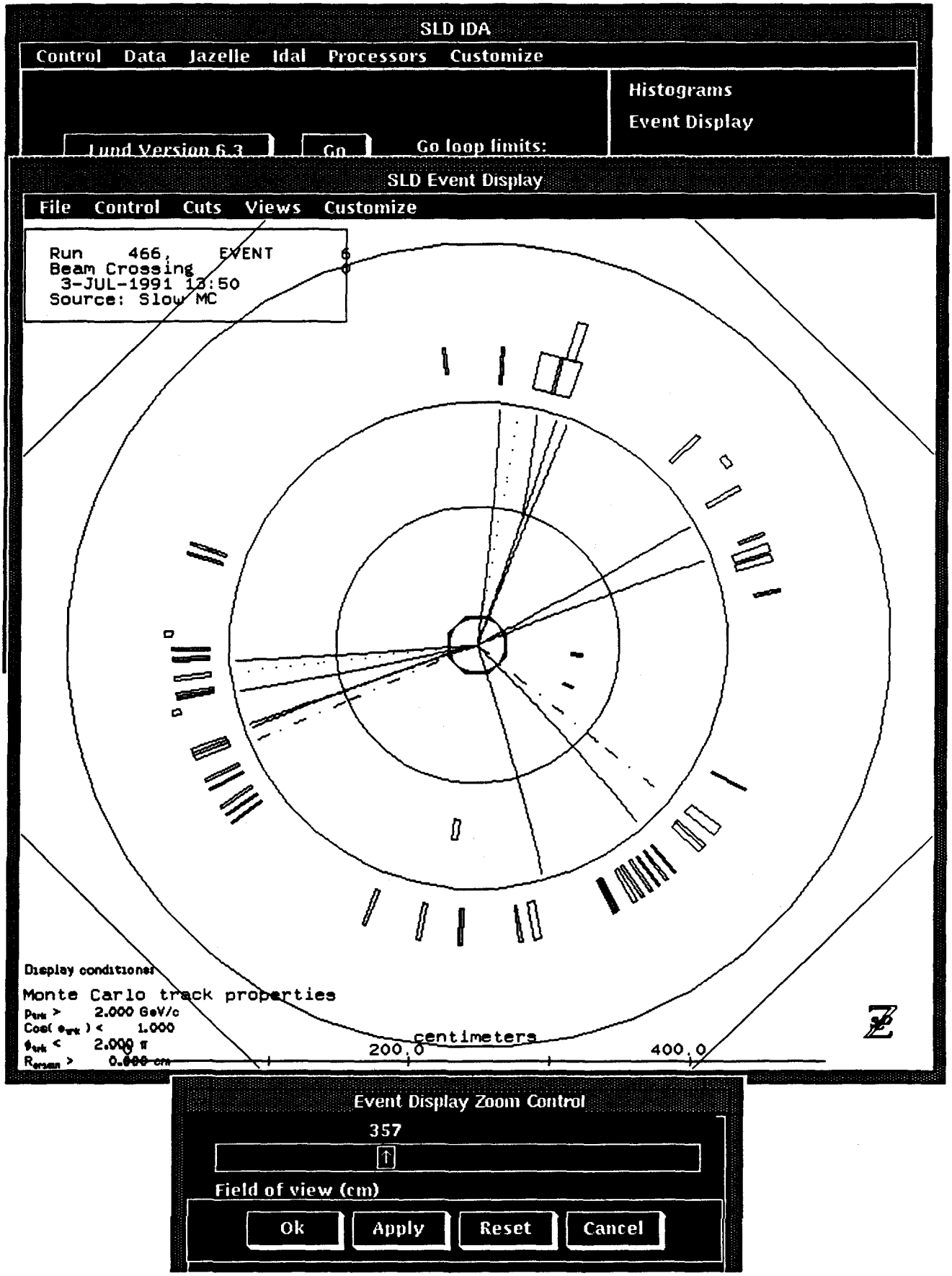

Histograms
File
Display
Help
All
None
Id
Title
98 DCP VS REALP
99 DCDP VS REALP
100 DCDP
101 <ABS(DCDP)> VS REALP
102 KALP VS REALP
103 KALDP VS REALP
104 KALDP
105 <ABS(KALDP)> VS REALP

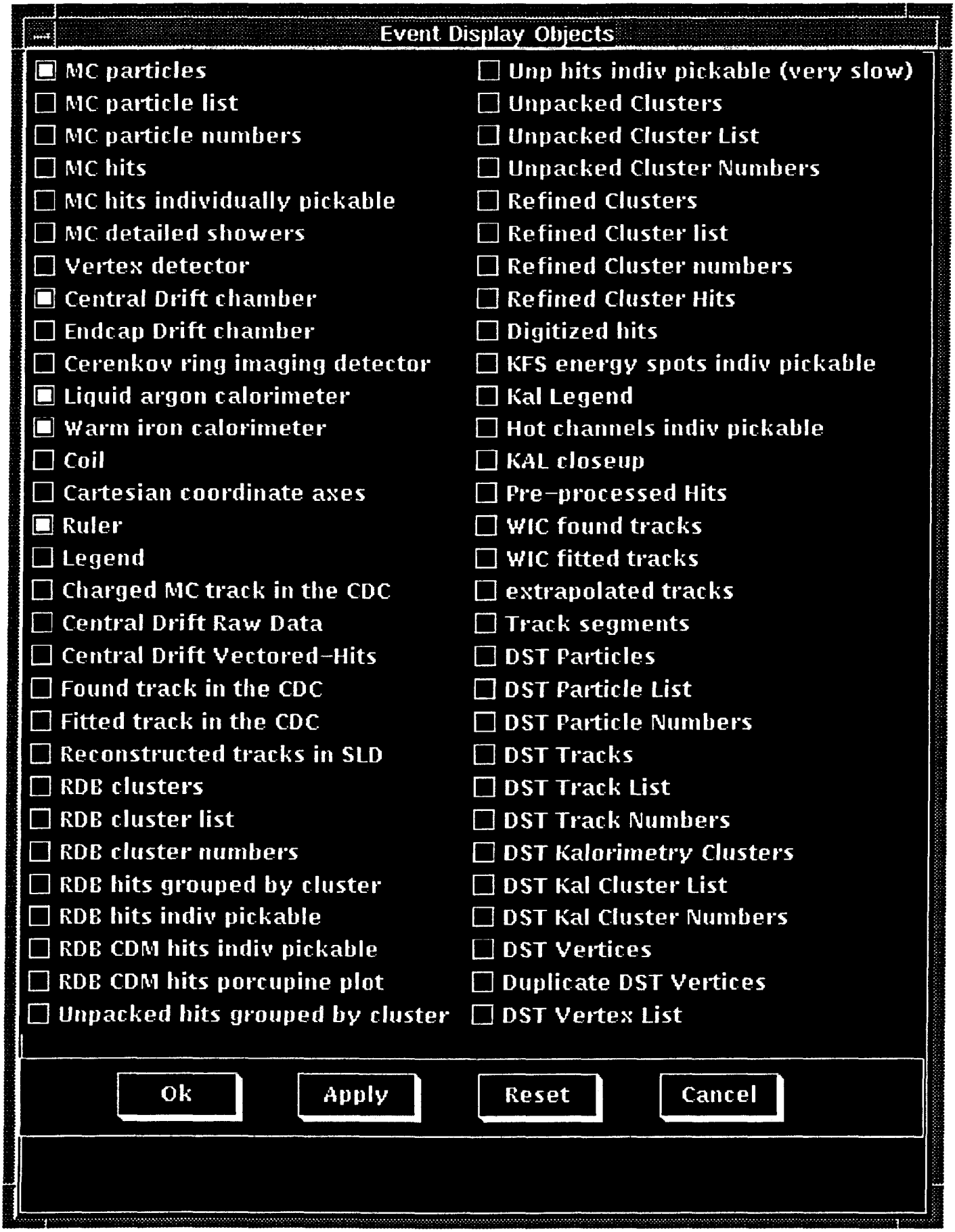

Histogram 1(D)
File
Fit
Options
Help
ID=
Straignht Line
Quadratic
Polynomial
Gaussian
Other ...
Remove Fit
DCDP
0 495 26 (UND
+- 93) E-5 SD= (2
ENTRIES/ 1.000E-02
2
1
1
50
0
-.08 -.04 .00 .04 .08
DCDP

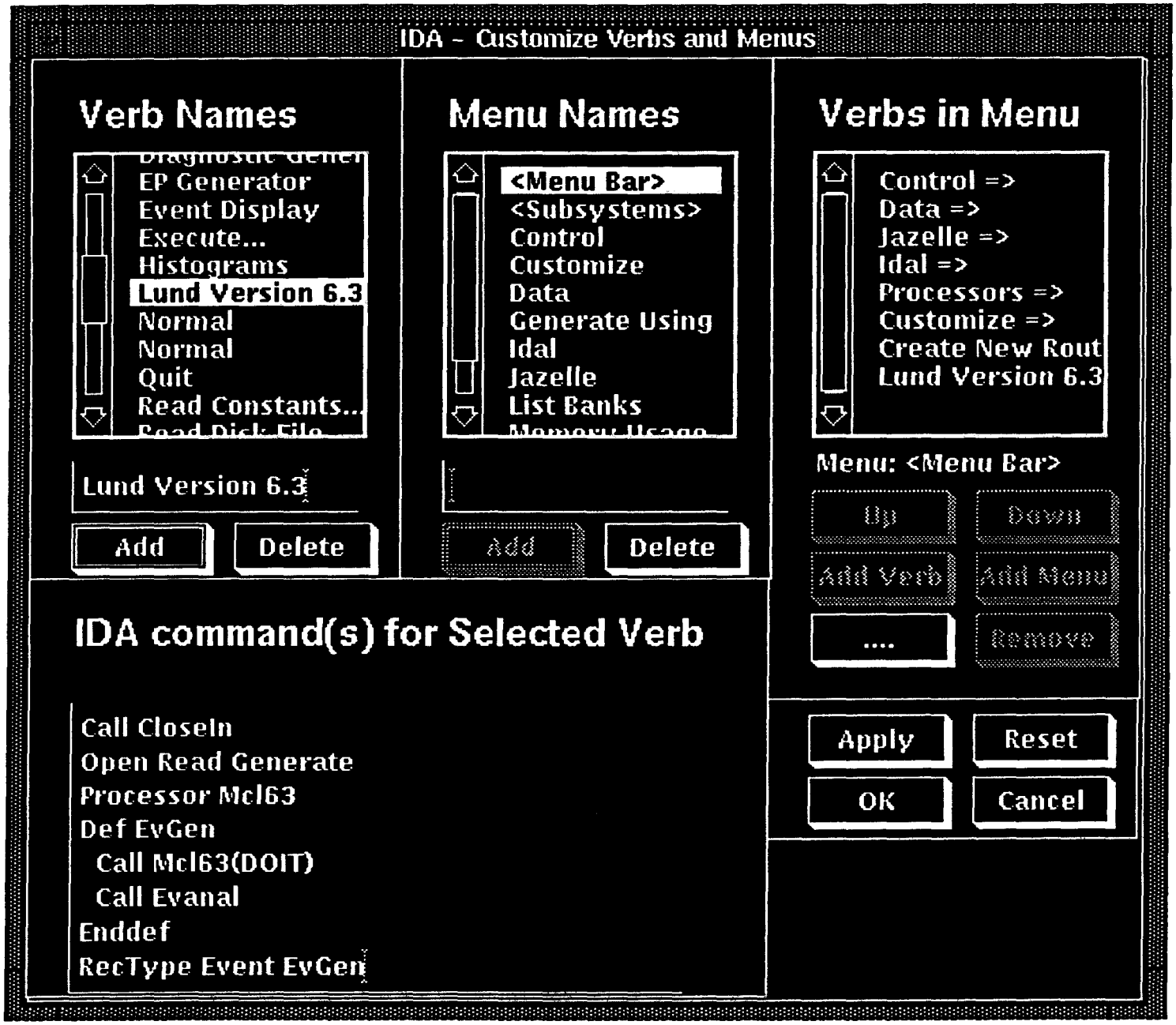

SLD IDA
Control Data Jazelle Idal Processors Customize
Run 466Event 1100 3(SLOWMC) 4-JUL-1991 11:50:55.47
Lund Version 6.3
Go
SLD Geant
Go 1
Reconstruct SLD
Evanal
Go loop limits:
1000 events
300 seconds
Histograms
Event Display
bins 1 to 20 (excluding 0- 0) LINEAR= 1 NONLIN= 2
Beta Alpha
Fit: 6910. 0.5321 0.2260
+/- 1031. 3.5318E-02 1.9950E-02
CORRELATION MATRIX
1.00
0.00 1.00
0.00 -0.28 1.00
CHISQ/NDF = 0.870E+04/ 17 = 511.9
Hist id(s) 2- 2 written to SDDXWDO
Histogram 1(D)
File Fit Options Help
ID= 100
DCDP
#CALLS= 10 495 26 (UND
<X>=(100 +- 93)E-5 SD=(2
ENTRIES/ 1.000E-02
200
150
100
50
0
-.08 -.04 .00 .04 .08
DCDP


Histogram 2(D)
File Fit Options Help
KALP VS REAL
KALDP VS REA
KALDP
<ABS(KALDP)> V
REALP
KALP
KALDP
Histogram 1
File Fit Options Help
DCP VS REA
DCDP VS REA
DCDP
<ABS(DCDP)> V
REALP
DCDP
Histograms
File Display Help
All
None
Id Title
98 DCP VS REALP
99 DCDP VS REALP
100 DCDP
101 <ABS(DCDP)> VS
102 KALP VS REALP
103 KALDP VS REALP
104 KALDP
105 <ABS(KALDP)> VS


Rebin
Storage mode
R*4
From
To
Bins
-0.30000
+0.30000
20
x2
÷2
x2
÷2
x2
÷2
Dismiss
OK
Apply


SLD IDA
Control Data Jazelle Idal Processors Customize
Histograms
Event Display
Lund Version 6.3
Go
Go loop limits:
SLD Event Display
File Control Cuts Views Customize
Run 466, EVENT 6
Beam Crossing
3-JUL-1991 13:50
Source: Slow MC
Display conditions
Monte Carlo track properties
200.0
400.0
centimeters
Event Display Zoom Control
357
Field of view (cm)
Ok
Apply
Reset
Cancel


Event Display Objects
MC particles
MC particle list
MC particle numbers
MC hits
MC hits individually pickable
MC detailed showers
Vertex detector
Central Drift chamber
Endcap Drift chamber
Cerenkov ring imaging detector
Liquid argon calorimeter
Warm iron calorimeter
Coil
Cartesian coordinate axes
Ruler
Legend
Charged MC track in the CDC
Central Drift Raw Data
Central Drift Vectored-Hits
Found track in the CDC
Fitted track in the CDC
Reconstructed tracks in SLD
RDB clusters
RDB cluster list
RDB cluster numbers
RDB hits grouped by cluster
RDB hits indiv pickable
RDB CDM hits indiv pickable
RDB CDM hits porcupine plot
Unpacked hits grouped by cluster
Unp hits indiv pickable (very slow)
Unpacked Clusters
Unpacked Cluster List
Unpacked Cluster Numbers
Refined Clusters
Refined Cluster list
Refined Cluster numbers
Refined Cluster Hits
Digitized hits
KFS energy spots indiv pickable
Kal Legend
Hot channels indiv pickable
KAL closeup
Pre-processed Hits
WIC found tracks
WIC fitted tracks
extrapolated tracks
Track segments
DST Particles
DST Particle List
DST Particle Numbers
DST Tracks
DST Track List
DST Track Numbers
DST Kalorimetry Clusters
DST Kal Cluster List
DST Kal Cluster Numbers
DST Vertices
Duplicate DST Vertices
DST Vertex List
Ok
Apply
Reset
Cancel


IDA - Customize Verbs and Menus
Verb Names
EP Generator
Event Display
Execute...
Histograms
Lund Version 6.3
Normal
Normal
Quit
Read Constants...
Lund Version 6.3
Add
Delete
Menu Names
<Menu Bar>
<Subsystems>
Control
Customize
Data
Generate Using
Idal
Jazelle
List Banks
Add
Delete
Verbs in Menu
Control =>
Data =>
Jazelle =>
Idal =>
Processors =>
Customize =>
Create New Rout
Lund Version 6.3
Menu: <Menu Bar>
Up
Down
Add Verb
Add Menu
....
Remove
IDA command(s) for Selected Verb
Call CloseIn
Open Read Generate
Processor Mcl63
Def EvGen
Call Mcl63(DOIT)
Call Evanal
Enddef
RecType Event EvGen
Apply
Reset
OK
Cancel

SLD's Goals for IDA

- Wanted to place an emphasis on interactive tools for analysis and for developing and testing MC-simulation-reconstruction.
- Wanted a uniform approach so that same tools would be available throughout the off-line environment.
- Immediate access to all data stored in data manager (no intermediate (N-tuple) files).
- Modular code design..easily extensible.
- Include fully interactive event display.
- Analysis environment should be fully/easily customizable by user.
- "Interactive" facilities must be available in batch.

"Maximize physicist productivity by providing a powerful, user-friendly analysis environment"

What exactly is MIDAS?

A tool to be used in conjunction with a "interface builder" to aid in the rapid development of Motif based tools.

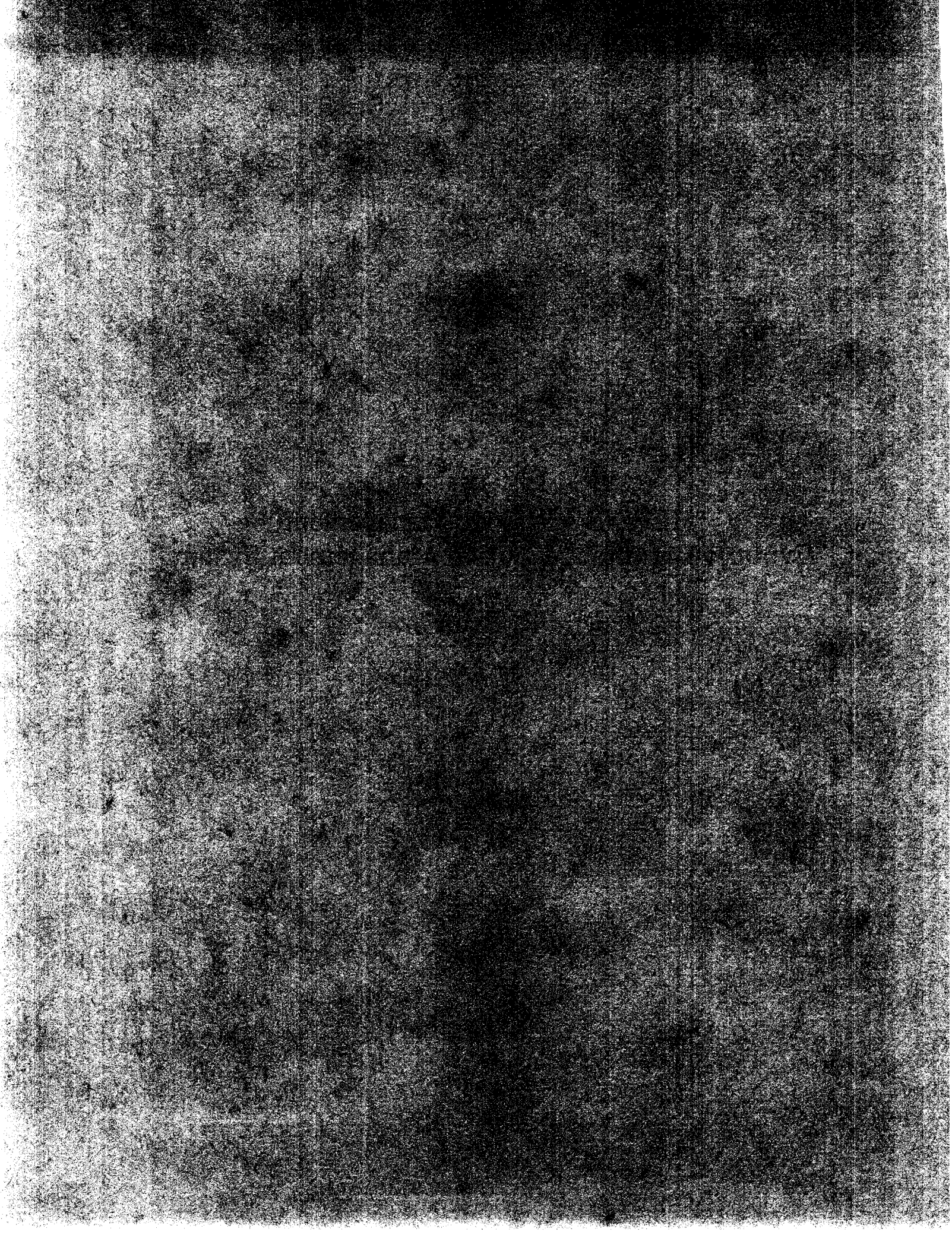
- Communicates with underlying application via "commands":
  - -- Provides automatic "accelerators".
  - -- Provides mechanism for running application from batch/terminals.
  - -- Provides "macro" programming capability.
  - -- Modular/Easily extensible.
- Is not limited by underlying "command" mechanism.
- Provides language for specifying functionality of interface.
- Isolates users wishing to extend user interface from Motif/X chaos.

Conclusions:

- IDA/MIDAS is so much fun to use people prefer playing with it rather than the "*flight simulator*"
- A very large fraction of the code written by SLD to support Interactive Data Analysis is completely non-specific to SLD.

"HEPLIB and Code Management"
"Evolution of the CMZ Code Management System"

R. Brun

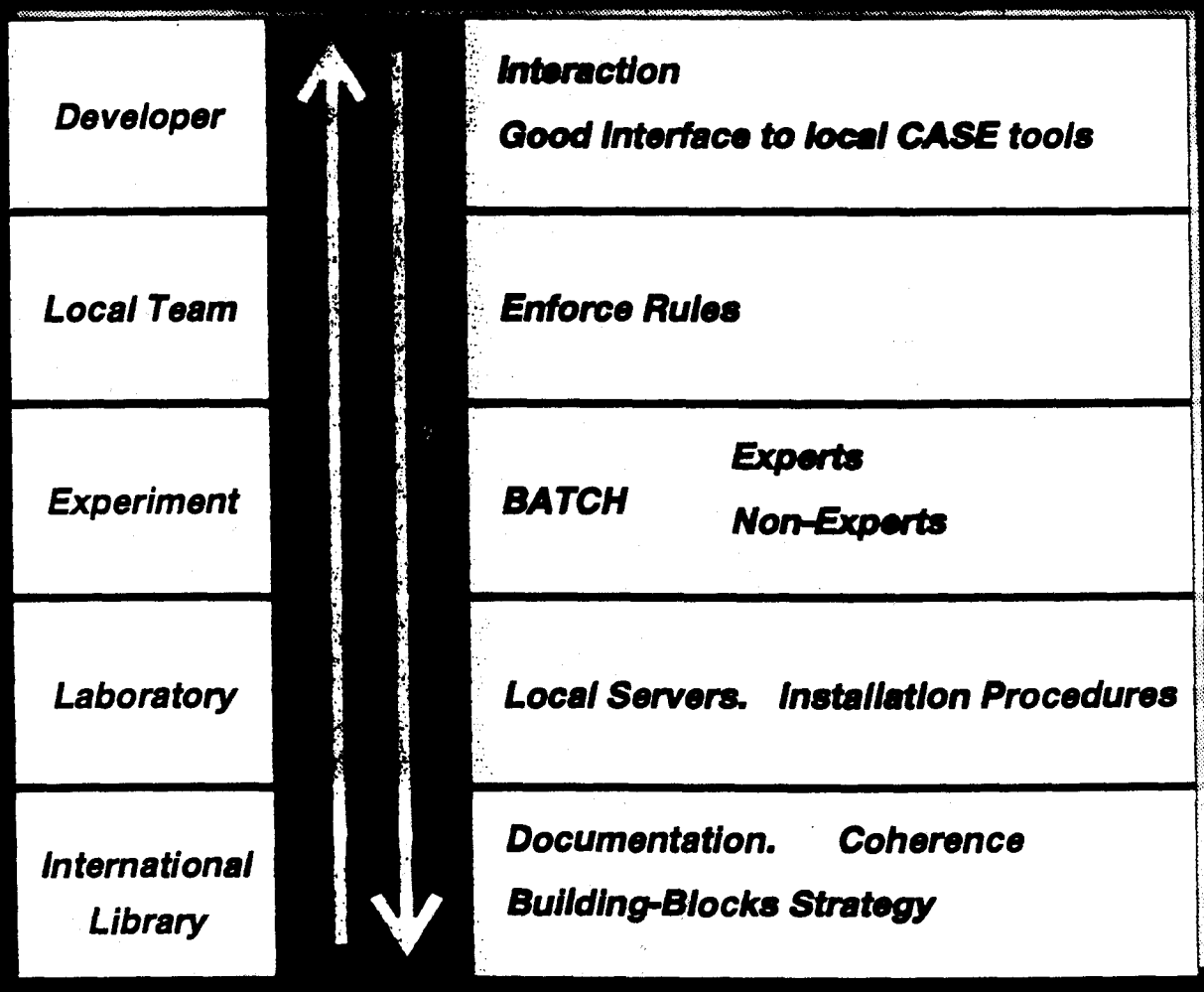


International Library : Requirements

- A collection of programs and tools developed at many center
- An international organization to decide on the contents,
  Give priorities to developments, avoid duplication
  and optimize information and exchanges.
- A manpower infrastructure to create the local libraries
  or executables and provide adequate User Support.
- A technical infrastructure to facilitate the distribution,
  the installation, the documentation and testing procedures.
  Here Code Management tools are essential.

The Code Management System

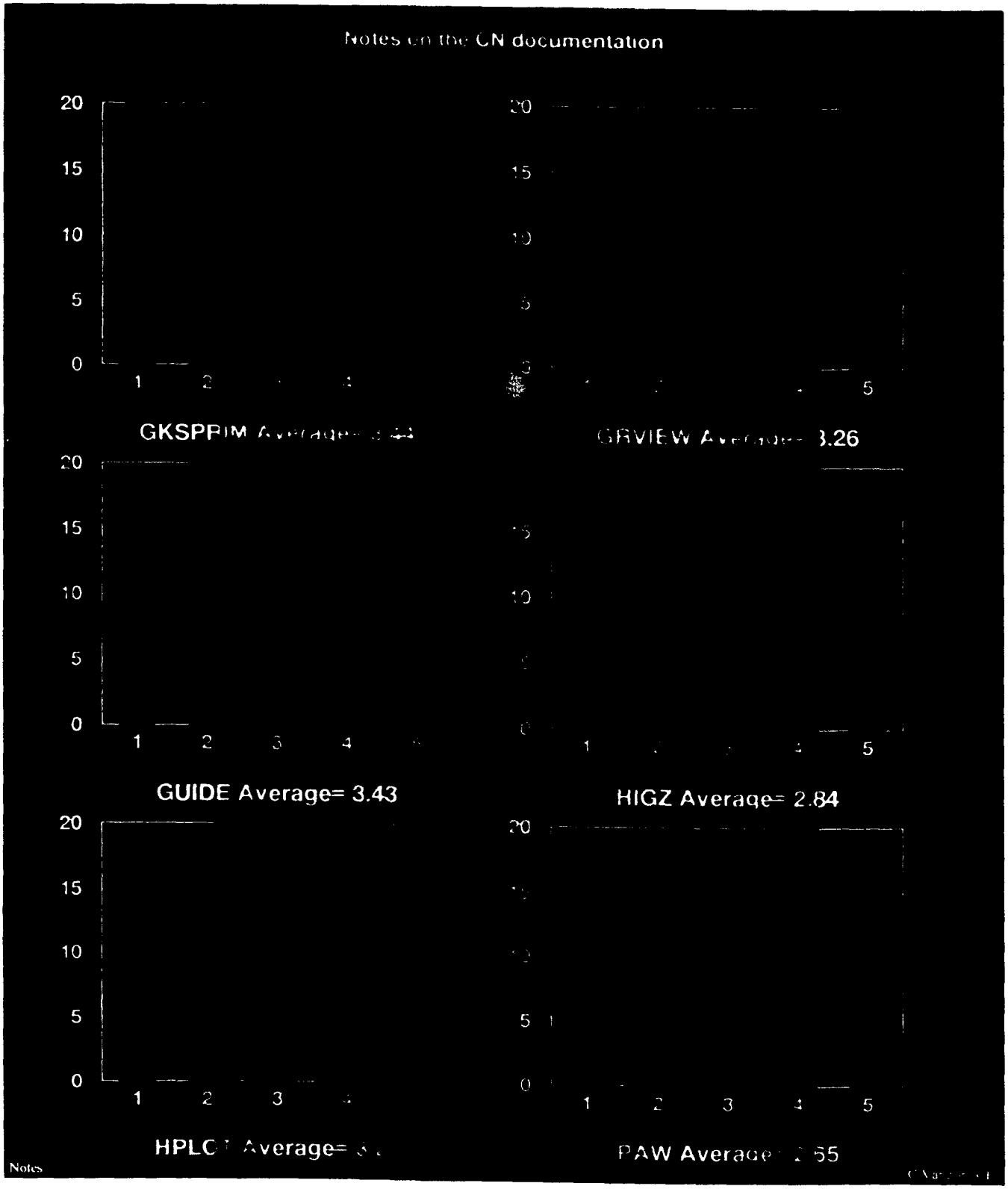
- Used at many levels
- Development, Maintenance and Management
- Many conflicting requirements



Plans for CERNLIB

- Upgrade existing programs to the file structure below
- Accept only programs following this structure
- Automatisation Installation Testing procedures
- Each file XXXX.CMZ self-describing
- The MakeF[illegible] files to build a library
  not the file [illegible]
- All EXECs [illegible] CMZ

File XXXX.CMZ	
Source Code	+PATCH, +DECK, etc...
Installation EXECs and HELP files	$KUMACS
Documentation / tools LateX, XDVI, GS, etc.	$DOC
DDL DZDOC	Data Dictionary Global data (static and dynamic)
Examples Test Procedures	Test Program with validation suite used by the Install-Server


Notes on the CN documentation
20
15
10
5
0
1
2
3
4
5
GKSPRIM Average= 3.43
GRVIEW Average= 3.26
GUIDE Average= 3.43
HIGZ Average= 2.84
HPLOT Average=
PAW Average= 2.55
Notes

```
*B..GENE      Generator Bank
*B.AU         S.L.Lloyd
*B.ST         IXSTOR
*B.DV         IXDIV
*B.NL         3
*B.NS         3
*B.ND         8
*B.UP         MC -1
*B.OR         MC -1
*B.LINK
*B.1 TREE     Tree of generated particles
*B.2 JETS     Generated partons
*B.3 RAWG     Raw generator bank e.g. LUND
*B/LINK
*B.DATA
*B.1 NGEN     Number of generated particles
*B.2 NPRIM    Number of primary particles
*B.3 NJET     Number of primary partons
*B.4 NSJET    Number of secondary partons
*B.5 LENT     Length of a TREE entry (=22)
*B.6 LENJ     Length of a JETS entry (=8)
*B.7 IOTREE   IO pointer for TREE bank
*B.8 IOJETS   IO pointer for JETS bank
*B/DATA
*B/
```

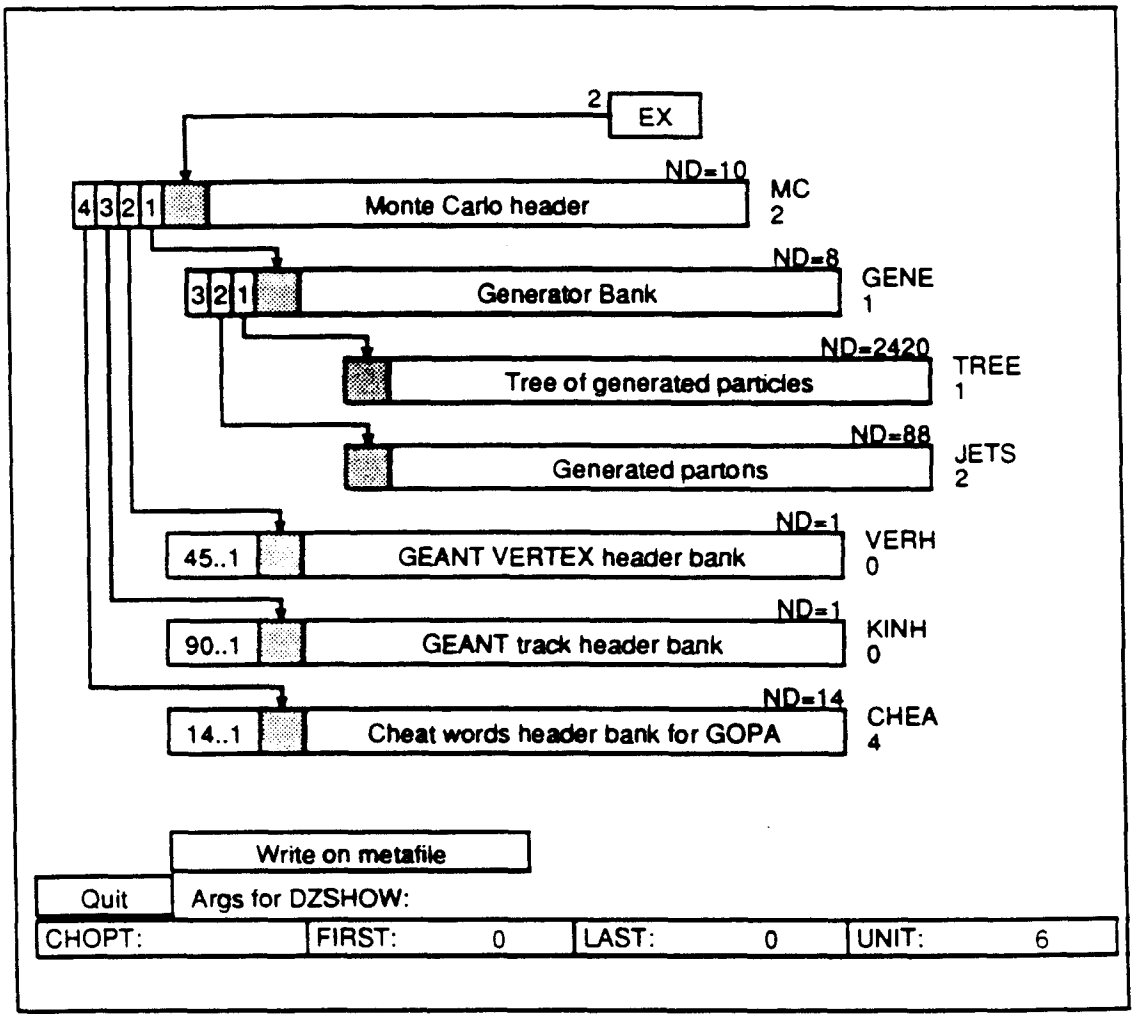
Example of Bank descriptor cards

1	NGEN	110	Number of generated particles
2	NPRIM	16	Number of primary particles
3	NJET	11	Number of primary partons
4	NSJET	0	Number of secondary partons
5	LENT	22	Length of a TREE entry (=22)
6	LENJ	8	Length of a JETS entry (=8)
7	IOTREE	2293890	IO pointer for TREE bank
8	IOJETS	2424962	IO pointer for JETS bank

Documented contents of data words

First word (IOcb/NOFF)HEX	2000F
Bank status word (HEX)	0
Hollerith ID	GENE
Numerical ID	1
Total number of links	3
Number of structural links	3
Number of data words	8
Next link	0
Up - link	192928 MC
Origin link	192927 JBIAS = 1
1. down link	190314 TREE
2. down link	190215 JETS
3. down link	0

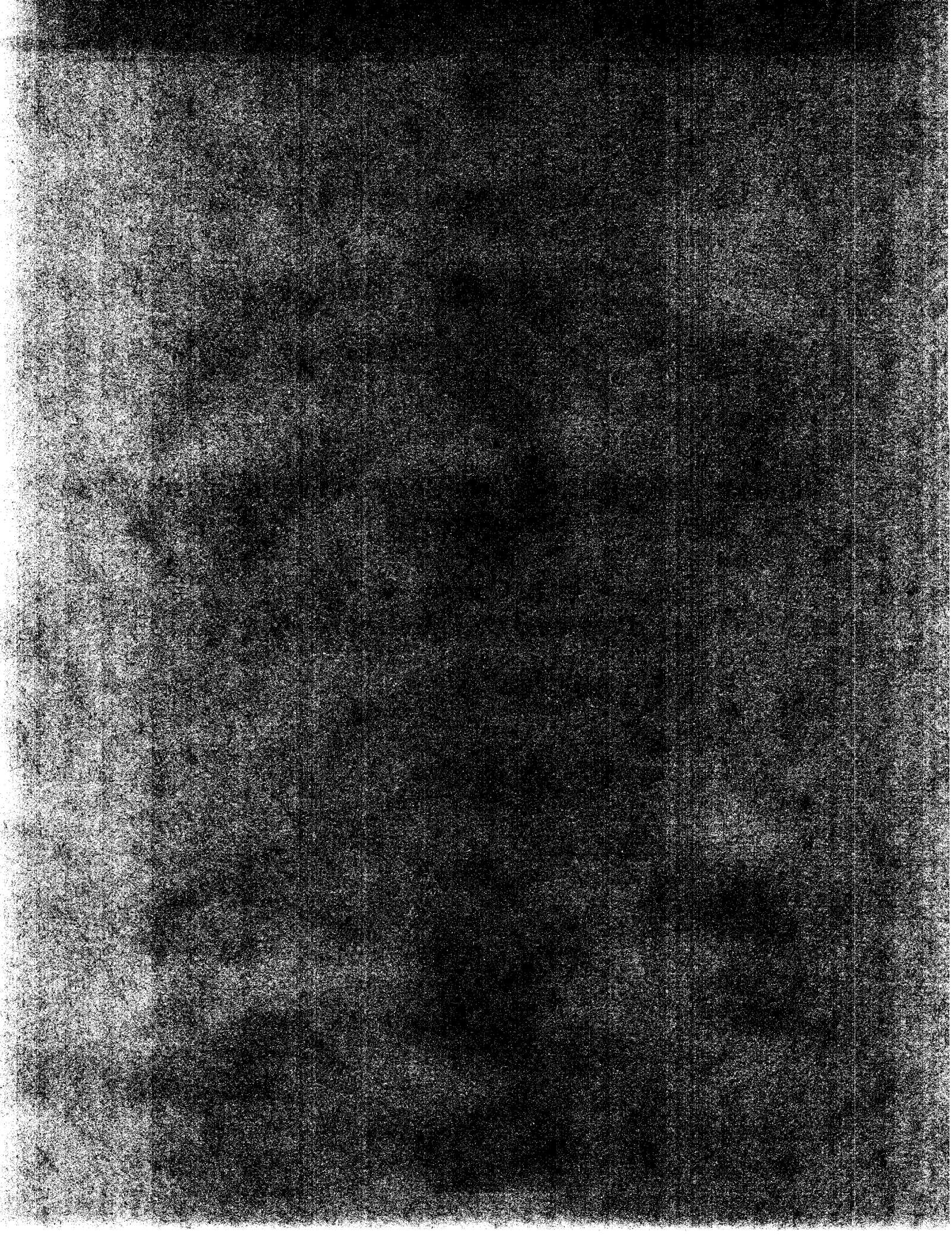
Contents of system words and links



Display of bank tree generated by DZDISP

"Standard HEP Particle Data Book on the Physicist's Workstation"

L. Garren
Presented by P. LeBrun



as advertised:

Standard HEP Particle Data Book on the physicist's workstation

what I really will talk about:

Beyond HEPEVT

a standard Monte Carlo generator interface

Lynn Garren, F.N.A.L.

- multiple generators
  - ISAJET
  - HERWIG
  - JETSET
  - others
- output to detector simulation
  - GEANT
  - "home-grown"
- decay and particle definitions
- **physicist ???????**

1987 proposal for standard HEP particle ID numbers

- T. G. Trippe and G. R. Lynch, "Proceedings of the Workshop on Detector Simulation for the SSC", (August 1987).

- Particle Data Group, J. J. Hernandez, et al., Phys. Lett. B239, III.67-III.68, (1990).

standard event format proposed in 1989 at CERN

- not well publicized

- B. Bambah et al., Z Physics at LEP 1, v3, (1989).

THIS IS A GOOD START

The standard output common block has the following form:

```
PARAMETER (NMXHEP=5000)
COMMON/HEPEVT/NEVHEP,NHEP,ISTHEP(NMXHEP),IDHEP(NMXHEP),
1  JMOHEP(2,NMXHEP),JDAHEP(2,NMXHEP),PHEP(5,NMXHEP),VHEP(4,NMXHEP)
```

NEVHEP	-	event number
NHEP	-	number of entries in this event record
ISTHEP(..)	-	status code
IDHEP(..)	-	particle ID, P.D.G. standard
JMOHEP(1,..)	-	position of mother particle in list
JMOHEP(2,..)	-	position of second mother particle in list
JDAHEP(1,..)	-	position of first daughter in list
JDAHEP(2,..)	-	position of last daughter in list
PHEP(1,..)	-	x momentum in GeV/c
PHEP(2,..)	-	y momentum in GeV/c
PHEP(3,..)	-	z momentum in GeV/c
PHEP(4,..)	-	energy in GeV
PHEP(5,..)	-	mass in GeV/c**2
VHEP(1,..)	-	x vertex position in mm
VHEP(2,..)	-	y vertex position in mm
VHEP(3,..)	-	z vertex position in mm
VHEP(4,..)	-	production time in mm/c

NEVHEP is -1 for initialization records and -2 for final records. Jetset does not use event numbers, so NEVHEP is initialized the first time you call the Jetset translation routine and incremented by one for each call thereafter. In order to keep Isajet information available, JMOHEP(2,..) contains the jet identification. Neither Herwig nor Isajet have any vertex information, so VHEP is zero.

Appendix A

Elementary Particle Identification Code Listing

Quarks and Leptons					
Particle	ISAJET 6.36	JETSET 7.3	HERWIG 5.3	1990 PDG	my solution
d	2	1	1	1	1
u	1	2	2	2	2
s	3	3	3	3	3
c	4	4	4	4	4
b	5	5	5	5	5
t	6	6	6	6	6
l	7 (x)	7	7 (bl)		7
h	8 (y)	8	8 (tl)		8
e^-	12	11	11	11	11
ν_e	11	12	12	12	12
μ^-	14	13	13	13	13
ν_μ	13	14	14	14	14
τ^-	16	15	15	15	15
ν_τ	15	16	16	16	16
χ^-		17			17
ν_χ		18			18

Gauge and Higgs Bosons					
Particle	ISAJET 6.36	JETSET 7.3	HERWIG 5.3	1990 PDG	my solution
g	9	21	21	21 (9)	21 (9)
γ	10	22	22	22	22
Z^0	90	23	23	23	23
W^+	80	24	24	24	24
H_1^0	81 (H^0)	25 (H^0)	25	25	25
LQ^0		39			28
R^0		40			29
Z_2^0		32 (Z'^0)			30
W_2^+		34 (W'^+)			31
H_2^0	82 (H_1^0)	35 (H'^0)		35 (H_2^0)	32
H_2^+	86 (H_1^+)	37 (H^+)		37 (H^+)	33
H_2^{++}	88 (H_1^{++})				34
Z_3^0		33 (Z''^0)			35
H_3^0	83 (H_2^0)	36 (A^0)		36 (H_3^0)	36
H_3^+	87 (H_2^+)				37
H_3^{++}	89 (H_2^{++})				38
H_4^0	84 (H_3^0)				39
H_5^0	85 (H_4^0)				40

1666

Charmed Mesons					
Meson	ISAJET 6.36	JETSET 7.3	HERWIG 5.3	1990 PDG	my solution
$\eta_c(1S)$	440	441	441	441	441
χ_{c0}		10441	10441 (χ)	10441	10441
$\eta_c(2S)$					20441
J/ψ	441	443	443	443	443
$h_c(1P)$		10443			10443
χ_{c1}		20443	10441 (χ)	10443	20443
$\psi(2S)$			20443 (ψ')	20443	30443
$\psi(3770)$			30443 (ψ'')	30443	40443
$\psi(4040)$				40443	50443
$\psi(4160)$				50443	60443
$\psi(4415)$				60443	70443
χ_{c2}		445	10441 (χ)	445	445

Charmed Baryons						
Baryon	quark content	ISAJET 6.36	JETSET 7.3	HERWIG 5.3	1990 PDG	my solution
Λ_c^+	duc	2140	4122	4122	4122	4122
Σ_c^{++}	uuc	1140	4222	4222	4222	4222
Σ_c^+	duc	1240	4212	4212	4212	4212
Σ_c^0	ddc	2240	4112	4112	4112	4112
Σ_c^{*++}	uuc	1141	4224	4224		4224
Σ_c^{*+}	duc	1241	4214	4214		4214
Σ_c^{*0}	ddc	2241	4114	4114		4114
Ξ_c^+	usc	1340	4232	4232	4322	4322
Ξ_c^0	dsc	2340	4132	4132	4312	4312
Ξ_c^{*+}	usc	1341	4324	4324 (SS*+)		4324
Ξ_c^{*0}	dsc	2341	4314	4314 (SS*0)		4314
$\Xi_c^{\prime+}$	usc	3140	4322	4322 (SS+)		4232
$\Xi_c^{\prime 0}$	dsc	3240	4312	4312 (SS0)		4132
Ω_c^0	ssc	3340	4332	4332	4332	4332
Ω_c^{*0}	ssc	3341	4334	4334 (T^{*0})		4334
Ξ_{cc}^+	dcc	2440	4412			4412
Ξ_{cc}^{++}	ucc	1440	4422			4422
Ξ_{cc}^{*+}	dcc	2441	4414			4414
Ξ_{cc}^{*++}	ucc	1441	4424			4424
Ω_{cc}^+	scc	3440	4432			4432
Ω_{cc}^{*+}	scc	3441	4434			4434
Ω_{ccc}^{*++}	ccc	4441	4444			4444

STDHEP

- today
  - provide translation routines to standard common block
  - make sure particle ID numbers are the same
- future
  - input particle definitions from standard decay table (Paul Avery, Univ. of Florida)
  - provide event and decay table I/O
- pie in the sky
  - standard generator interface

STDHEP Routines	
subroutine	action
HEPLST(MLST)	print an event
HEPNAM(ID,CHAU)	return a 20 character name, CHAU, for particle ID
HEPWRT(LOK)	write out an event
HEPRD(LOK)	read in an event
ISAHEP(MCONV)	translate to/from Isajet event format
ISAPRTLST(JTLIS)	print a list of particles defined by Isajet
LUNHEP(MCONV)	translate to/from Jetset event format
HWGHEP(MCONV)	translate to/from Herwig event format
function	action
HEPCMP(ID)	compress particle ID for use in array (1600)
HEPCHG(ID)	get charge for particle ID
ISTRAN(ID,MCONV)	translate to/from Isajet particle ID numbering
LUTRAN(ID,MCONV)	translate to/from Jetset particle ID numbering
HWTRAN(ID,MCONV)	translate to/from Herwig particle ID numbering

Appendix A

Elementary Particle Identification Code Listing

Quarks and Leptons					
Particle	ISAJET 6.36	JETSET 7.3	HERWIG 5.3	1990 PDG	my solution
d	2	1	1	1	1
u	1	2	2	2	2
s	3	3	3	3	3
c	4	4	4	4	4
b	5	5	5	5	5
t	6	6	6	6	6
l	7 (x)	7	7 (bl)		7
h	8 (y)	8	8 (tl)		8
e^-	12	11	11	11	11
ν_e	11	12	12	12	12
μ^-	14	13	13	13	13
ν_μ	13	14	14	14	14
τ^-	16	15	15	15	15
ν_τ	15	16	16	16	16
χ^-		17			17
ν_χ		18			18

FIRST PROBLEM:

- PDG particle numbering scheme deliberately leaves room for interpretation

- PDG only actually defines numbers for observed particles

- any standard scheme must allow a consistent set of particle ID numbers for actual use

standard generator interface

- menu driven?
- select general event type
  - fixed target
  - collider
  - nuclear
- select particle type and energy
- select specific event type
- select generator

- Fixed Target
  - beam particle and energy
  - target particle
  - event type
    * minimum bias
    * QCD heavy quark production
    * ...

- Collider
  - beam particle and energy
  - beam particle and energy
  - event type
    * B Bbar
    * continuum
    * ...
- Nuclear

event generator selection

- e+ e-
  - HERWIG
  - PYTHIA
  - QQ
- fixed target
  - JETSET
  - ISAJET
  - HERWIG
- Nuclear
  - FRITIOF

finally ask for specific parameters

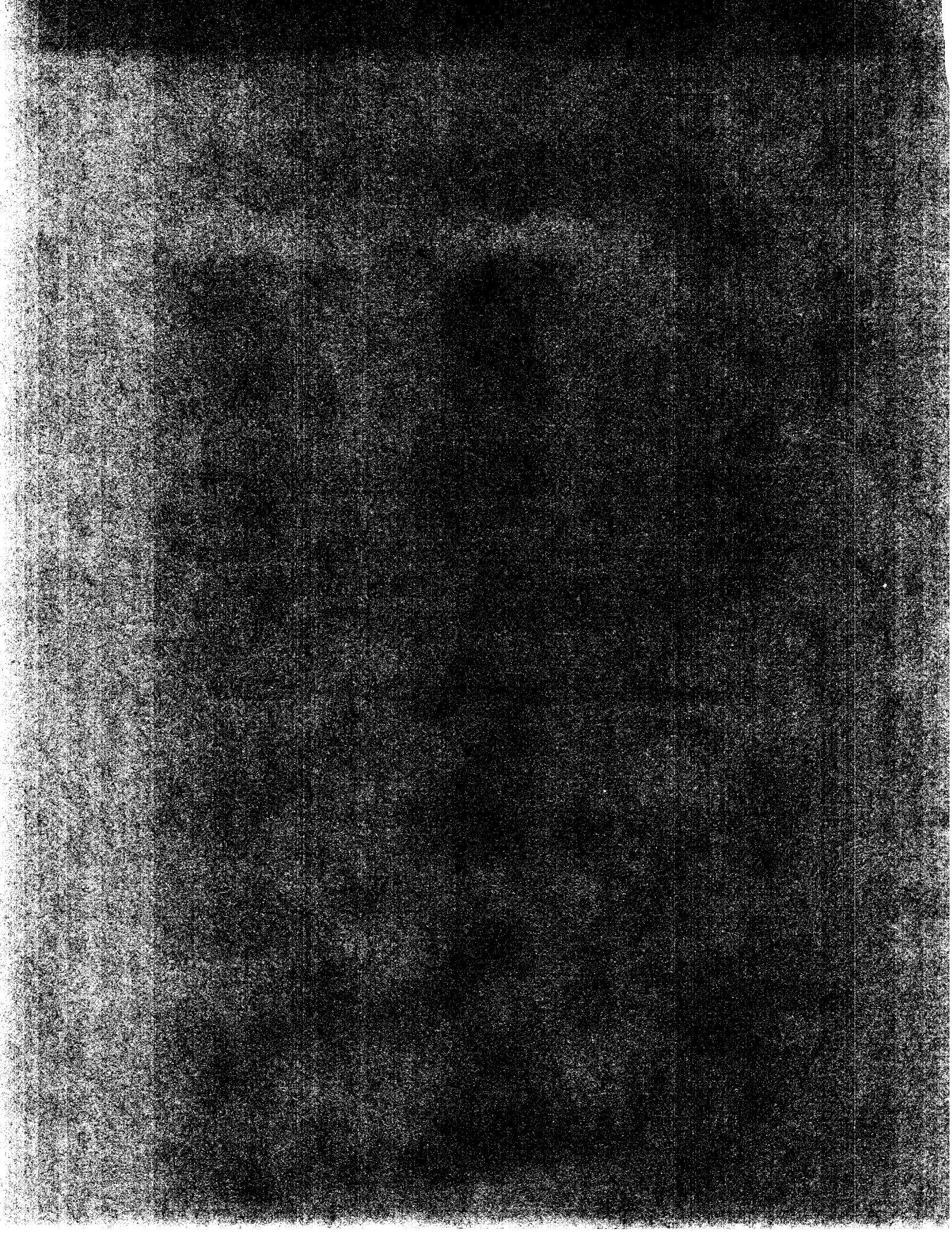
- menu driven
- read input files
- standard event types should have well-defined defaults

CONCLUSIONS

- 1987 physicist ??????
- 1991 physicist ???!!!
- future physicist !!!!!!

"CERNLIB Maintenance"/"From Patchy to CMZ"
"Installation Servers"/"Improving the Test Phase"
"Information Exchange"/"CERN List Servers"
"The ASIS World-wide CERNLIB Server"
"The WWW Project"

M. Marquina, T. Berners-Lee, R. Caillian

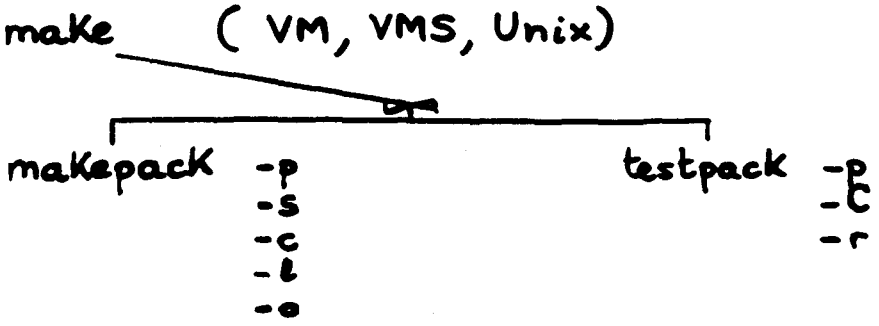


MAINTENANCE + DISTRIBUTION OF INFORMATION OF CERN Lib

- IMPLEMENTATION mechanism
- AREAS OF IMPROVEMENT
- INFORMATION EXCHANGE

M.A Marquina /CERN

* Implementation is local




```
make minuit    { makepack -p minuit
               { makepack -s minuit
               { makepack -l packlib -c minuit

make pdflib    { makepack -p pdflib
               { makepack -s -c pdflib
               { makepack -l pdflib pdflib

make pawX11    { makepack -p pawm
               { makepack -s -c pawm
               { makepack -o pawX11 pawm

make test_minuit { testpack -p minuit
                 { testpack -c minuit
                 { Testpack -r minuit

make test_zebmz1 { Testpack -p zebra zebmz1
                 {          -c
                 {          -r
```

* IBM : service machine { implementation
                        { BITNET distribution
                        { PAM delivery to all other CERN systems

Areas of improvement

* Move to a modern SCM system	One Solution: CMZ
* Trigger installation centrally	ISIS
* Absorb installation dependencies + doc in the distributed file	CMZ
* Improve testing phase – - homogeneity - completeness - simplification (OK - $\overline{\text{OK}}$ answers)	close coll. with developers
* Develop good human interfaces to PL programs for easing its use	more staff
* Define a proper and consistent user environment (UNIX) to access soft...	HEPiX?

Information Exchange

* Information newsgroups — HEPMGR @ CERNVM
* Discussion newsgroups — HEPLIB @ CERNVM

HEPDB @ CERNVM
LGEANT
LPAW
LFATMEN

* The ASIS server (Applications Software Installation Server) is being designed to distribute implemented soft:
  - dispatch CERN Software (CERNlib + $\frac{\text{CERNSoft + PD}}{\text{only at CERN}}$) to requestor's machine
  - logs requests for automatic reporting/updating of new releases
  - hosts CERNlib releases for distribution via controlled anonymous ftp
  - communication via e-mail

* The www is designed to access documentation/information in a distributed environment
  - HyperText based (NeXTStep, linemode versions)
  - HTML tags marking references to doc { local / remote

The WorldWideWeb Project

In this short presentation we will cover:

- HyperText in general
- A summary of the project – People
- How we inform ourselves
- The WWW architecture
- Development Status
- The NeXT HyperText editor
- The Line Mode browser
- The CERNVM "FIND" gateway

Tim Berners-Lee, Robert Cailliau, CERN February 19

The WorldWideWeb Project

The WorldWideWeb (WWW) project aims to create a distributed information system covering a wide range of uses . At CERN it must operate with:

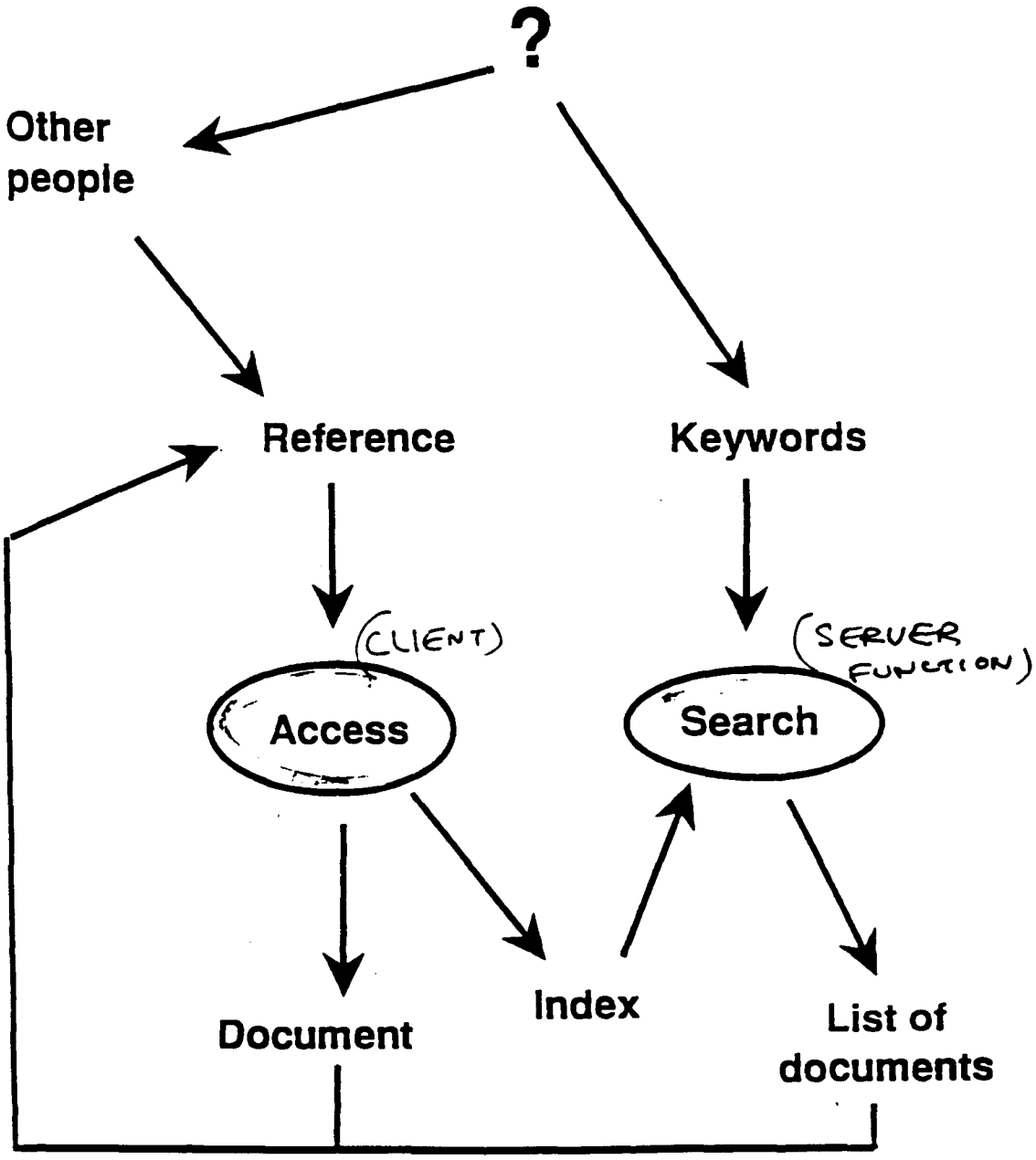
- A Heterogeneous distributed environment.
- A Multitude of document formats.
- Access by reference or keywords.
- Existing data in existing information systems

Key concepts used by WWW to acheive this are:

- Browser/Server independence
- Format negociation
- Virtual Hypertext

Tim E

How we inform ourselves



This is a general model. WWW follows it, automating each stage.

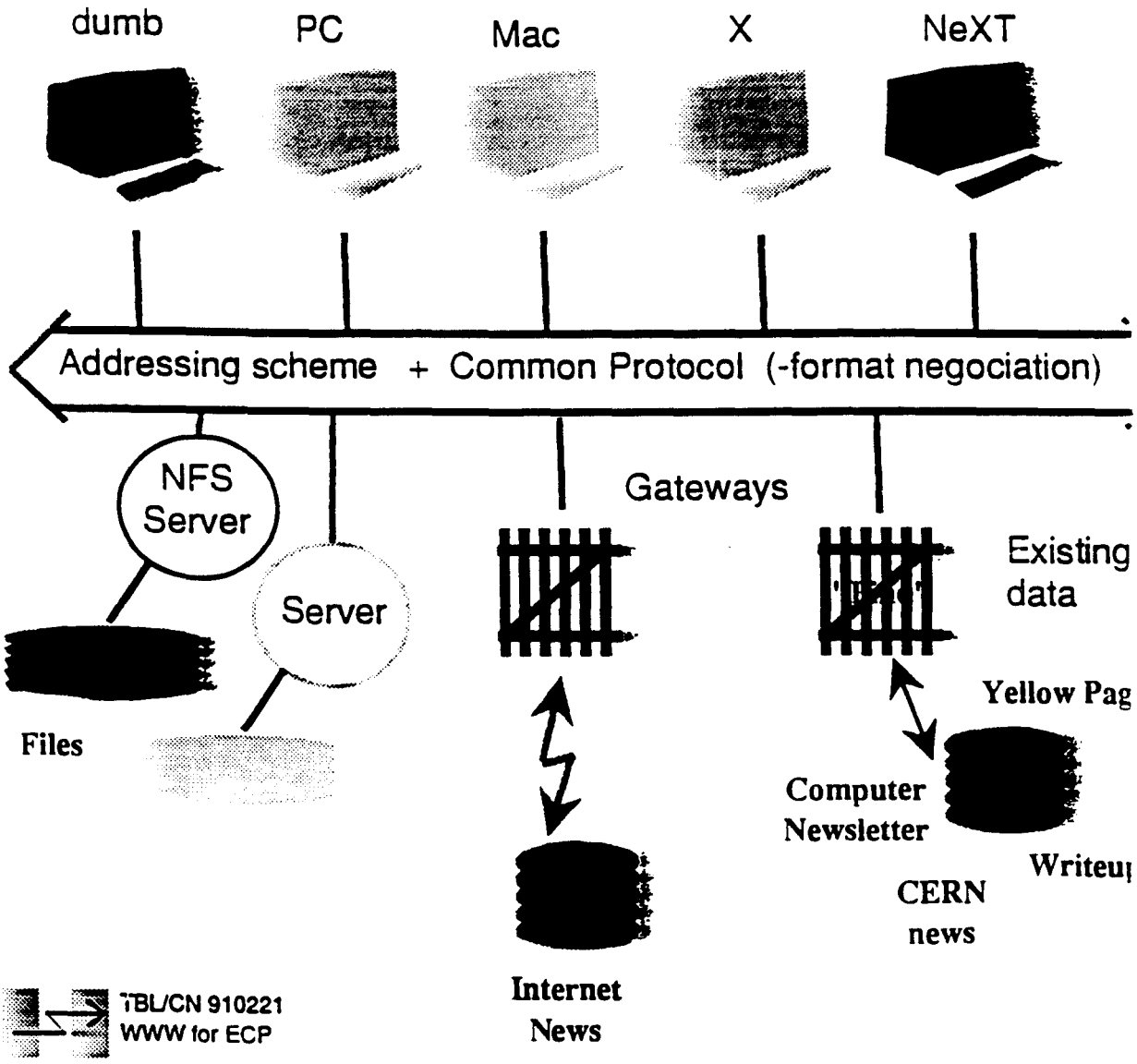
Some Technical bits

Architecture	for wide-area hypertext information systems
Addressing	How to name a document
Protocols	How to get a document
HTML	The transfer format for hypertext

(Overview)

Tim B.

Development Status:



Addressing

access : // host / path / path # id

access	defines the name space for the rest of the address: http, file, news, (etc...?)
//host	(For most name spaces) Host machine
/path	Hierarchical, unix convention. Allows relative names.
id	Specifies a section of a document ("Anchor").

Most of the name if omitted, assumed same as refering document.

(search)_____

Tim Bl

Addressing: Index Search

If an index search is required, the format is

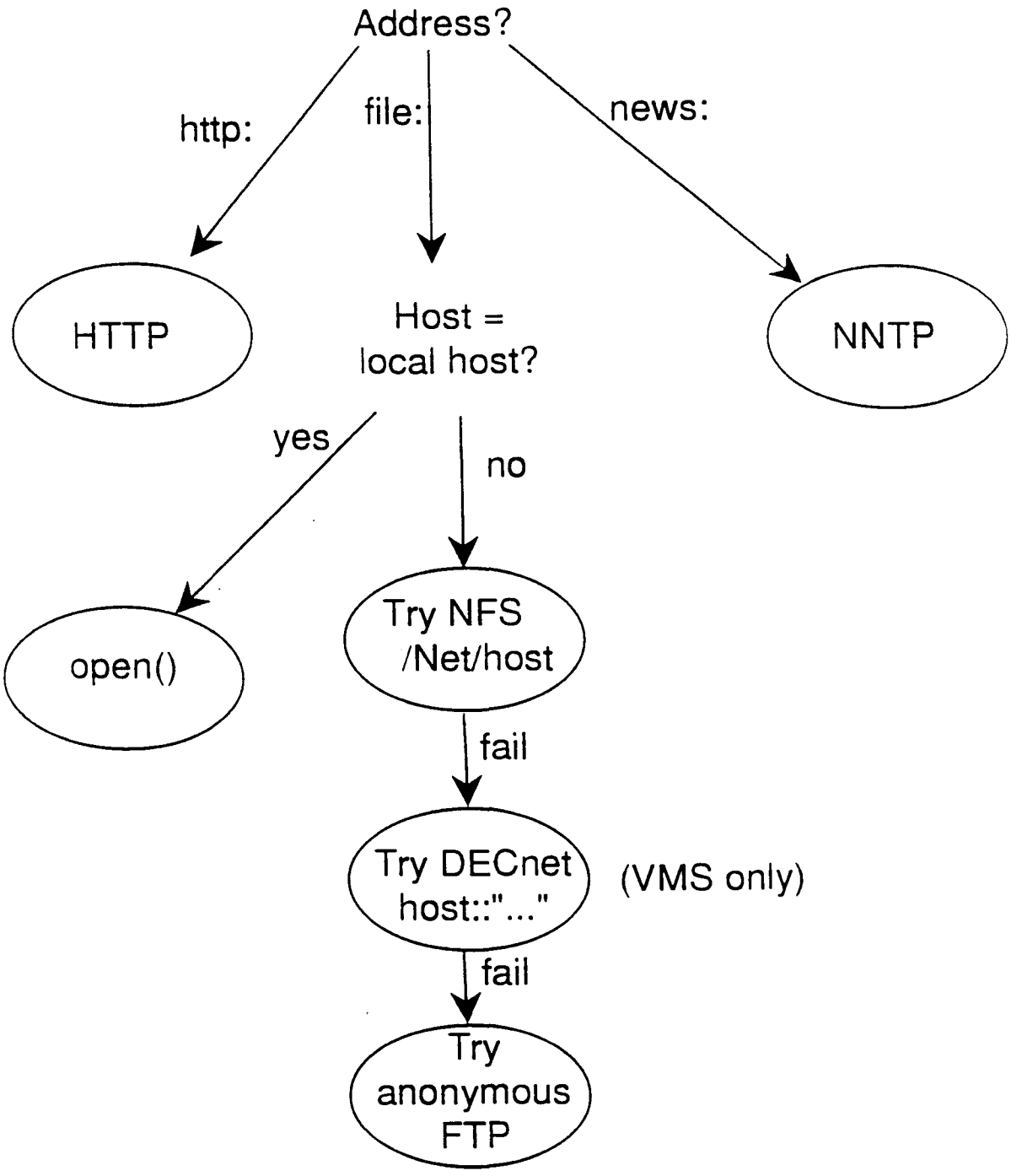
indexname ? keyword+keyword+keyword

indexname	The full address of the index document for a particular type of search.
?	A search operator
keyword	Picked up from the user
+	Currently implies "and" search semantics.

(next)

Tim Bl

Protocol Choice



HTML

HTML is a particular SGML document type - set of tags .

- SGML syntax is not beautiful
- SGML is high level
- SGML is a standard is hypertext circles

We use, for example, a special anchor tag:

```
<A NAME=id HREF=address> text of reference </A>
```

This is used for all the links.

(back)

Tim BL

```
<TITLE>/usr/local/lib/WWW/default.html</TITLE>
<NEXTID 7>
<H1>Select by number</H1>
<DL>
<DT><A NAME=0 HREF=QuickGuide.html>Help</A>
<DD> About this program
<DT><A NAME=1 HREF=http://crnvmc/FIND>"FIND" index</A>
<DD>Index of CERN computer centre documentation, newsletters, news,
IBM and Cray help files, etc...
<DT><A NAME=2 HREF=http://crnvmc/FIND/yellow?>Yellow Pages</A>
<DD> An analytical telepnone directory of CERN
<DT><A NAME=3 HREF=http://crnvmc/NEWS/cern>CERN news</A>
<DD> News from the CERN User's Office
<DT><A NAME=4 HREF=http://crnvmc/NEWS/vmnews>VMNEWS</A>
<DD> CERN computer center news
<DT><A NAME=5 HREF=http://info.cern.ch/hypertext/DataSources/NewsFromVM/Overview.html>C
r CERN news</A>
<DD> A list of public groups. Private groups also exist.
<DT><A NAME=6 HREF=http://info.cern.ch/rpc/doc/User/UserGuide.html>CERN RPC</A>
<DD> The user guide for the RPC system developed in CERN CN division
(not Sun/RPC). This is an example of documentation (partially) converted
into hypertext.</A>
</DL>
```

Format Negociation

It is negociation between the browsing client and the data supplying server which allows us to cope with the large, growing, number of data formats.

Browser:

Formats acceptable, with penalty points for speed and degradation.

Server:

Data in the best format which is appropriate.

The result is an optimisation of the delay and the degradation of the data, using whatever conversion facilities exist.

Tim

On the NeXT

The prototype browser/editor was written for the NeXT machine, using the NeXTStep human intreface software.

Why the NeXT for prototyping?

- Large reusability of NeXT Human Interface libraries using Objective-C
- Much higher functionality than Motif, for example
- Fast prototyping with the Interface Buildier

Functionality of prototype:

- Full Browser with keyword search, backtracking, etc
- Editing and link creation
- File, NNTP and HTTP protocols supported.
- Import/exprort of HTML, RTF and plain ascii files.
- (More...)

TBL, RC Feb 1

Gateway to FIND

- Around 5000 existing documents, including newsletters, help pages from IBM, CERN and Cray, VM news articles, CERN yellow pages, CERN people (to come...).
- Server provides keyword search facility.
- Several different indexes.
- Data mainly in plain text, some SGML, Postscript, etc.

TBL,

The Line Mode Browser

In order to cover the very large number of computer users who still do not have windows, Nicola Pellow has written a browser for dumb terminals.

- Coventional command-line interface
- Numeric references [1] as in papers
- Same world as other browsers

TBL, RC Feb 1991

Line Mode Browser

The line mode browser allows you to find information by following references and/or by using keywords.

References are numbers in [brackets] after particular phrases. Type the number and RETURN for more information on the phrase.

Some documents are indexes. These contain little text, but allow you to search for information with keywords. Type "k" (space) and the keywords. For example, "k sgml examples" searches the index for items with keywords SGML and EXAMPLE. You can only use the "k" command when it is present in the prompt.

See also other commands , command line syntax , shortcuts , more .
Please send any bugs and suggestions to timbl@nxoc01.cern.ch.

Tim E

QuickGuide -- /LineMode

Line Mode Browser

The line mode browser allows you to find information by following references and/or by using keywords.

References are numbers in [brackets] after particular phrases. Type the number and RETURN for more information on the phrase.

Some documents are indexes. These contain little text, but allow you to search for information with keywords. Type "k" (space) and the keywords. For example, "k sgml examples" searches the index for items with keywords SGML and EXAMPLE. You can only use the "k" command when it is present in the prompt.

See also other commands[1] , command line syntax[2] , shortcuts[3], more[4]. Please send any bugs and suggestions to timbl@nxoc01.cern.ch.

Tim Bl

[End]
<ref.number>, 'Back', 'Quit', or H for help:

10) www

Select by number

Help[1]	About this program
"FIND" index[2]	Index of computer centre documentation, newsletters, news, IBM and Cray help files, etc...
Yellow Pages[3]	An analytical telephone directory of CERN
VMNEWS[4]	CERN computer center news
STING news[5]	News from the Software Technology Interest Group
CERN RPC[6]	The user guide for the RPC system developped in CERN CN division (not Sun/RPC). This is an example of documentation (partially) converted into hypertext.

[End]

<ref.number>, 'Quit', or H for help: 2

This is the general CERN public document index.

Use a keyword search to get documents, help files, etc.

[End]

K <keywords>, 'Back', 'Quit', or H for help: SGML

200

```
PUBLIC disks

  INFO= Miscellaneous INFOrmation
    GKS  GKS Graphics programming in VM-CMS[1]
    MACTOOLS  VM/CMS utilities for MacIntosh[2]
    SGMLEXAM  SGML text processor examples[3]
    TPHELP  Help with Text-Processing Products and Problems[4]
  CMS= CMS commands, execs and modules
    APOLA67  Preparation of an APOLLO bit map for use with SGML[5]
    BOOKIE  SGML formatting using SGMLDCF, BookMaster and SCRIPT/DCF[6]
    CONVSGMLprogram  to convert text-processing files[7]
    GRCONV  converts GKS metafiles to METAFILE POSTSCRIPT TEKTRONIX SGML[8]
    GRVIEW  To VIEW or EDIT a GKS metafile on a graphics terminal[9]
    INDXFILE  creates input file for MAKEINDX XEDIT[10]
    MAKESGML  helps in the choice of SGML layout and creation of file[11]
    PREPRINTPreprint  transmission to div secr & printshop[12]
    SGML  SGML formatting using Waterloo SCRIPT[13]
    XSCRIPT  executes Waterloo SCRIPT[14]
  CNL= Computer NewsLetter articles (CERN)
    200_2__1  Text Processing: Installation of TeX 3.0 on all central
    systems[15]
    200_2__2  Text Processing: LaTeX Developments and News[16]
K <keywords>, <ref.number>, 'Back', <RETURN> for more, or H for help: 12
```

Preprint transmission to div secr & printshop

PREPRINT

Use the PREPRINT command to send details of a preprint to your divisional secretary. Subsequently the preprint will be read by the Division Leader or his representative and, if approved by him, will be sent electronically to the CERN printshop for printing.
For this exec file to work the following fields MUST be filled in:

Division
Contact Person, Phone Number and Contact Userid
These are by default your name, your phone number and your Userid. However, if you are likely to be absent from CERN for a long preprint has been printed, you should provide details of some person who can be contacted if necessary during your absence
As much of the title of the preprint as fits on the line
The name of the first author in the list of authors
The name of the mark-up language, LaTeX (the default), TeX or SGML
The date by which internal distribution should take place
The name of the input file to be processed.

'Back', <RETURN> for more, 'Quit', or H for help:

11) yellow tax

YELLOW PAGES (ANALYTICAL PHONE DIRECTORY)

FI= FINANCE DIVISION

2777 F-CF Financial accounting section:

2777 F-CF - Statements. budgets, taxes

PE= PERSONNEL DIVISION

AS Administrative and Social Services:

2855 AS - Income tax refunds

[End]

K <keywords>, 'Quit', or H for help:

Availability

Code is available now by anonymous FTP from info.cern.ch [128.141.201.74], directory /pub.

Line Mode Browsers for (almost) any machine. (also HyperText editor for NeXT)

Get a browser, select "Help" then "More" to read all project details from our server. CERN also has a server for all computer centre documents, newsletters, CERN program library documentation, etc. etc.

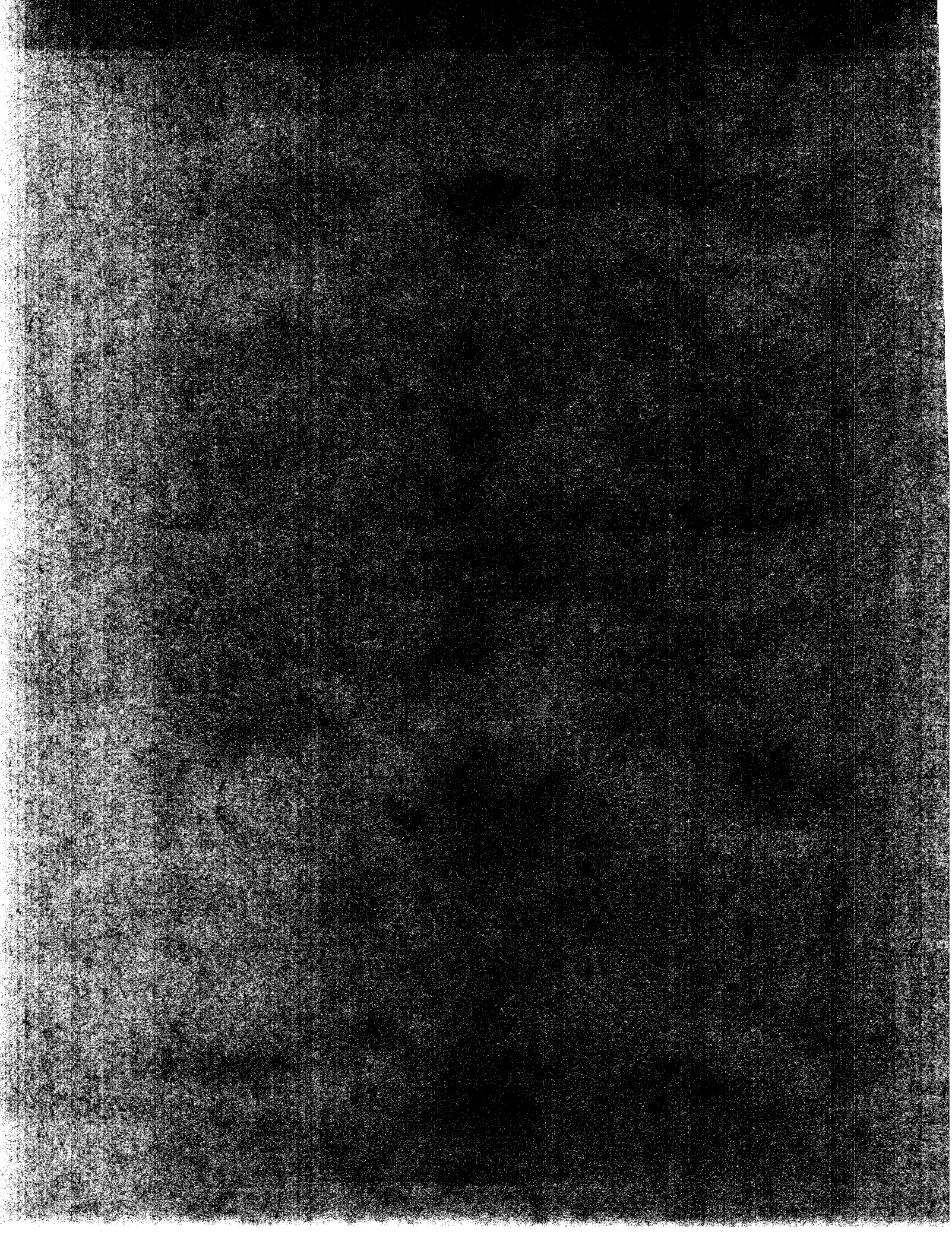
Anyone can make data available using an FTP server or the HTTP server (available). Several other sites are setting up servers.

Please mail timbl@info.cern.ch. with any questions.

Tim B.

"Distributed Data Management for High Performance Data Systems"
"IEEE Storage Standards"

B. Coyne



HEPLIB'91 Users Meeting

High Performance Storage Systems

Robert A. Coyne

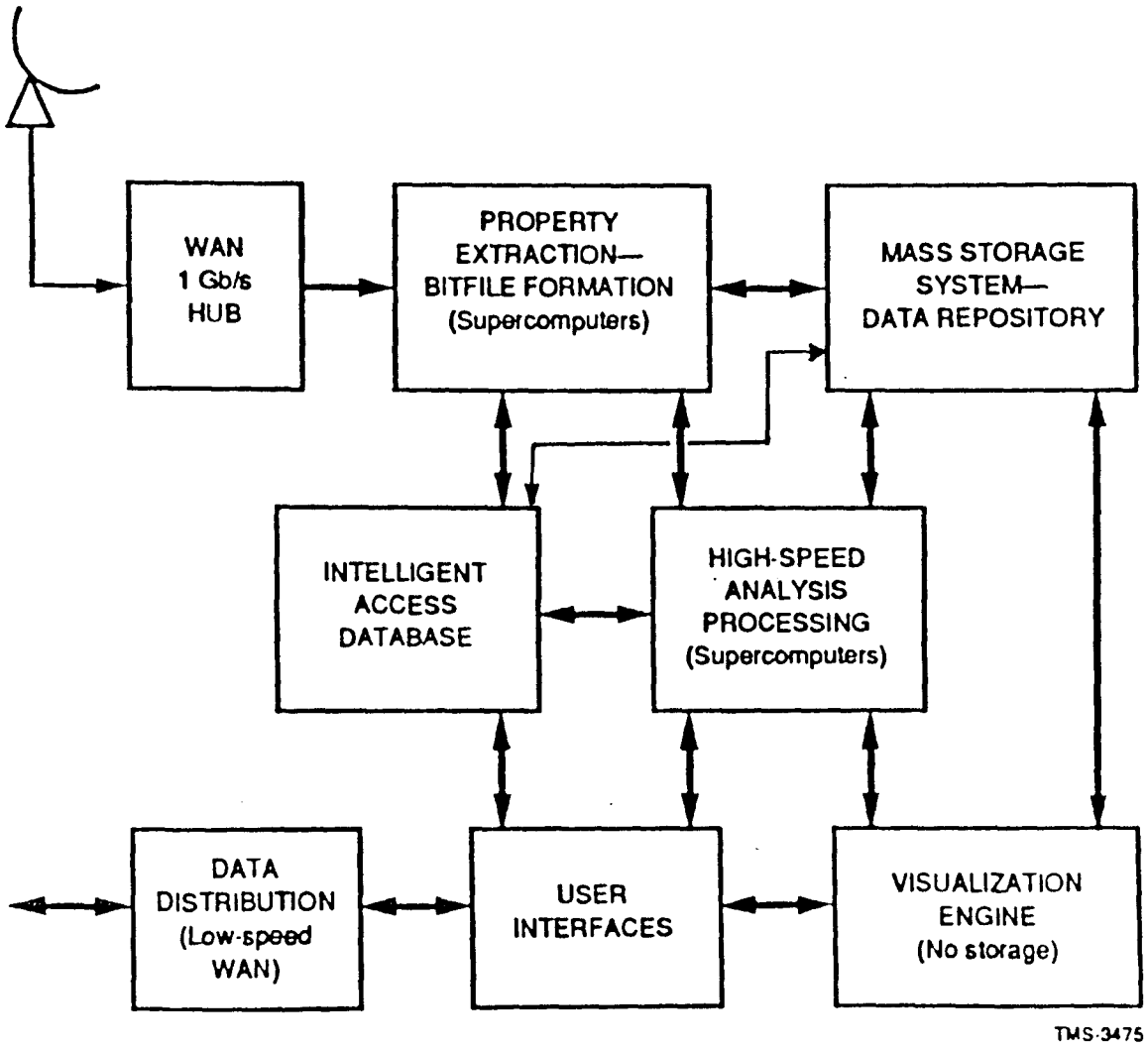
IBM Federal Sector Division
3700 Bay Area Blvd
Houston, Texas 77058-1199
coyne@houvmscc.vnet.ibm.com
(713)-282-7274

September 20, 1991

IBM

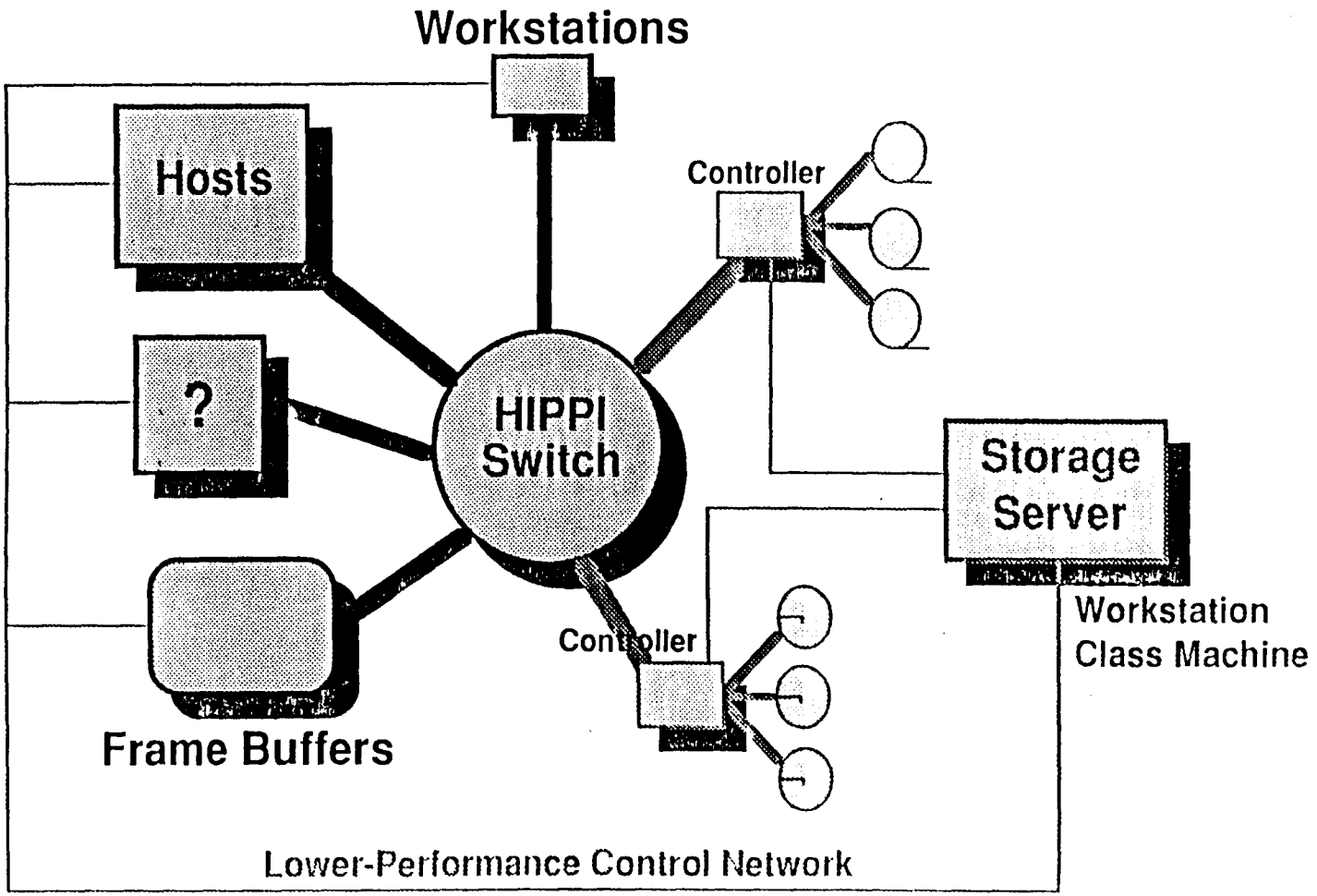
Independent Research and Development

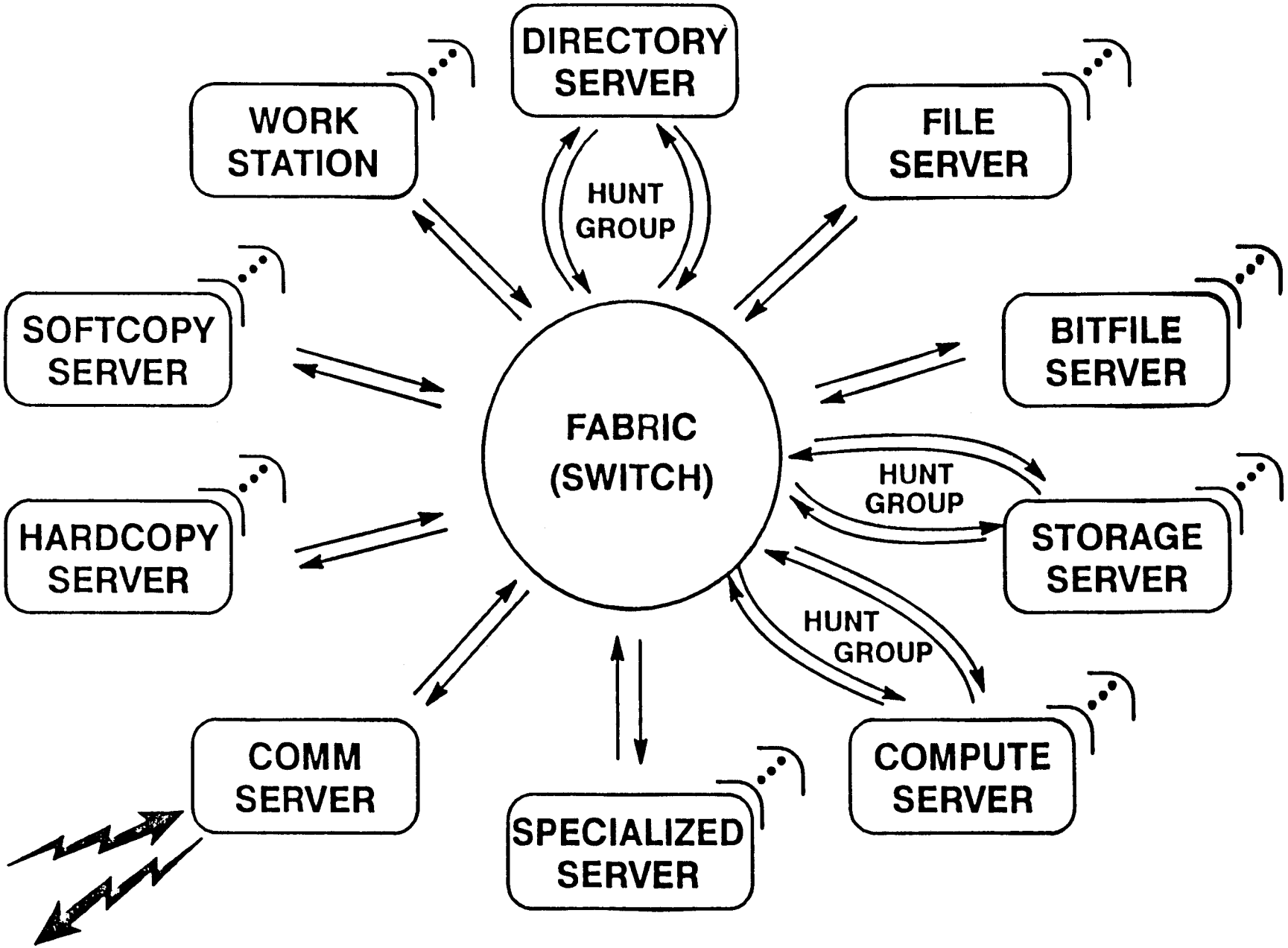
- Environment:
  - The next generation of data management centers must be able to process data at much higher rates
  - Data management centers must be able to archive and retrieve much larger quantities of data (128 kbs vs one LOC per month)
  - The government requires industry standard solutions incorporating components from multiple vendors
- Objective: Research and develop technology, systems, architectures and methods required to implement distributed, high performance information systems related to DoD, DoE and NASA programs



GROUND TECHNOLOGY FUNCTIONALITY

High Performance Storage Architecture

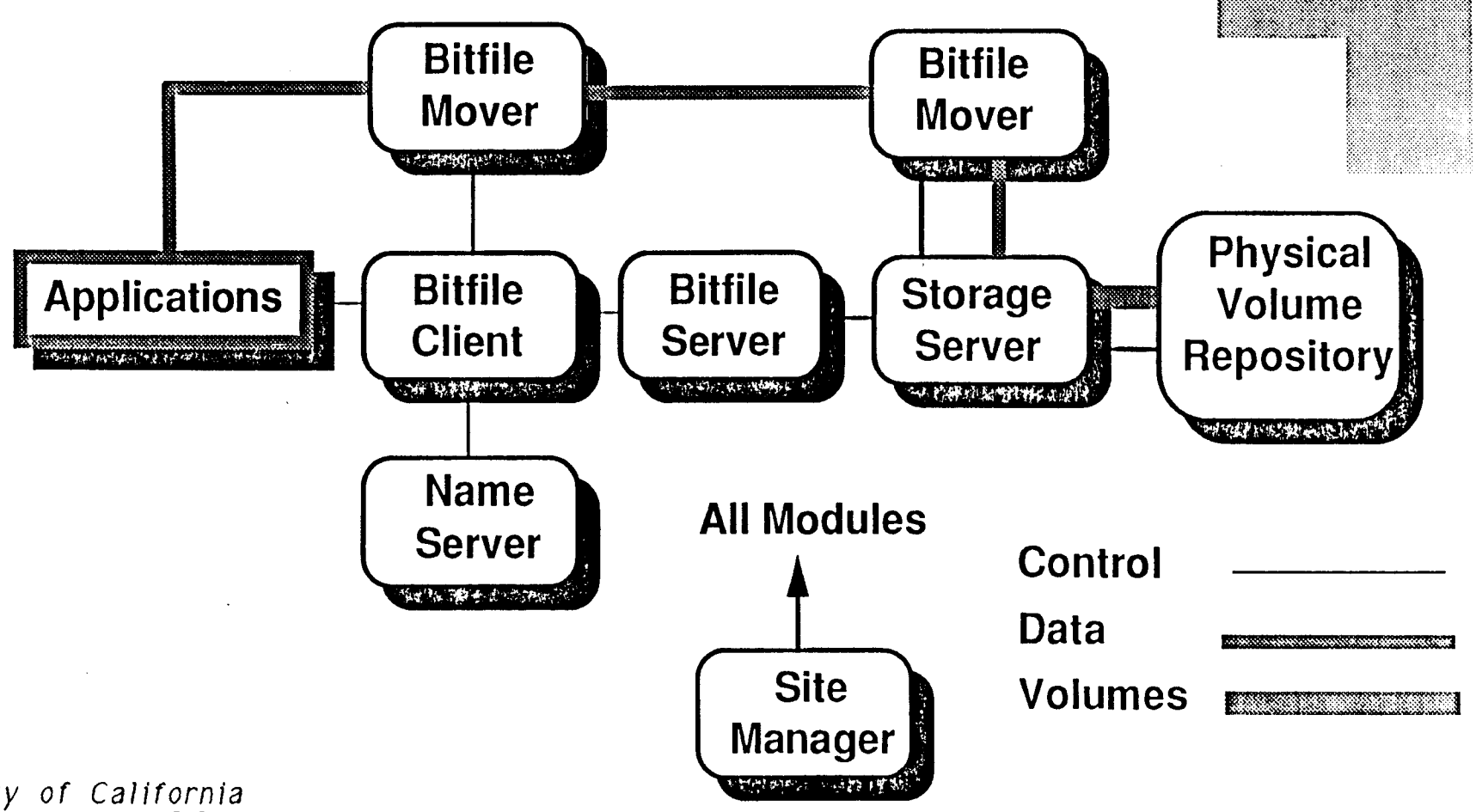



DIRECTORY SERVER
WORK STATION
FILE SERVER
HUNT GROUP
SOFTCOPY SERVER
BITFILE SERVER
FABRIC (SWITCH)
HARDCOPY SERVER
HUNT GROUP
STORAGE SERVER
HUNT GROUP
COMM SERVER
SPECIALIZED SERVER
COMPUTE SERVER

Moving Toward Storage Software Standards

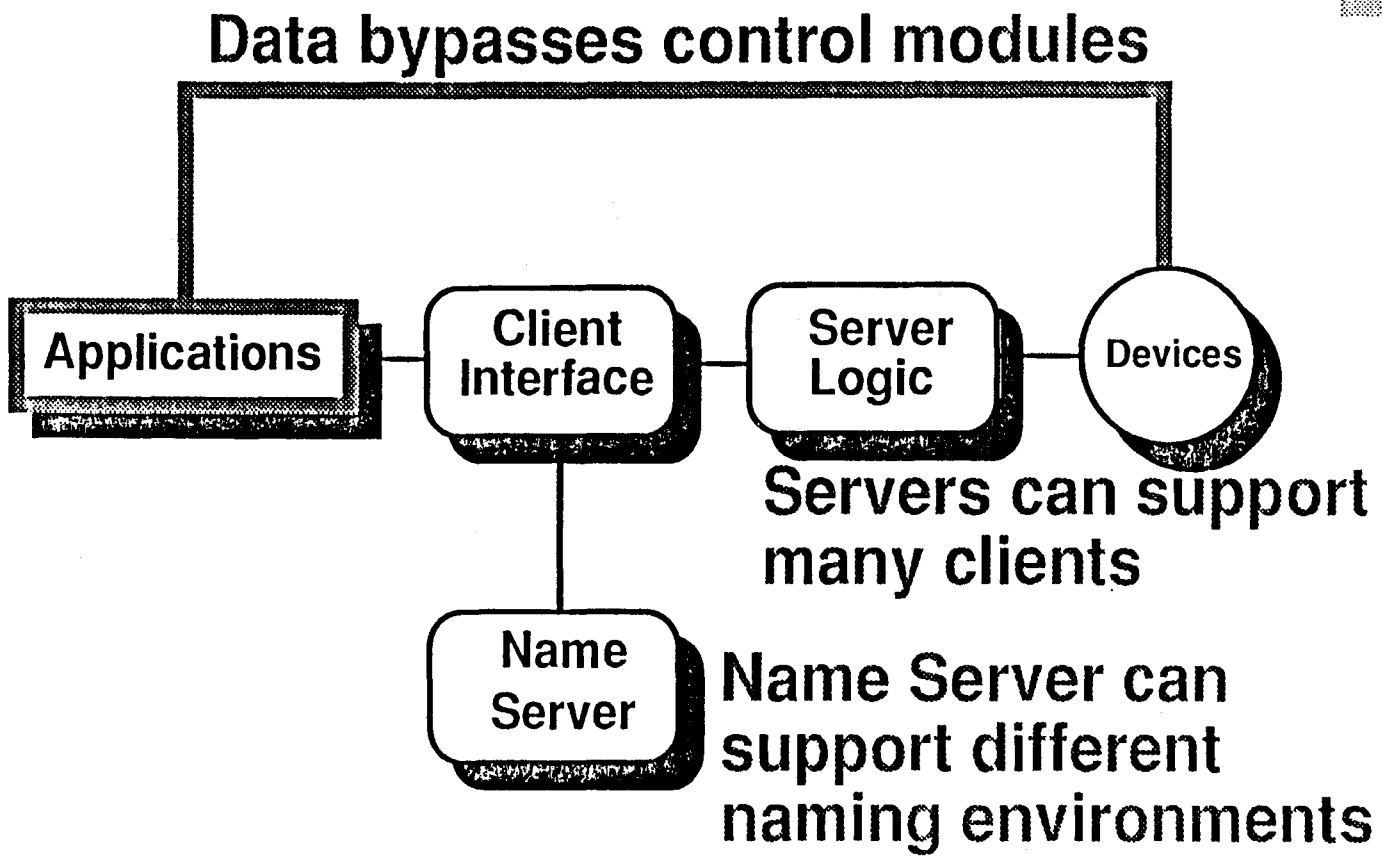
- Industry moving toward Unix / Posix
- IEEE Storage Reference Model developed
  - Stimulated discussion of storage issues
  - Promoted development of standard, modular products
  - Spawned new standardization effort (IEEE project 1244)

A Simplified View of the Model



University of California
Lawrence Livermore
National Laboratory

Significant Modularity

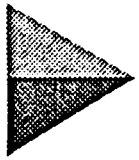


Working Group Participants

- Amdahl, Ampex, AT&T, Convex, Cornell, Cray Research, CSC, DARPA, Datamax, Datatape, DEC, Discos, Epoch, E-Systems, HP, Hughes, IBM, Kodak, LANL, LBL, LLNL, Mitre, Mobil, NASA, NCAR, NSA, Odetics, Ohio Supercomputer Center, Metrum, NERSC, NCSC, Sandia, Shell Development, SRI, STK, STX, Supercomputer Research Center, TRW, University of Michigan

Goals for the Standards

- **Be applicable to**
  - Small to large systems
  - Stand-alone to distributed systems
  - Various storage devices, networks
  - Many application interfaces
- **Maintain the separation of**
  - Control and data
  - File Naming and file storage



The UniTree™ Implementation of the Model

- **Developed at LLNL in parallel with the model**
- **Marketed by General Atomics** (Discos Division)
- **Runs on AIX, UTS, Ultrix, SunOS, and others**
- **Bitfile Clients for FTP and NFS** (file server switch in progress)
- **Berkeley 4.3 Semantics** (V.4 and Posix future options)
- **Multi-tasked servers** (requires asynchronous I/O)

To Complete the Picture

Attend the 11th IEEE Mass Storage Symposium

October 7-10, 1991

Monterey Sheraton Hotel, Monterey, CA.

Arranged by

Bernie O'Lear
National Center for Atmospheric Research
P. O. Box 3000
Boulder, Colorado 80307

Summary of Important Issues for Future Storage Systems

- High-performance architectures
  - Network-attached devices
- Device striping technology
- Distributed, hierarchical software architectures
- Software standards
  - IEEE Storage System Standards Working Group
- Open Systems
  - Posix

University of California
Lawrence Livermore
National Laboratory

Storage Systems and Related Requirements

- Transparency vs Application Optimization Schemes
  - Designers should consider common transparencies
    - access, concurrency, data representations
    - execution, fault, identity, location, migration
    - naming, replication, semantic, syntactic
  - Designers should consider application defined schemes
    - location, migration, performance

Storage Systems and Related Requirements ...

- **Functionality**
  - Dynamic migration hierarchies
  - Storage Server/Mover options
    - data movement is scheduled
    - aligned and unaligned transfers
    - pre-allocate storage for high speed data transfer
    - user selected level of service

Storage Systems and Related Requirements ...

- Functionality

  - Monitor performance and alter operating parameters

    - modify location scheme for performance

    - modify migration hierarchy for performance

File Server and Related Requirements ...

- Portability and Standards
  - Posix and OSF/Unix
    - IEEE File Server as selectable Unix file system
  - Transparent network interface for messages and bitfile moving
    - file Transfer - FTP, FTAM, Posix, Light Weight Protocols (eg XTP)
    - file Access - NFS, AFS, IEEE SSSWG defined, Posix
    - message Passing - TCP/IP, GOSIP, XTP
    - allow vendor specific protocols (eg DECNET, SNA)

File Server and Related Requirements ...

- External Interfaces
  - Interaction with network job/process scheduling
  - Interaction with application name servers, DBMS or OOBMS

WOCE-CME Data System Prototype: Objectives

- Develop a Reorganization Technique for WOCE-CME Model Data That Allows for More Efficient Access by Users
- Reorganize a Sample of Model Data and Demonstrate the Improved Accessability
- Investigate Software and Procedures Needed for Efficient Remote Access of Model Data Sets
- Develop Data Interfaces with a Visualization Tool, and Demonstrate with the WOCE-CME Data
- Help the scientific community gain better access to the WOCE-CME model data, and thereby promote investigative use of the data.

The Project Proceeded in Three Phases

Phase 1

Develop a Reorganization Technique for WOCE-CME Data

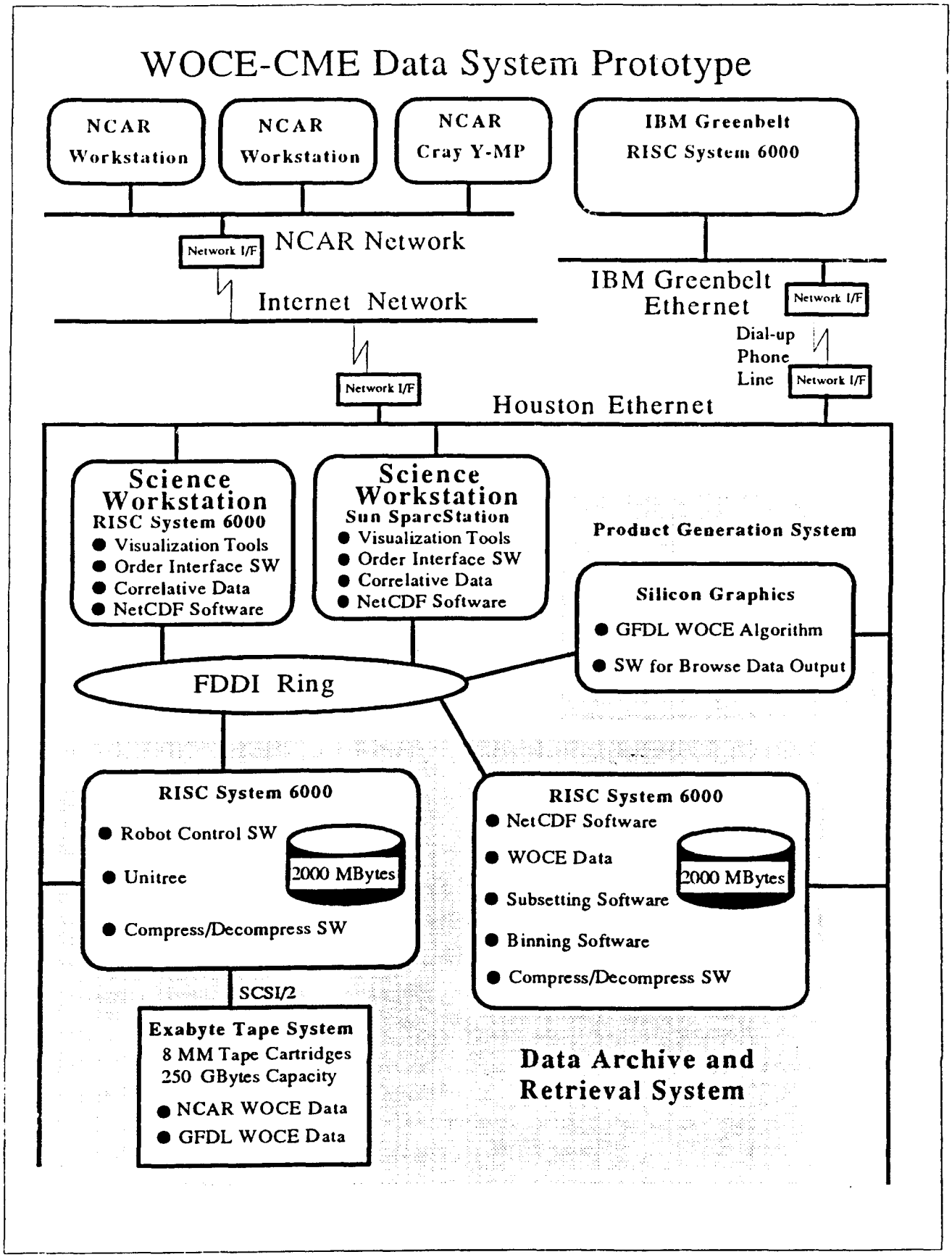
Reorganize a Sample of the Model Data

Phase 2

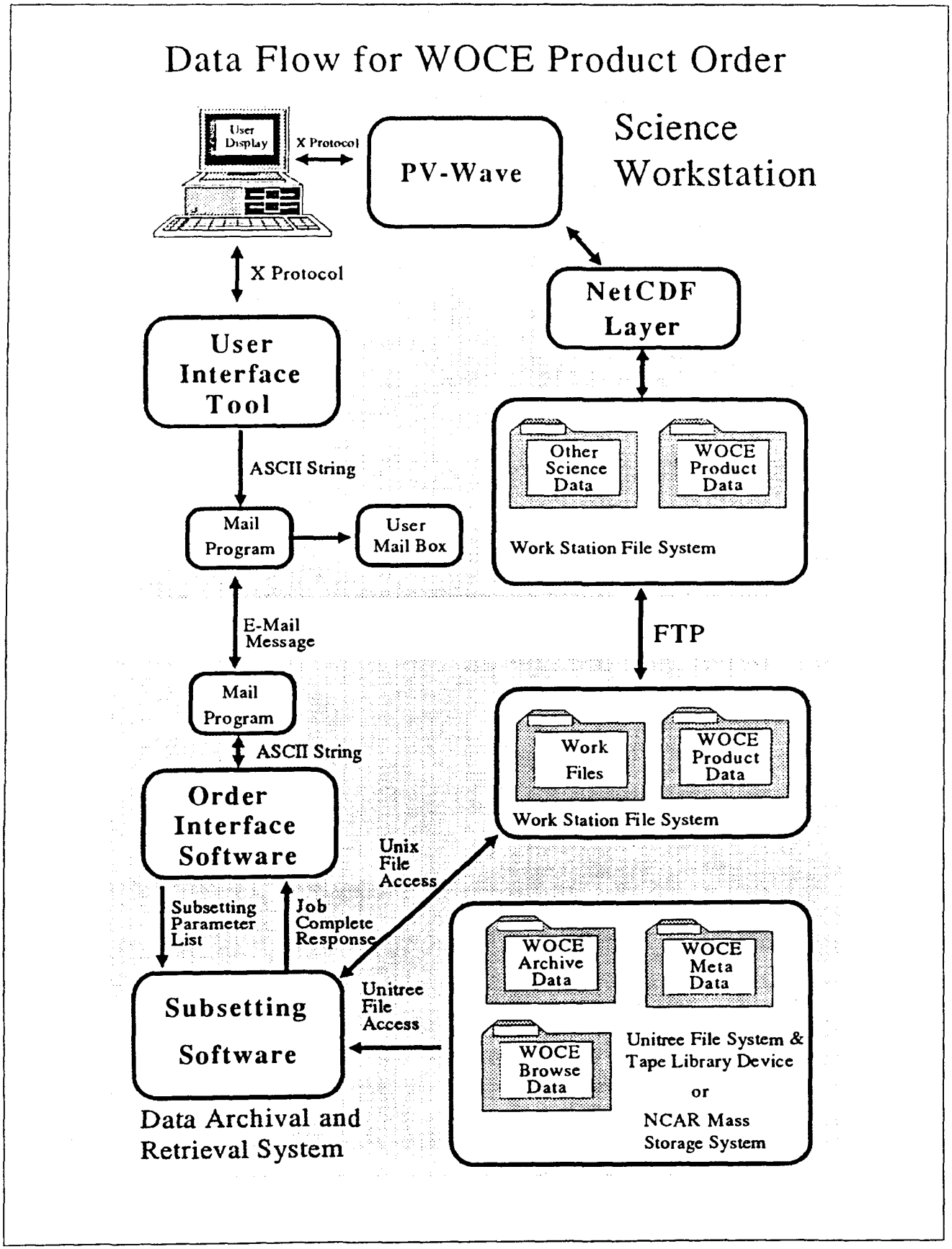
Investigate Efficient Remote Access Capabilities

Phase 3

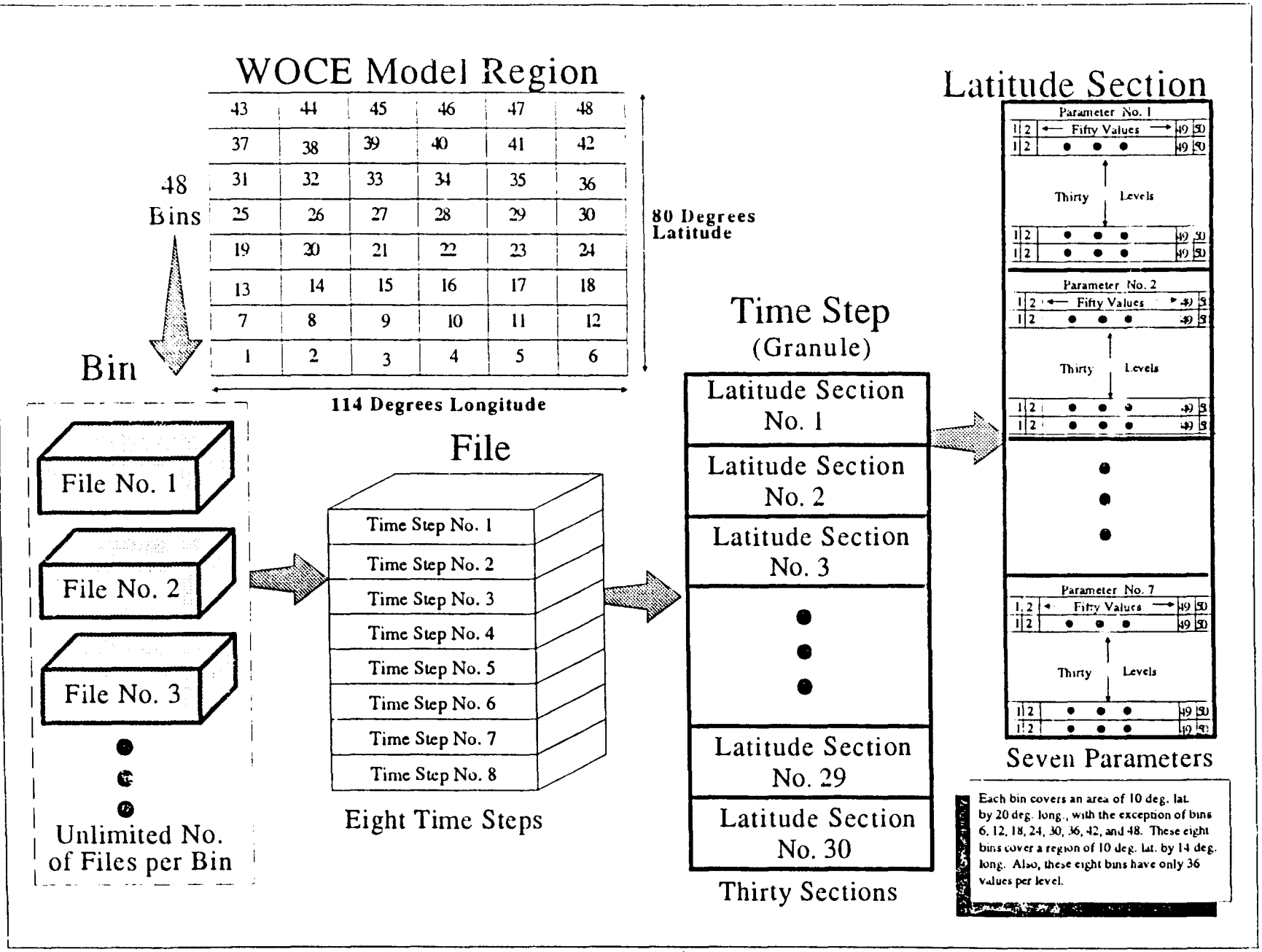
Demonstrate Data Visualization

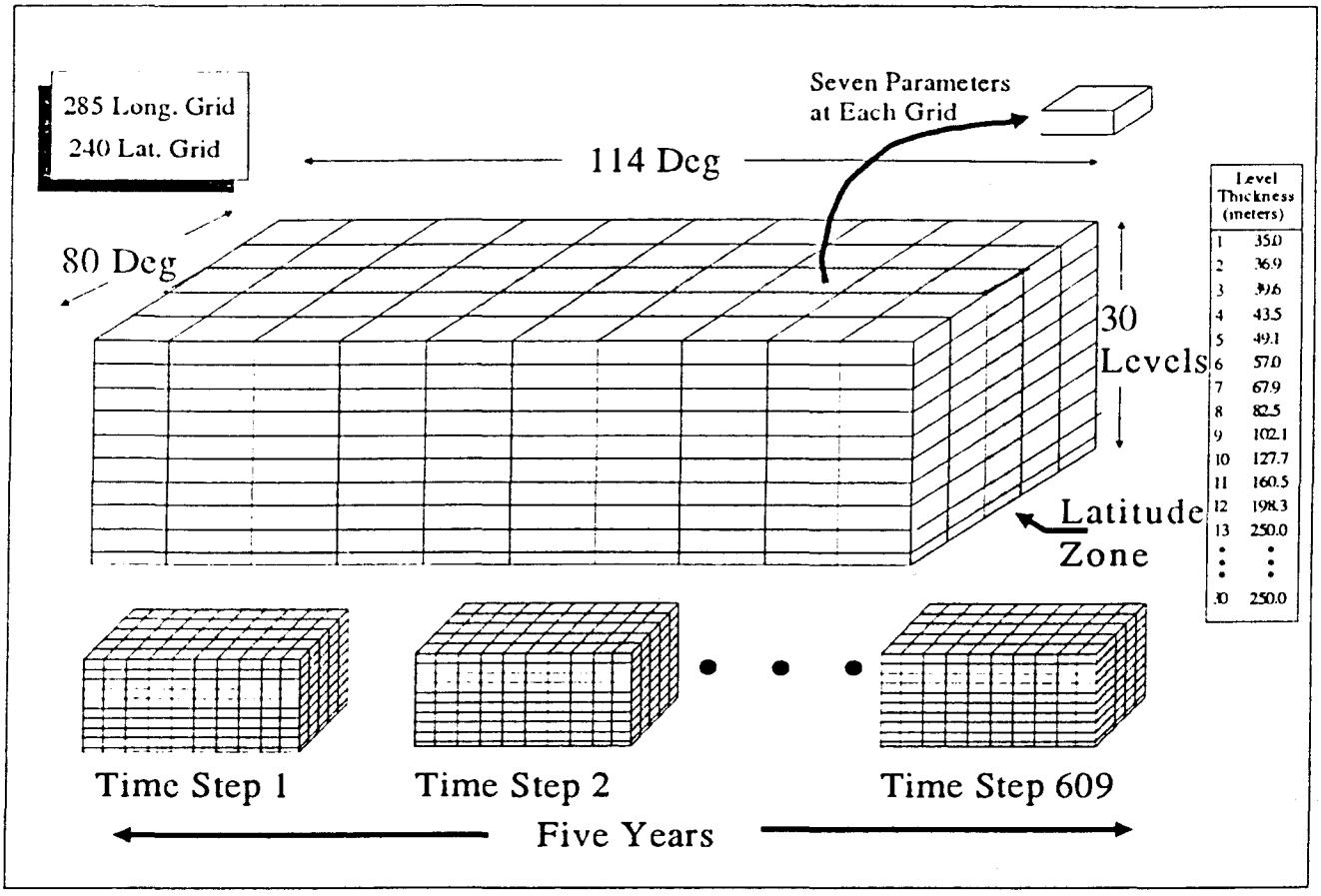


Overview of WOCE-CME Data System Prototype (FDSYSTEM)

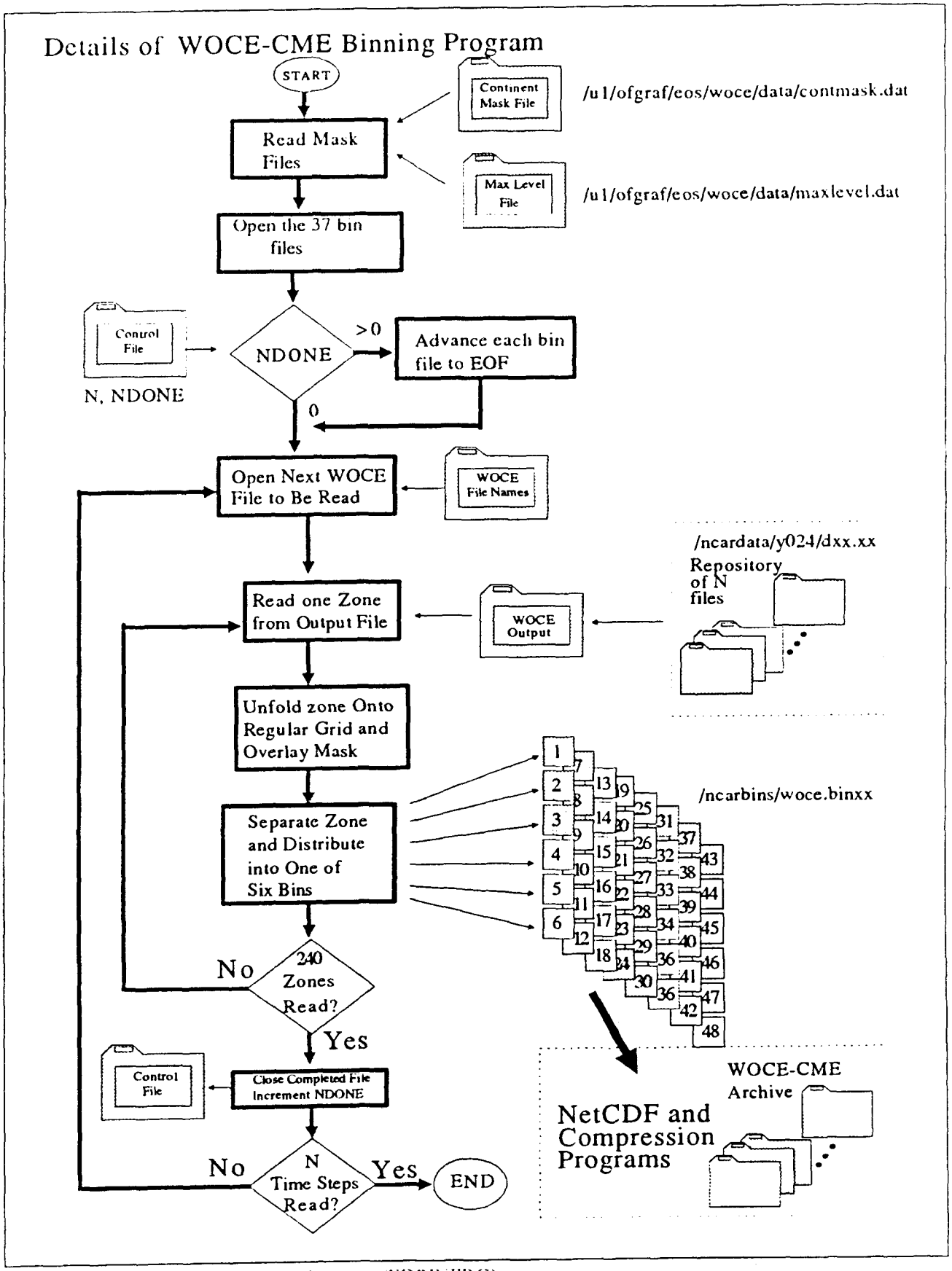


Data Flow Diagram for WOCE-CME Product Order (FDDATFL1)


WOCE Model Region
43 44 45 46 47 48
37 38 39 40 41 42
31 32 33 34 35 36
25 26 27 28 29 30
19 20 21 22 23 24
13 14 15 16 17 18
7 8 9 10 11 12
1 2 3 4 5 6
48 Bins
Bin
80 Degrees Latitude
114 Degrees Longitude
File No. 1
File No. 2
File No. 3
Unlimited No. of Files per Bin
File
Time Step No. 1
Time Step No. 2
Time Step No. 3
Time Step No. 4
Time Step No. 5
Time Step No. 6
Time Step No. 7
Time Step No. 8
Eight Time Steps
Time Step
(Granule)
Latitude Section No. 1
Latitude Section No. 2
Latitude Section No. 3
Latitude Section No. 29
Latitude Section No. 30
Thirty Sections
Latitude Section
Parameter No. 1
Fifty Values
Thirty Levels
Parameter No. 2
Parameter No. 7
Seven Parameters
Each bin covers an area of 10 deg. lat. by 20 deg. long., with the exception of bins 6, 12, 18, 24, 30, 36, 42, and 48. These eight bins cover a region of 10 deg. lat. by 14 deg. long. Also, these eight bins have only 36 values per level.



A Representation of the NCAR WOCE-CME Model Data (FDGRID)



Flow Diagram for Binning Program (FDBINPRG)

Dynamic Hierarchies and Optimization in Distributed Storage Systems

A. Lester Buck*
Robert A. Coyne, Jr.

IBM Federal Sector Division, Houston

August 28, 1991

Abstract

When designing a distributed storage system, transparent access to storage objects is crucial for software and object portability. On the other hand, large distributed production systems have very significant investments in high-performance storage devices and computing power. Clearly these distributed systems will require intimate knowledge of data location, storage media type and speed, and access delays and costs, as these are critical for efficient performance and scheduling of the system as a whole. We propose techniques to allow controlled penetration of the storage system transparencies by generalizing the simple disk-tape hierarchy paradigm of current storage systems to include multiple co-existing dynamic storage hierarchies. We propose an orthogonal, semantics-free optimization primitive to manipulate the relation between an object and its hierarchy. By managing the hierarchy attribute of a storage object and a system-wide configuration of storage device hierarchies, we can position objects on any desired set of devices without destroying the fundamental transparencies of the system.

*Compact Data, Inc.

1 Introduction

The current IEEE Mass Storage Reference Model paradigm, as evidenced by existing implementations, is a simple disk-tape hierarchy (Fig. 1). At the top of the hierarchy are all disk devices, then all robotic tape (or WORM jukeboxes), and finally human or operator tape vaults. The typical objects (bitfiles) stored in the system are files, although some systems may store database records, hypertext blocks, main memory pages, or other application specific entities. Moving objects down the hierarchy is termed migration, while bringing objects up is termed caching. A migrated object is typically not deleted from the lowest level in the hierarchy when it is cached up, so multiple replicas of an object may exist throughout the hierarchy. Indeed, if partial caching is supported, portions of an object may be resident on different media at the same time. When a replica of an object exists at a low level, unchanged cached copies at higher levels may be discarded whenever space is needed at those levels. No particular distinction is recognized between different disk devices or different tape devices. For example, mainframe magnetic disks connected by

230

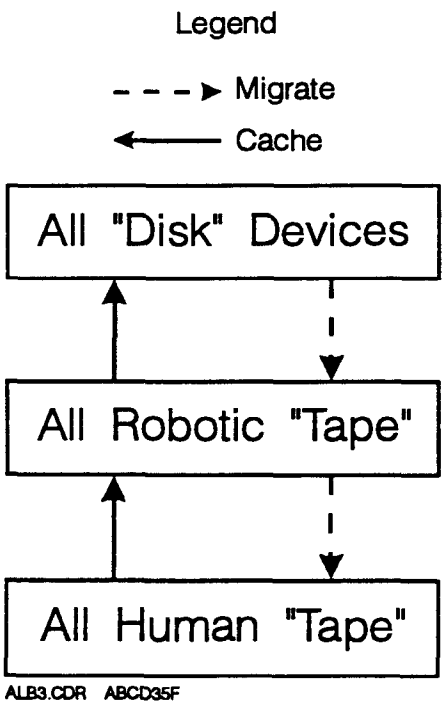


Figure 1: Current hierarchy paradigm.

standard channels (e.g., IBM 3390) have completely different characteristics from a RAID-3 HiPPI attached disk array. Mainframe disks can support smaller random transfers effectively, while the disk array is optimized for large sequential transfers at very high throughputs. The reference model does not directly address the issue of data placement on media of different characteristics.

With the myriad of potential media types becoming available in the next few years, this simple hierarchy will not be adequate for effectively using the I/O devices available in a distributed storage system. Deciding where in this hierarchy the various current and forthcoming media will fit is a serious challenge.

2 Generalized Media Hierarchies

If we are to handle multiple media types in a uniform fashion, we must generalize the simple hierarchy paradigm of Figure 1. The term *hierarchy* connotes an ordering among members. Between levels, members of a hierarchy have a definite ordering, while within a given level, no ordering is evident. The mathematical basis for hierarchies is termed a *partial order*[1]. A partial ordering of a set is a relation between the objects of the set which satisfies the properties of transitivity, antisymmetry, and reflexivity. In a hierarchy the relation may be interpreted to be "is higher or equal." It is well known that every partially ordered set may be mapped onto a directed acyclic graph which is composed of a set of nodes (the hierarchy members) connected by a set of directed arcs. In the case of a storage system with two types of arcs, migrate and cache, we have two overlaid graphs. As a simple example, Figure 2 shows the graph of a two level hierarchy with no distinction between "disk"

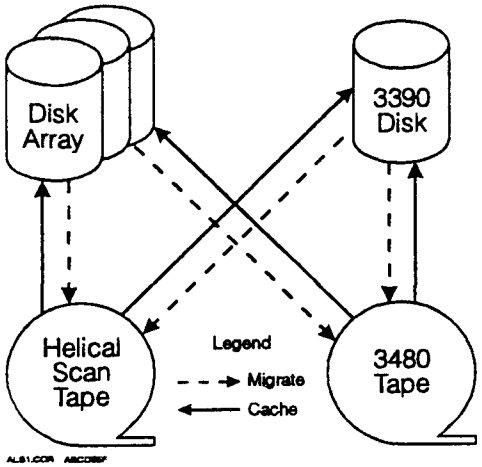


Figure 2: A simple storage hierarchy.

devices and "tape" devices. The disk types are equivalent because caching arcs leave both types of tape devices, and similarly for tape equivalence through migration arcs. Note that although disks are at the same level of the hierarchy, there is no direct arc across disk types. Objects may cross disk types only due to some migration and caching sequence as the storage system performs its functions. The directed graph is naturally acyclic, since no storage system would be configured to migrate (or cache) objects in cycles. The graphs do not necessarily form a rooted graph (a tree), as a storage hierarchy may have several equivalent classes at the highest (or lowest) level. To show a more complicated example, a complex hierarchy is displayed in Figure 3. This example looks ahead to a possible future integrated system using multiple device types. While greatly simplified compared to a real system, the arcs were chosen to demonstrate that the ordering of levels can become non-obvious and that object caching may follow different paths from

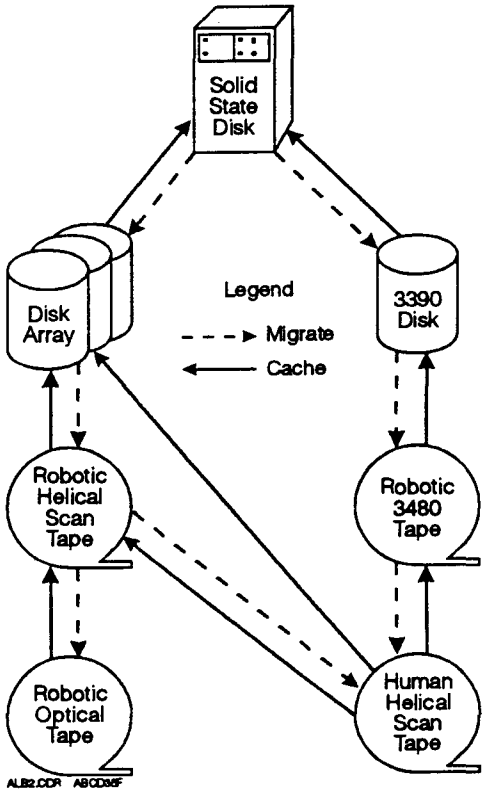


Figure 3: A complex storage hierarchy.

object migration.

The struture of such a graph must be available to the storage system for use by its component functions. There are various data structures used for manipulating directed graphs[2, 3]. Figure 4 is an example configuration syntax that records the arcs leaving each node. This information completely describes one hierarchy of storage devices.

3 Multiple Co-existing Hierarchies

While some storage systems may be described by a single, static hierarchy of existing devices, many systems must handle the problems of supporting objects in different hierarchies and of integrating new media types into an existing storage system. To generalize the idea of a collection of storage devices organized as a single hierarchy and recorded in a configuration file, we consider multiple co-existing hierarchies.

Assume we have a set of configured hierarchies arranged as an array. Let each object in the system be tagged with a hierarchy identifier (HID), which is a conceptual index into this array. In an actual implementation, the HID might be an index into a system administration file which lists the configured hierarchies(Figure 5). Alternatively, this system configuration information could be delivered by a distributed name service. The HID would be a permanent attribute of every object stored in the system. For reasons discussed below, it may be convenient to specify a temporary, effective HID which would override the permanent HID while the current storage operation is active. By centralizing the descriptions of the configured hierarchies, the migration module algorithm becomes uniform throughout the storage system. Todays migration modules are, of necessity, ad hoc, where a robotic tape migration server "knows" that it can migrate to the vault, while the vault migration server has built in knowledge that it is at the bottom of the hierarchy. With our proposal, all migration modules for all devices may share code, algorithms[4], and configuration files. Modules see a uniform view of the storage graph and can make decisions based on device-independent information, such as the caching and migration arcs leaving the node, free space on a given node, time of residence, object size, etc. These decisions do not depend on the object's current position in the hierarchy.

4 Storage Dynamics

We are now able to configure our system with multiple co-existing hierarchies and to tag objects with a hierarchy identifier. Within a given hierarchy of media, the storage system's migration and caching modules will implement policies for free space, time of residence, object size, etc., and decide when all objects should be migrated down. These are normal site policies needed by any storage system that offers automatic migration of objects. We propose adding the following extra condition as an explicit dynamic principle for a storage system:

> Objects may be delivered to or received from an application only when the object is at the *top* of its current hierarchy.

If the object is not at the top of its hierarchy, it is cached to the top and other objects are migrated downward, as necessary. If caching is necessary, the object is not accessible to the client until this caching has positioned at least some portion of the object at the top of its hierarchy. Whether to allow portions of an object to be delivered before it is entirely cached to the top level involves the question of partial

```
#C=caching arc from initial node, M=migration arc from initial node
#
#initial node     terminal nodes
SSD:              M(DA, 3390)                        #top of hierarchy
DA:               C(SSD), M(HST)
3390:             C(SSD), M(3480robot)
HSTrobot:         C(DA), M(HSThuman, OpticalTaperobot)
3480robot:        C(3390), M(HSTrobot, HSThuman)
HSThuman:         C(DA, HSTrobot, 3480robot)     #bottom of hierarchy
OpticalTaperobot: C(HSTrobot)                    #bottom of hierarchy
```

Figure 4: An example hierarchy syntax applied to the complex example. C(M) indicates caching (migration) arcs from the initial node to the terminal nodes.

HID Index	Description	Hierarchy Graph
0 (default)	merged	{DA,3390}/{HST,3480}
1	"fast"	{DA}/{HST}
2	"slow"	{3390}/{3480}
3	generic disk array	DA
4	generic 3480 tape	3480
5	ANSI labeled format	3480ansi
6	real time input	helical scan tape (HST)
7	any 3480 at site X	3480@siteX
8	specific 3480 drive	3480driveA@siteX
9	portable WORM	WORMansi

Figure 5: An example system hierarchy configuration file.

234

caching and is beyond the scope of this paper. Let us consider some of the practical benefits of these dynamic hierarchies.

The simplest case involves changing the hierarchy caching/migration paths in the graph. The administrator edits the configuration as necessary. The migration server is then activated and it schedules necessary movements for all affected objects. No change to stored HIDs is required, which is an important benefit for storage systems that may potentially track trillions of objects.

To add a new device type to the hierarchy, the system administrator edits the central hierarchy configuration file. There are two cases. If an existing hierarchy is added to, the storage system will automatically begin to cache/migrate objects into/out of the new hierarchy, seamlessly using the new device type. Again, there is no change to any stored hierarchy identifiers. If a new hierarchy is formed, then the storage system will utilize the new device type only for new objects that are stored. These new objects will need to use the new HID, but existing objects and their HIDs are unchanged.

When necessary to retire old devices from active service, the administrator edits the configuration file to remove all nodes for the old device type. Those objects which are in a hierarchy consisting of a single device type which is being retired must be handled specially, by supplying a new HID or by reconfiguring the existing hierarchy in the system configuration. After the old device type is eliminated from the configuration file, no new objects will be migrated or cached onto these devices. If it is necessary to completely remove the old devices from the storage system (e.g., copy the objects onto new media), an internal list of objects resident on these devices must be generated by the storage system. The system will then activate the migration server with this list as argument. Since the objects are not on a valid node, they will be moved to some valid residence within their new hierarchy. This operation may proceed at whatever rate desired, possibly scheduled as a background task over a period of months or years. A special case is presented by those objects which are in a hierarchy consisting of a single device type which is being retired. The administrator must supply a new HID for the object or reconfigure the existing (single node) hierarchy definition.

5 Transparency

When designing a distributed storage system, transparent access to storage objects is crucial for software and object portability. Location transparency[5] implies that an application is unable to determine the location of an object from its name. The stronger property of location independence means that the object may be moved at will without a change in its name, which is rare in existing systems. The Sun Network File System (NFS^{TM}), for example, displays location transparency, since once a client machine has mounted a filesystem the physical location of a file is unknown to an application. But NFS does not support location independence, because moving a file (filesystem) between servers is not transparent and requires reopening files. Depending on the adminstrative conventions, this may also entail using a new name for the same file. Only under a fully location-independent naming scheme may names be assigned once, and then be stored and reused at will thereafter. It is clear that software portability in a distributed environment requires location transparency, otherwise the behavior of applications with embedded object names differs depending on the node where they are executed. In the same manner, object portability requires location independence because it is unacceptable to rewrite applications to follow

the new names of objects as they are moved by the storage system. Automatic migration schemes obviously require some form of location independence.

As storage systems expand and distribute, their heterogeneity increases. New media and networking technologies present serious challenges to the management and integration of distributed storage systems. With new media types the requirement for location independence becomes more pressing, as more reasons arise to move existing objects for replication, migration, and/or caching. Large distributed storage systems can have very significant investments in high performance and high capacity hardware. One requirement is a system component which schedules placement and movement of storage objects depending on available device and network bandwidths and latencies. In some systems, the distributed application(s) may directly control and schedule the movement of storage objects. Emerald[6] is one such system which offered primitives to move objects between sites, fix objects at a given site, and inquire as to the location of an object. In others, either the native operating system or the general distributed storage management system built on heterogeneous underlying operating systems may provide object location status and the primitives necessary to affect object location. In a distributed language environment, many of these tasks are subsumed under the runtime support and are completely out of the control of the application programmer. Hermes[7] is such a distributed programming environment which offers a very high level view of a distributed system. The design for Hermes specifically disallowed access by applications to system level details.

6 Optimization Control

There is a conflict between a highly transparent distributed system and the efficient management of resources by the system and/or application. Popek and Walker[8] expressed the problem clearly:

> "One of the functions that conventional, nontransparent systems provide is the ability of applications to take explicit network-related actions for the sake of optimization. [...] Unless some means are provided in a transparent environment to accomplish the same optimization actions, one could fairly expect significant performance difficulties, even in a local network."

They describe the services of the distributed system as an *effect* language, which is the normal set of system calls such as read, write, lock, etc. which drive the computation being performed. They suggest implementing a separate optimization language, "orthogonal" to the effect language. Orthogonal here refers to the fact that the optimization language is *semantics-free*, so that no statement in the optimization language can affect the result of any computation expressed in the effect language. Figure 6 displays some sample optimization primitives.

Very few real systems have supported transparent object movement. In Eden[9], all objects are completely mobile and all operations on objects are location independent. The subsequent Emerald[6] system added the ability to explicitly position objects. Emerald primitives for object mobility included *locate*, *unfix* (from a site), *move*, *fix* and *refix* (atomic *unfix*, *move*, and *fix*). Orthogonal optimization is an active research topic in parallel processing and parallel languages. In a fully functional or declarative language environment,

236

my_loc()	returns the site name at which the call is made
object_loc(object)	returns the site name where *object* is stored
make_near(object, site)	tries to move *object* to a site where access from *site* will be efficient

Figure 6: Example optimization primitives.

the Church-Rosser[10, 11] property guarantees that the optimization annotations are semantics-free. But the performance of functional languages is poor when compared to familiar imperative languages, where the optimization primitives are orthogonal only by inspection and not by formal proof[12, 13].

We propose a new primitive for orthogonal optimization in a storage system:

hierarchy(*object, HID, persistence*)

This primitive sets the hierarchy identifier for *object* to *HID*. *persistence* may be either temporary or permanent. Permanent persistance changes the system's stored hierarchy identifier attribute for the object. Temporary persistance sets the effective HID for the object to the argument HID for the duration of the current operation on the object, but does not affect the permanent object attribute. This primitive is semantics-free in the sense that if it fails, returns an error, or is not supported by a given system, the base computation will proceed unchanged, though possibly more slowly and less efficiently. The failure of the optimization primitive in a realtime system would, of course, be a failure for the entire computation, but for those systems which are insensitive to realtime performance, the success or failure of the *hierarchy*() primitive is transparent to the computation. The standard *open*() system call can be extended to accept a desired permanent HID, which determines the initial placement of a new object in the storage system. If no HID is specified, HID=0 may be configured as the system default (Figure 5).

7 Optimization Examples

In this section, we present a few examples of the use of *hierarchy*(). As a straightforward optimization example, assume we need to position an object permanently on any disk array. We would perform the following steps: (1) find (or create) the hierarchy identifier of a hierarchy with exactly one member, the disk array (HID=3 in Figure 5); (2) permanently set the object hierarchy identifier to be HID, *hierarchy*(object_foo, HID, perm); (3) *open*(object_foo) and, if the object is not already on a disk array, the storage system moves the object to an array; (4) *close*(object_foo) (no read/write necessary) and the object is now permanently resident on some (undetermined) disk array.

For a storage system that has a variety of objects requiring various migration/caching strategies and different latency and throughput, the *hierarchy*() primitive may be used to determine the device type on which objects are received as they enter the storage system. Consider the case of a storage system that accepts data at very high rates from a realtime source, archives this data for long term storage, and then performs various processing functions on the stored objects as background tasks. If the long term archive is high-speed (e.g., striped) helical scan tape (HST), we may force the input to this specialized storage device with: (1) *open*(realtime_object) might set a permanent HID which meets the system needs for migration with background processing; (2) *hierarchy*(realtime_object, HID=6,

temporary) sets a temporary hierarchy which contains only helical scan tape drives, so the media chosen in the *open*() above is switched to helical scan tape for initial residence of the object; (3) *write*(realtime_object) accepts the data into the storage system, directly onto the high speed device; (4) *close*(realtime_object) releases the high speed tape drives and the realtime data is free to be migrated and cached by the storage system as determined by the permanent HID.

To move an object to the "fast" hierarchy, which consists of only high speed disk array and helical scan tape, we would: (1) *open*(fast_object); (2) *hierarchy*(fast_object, HID=1, permanent); (3) *close*(fast_object). The fast object is now in the fast hierarchy. We might be able to skip the open()/close() if it was sufficient that the object actually move only at its next access (read/write). A *migrate*() optimization primitive could be designed to place the object at a lower level on its entry into the new hierarchy.

Depending on the detail of the storage hierarchies expressed in the system configuration file, we could move an object to any drive at a specific site (Figure 5, HID=7) or even to a single, specified drive at a site (HID=8). By using a temporary HID, a copy of an object can be produced on the desired media. This might be of some use in a backup or replication scheme, since this copy of the object is not permanently tracked by the storage system and can be removed from the site. We do not consider this a mechanism to exchange storage objects between sites. In a standard for distributed storage, the system software must interoperate over communication networks with storage modules from various vendors but we consider it inappropriate to standardize the layout of bits on the storage media. We feel this is better left to media interchange standards for the limited cases where physical media transfer between storage systems and sites is necessary.

Nevertheless, the *hierarchy*() primitive may be used to export and import objects in industry standard interchange format. We need only define new hierarchies which contain a specific device type and read/write through an implementation specific software layer which translates the internal format of storage objects to/from the interchange format. For example, to export an object onto an ANSI standard labeled 3480 tape, we would: (1) *open*(outgoing_object); (2) *hierarchy*(outgoing_object, HID=5, temporary); (3) *close*(outgoing_object). The outgoing object remains in the system, but its contents have been externalized. As another example, entire sets of objects might be imported and exported by appropriate operations on a hierarchy which supported the forthcoming ANSI X3B11.1 exchange format for WORM filesystems (HID=9). These software format conversions are typical of the way existing proprietary storage libraries may be absorbed into a standard storage system.

8 Plans

We have described a method of penetrating the location transparencies of a distributed storage system. The level of detail concerning the storage location of an object can range from general (somewhere in the "fast" storage hierarchy) to very explicit (a specific disk drive). Toward the general end of the spectrum, these ideas are similar to System Managed Storage[14, 15] and the proposed DataMesh[16]. This facility may be made available to applications directly or may be inserted as a uniform storage access layer at the system software level. However it is implemented, access to these facilities of the storage system must be limited to authorized processes because misuse can lead to serious performance difficulties with uncoordinated opti-

mization.

We are in the process of implementing the facilities herein described in a commercial storage product[17]. Once the structure of the hierarchy graph is accessible throughout a distributed system, adding extra information is straightforward. For example, the arcs of the storage graph are currently tagged as either cache or migrate. These arcs could easily carry network throughput, latency, and cost information. Additional orthogonal optimization primitives could then search for minimum cost or maximum bandwidth paths and report this information in a uniform manner to system schedulers, applications, etc. Primitives to access queue statistics (lengths, residence times) for object movement between nodes is also a possibility. Again, the failure of these primitives to return useful information would not affect the outcome of the application's computation, only its efficiency. Collecting this information is not difficult. Instead, the problem is standardizing how the appropriate information is shared among heterogeneous clients in a distributed storage system. We think the inclusion of facilities similar to what we have described here is critical to a storage system standard which is to meet the challenge of diverse media types and which will allow various levels of the storage system software to penetrate the location transparencies and optimize system throughput.

References

[1] Donald E. Knuth. *The Art of Computer Programming, Volume 1: Fundamental Algorithms*, pages 258–262. Addison-Wesley, 2nd edition, 1973.

[2] Frances Newbery Paulisch and Walter F. Tichy. EDGE: An extendible graph editor. *Software - Practice and Experience*, 20(S1):63–88, 1990.

[3] H. T. Lau. *Algorithms on Graphs*. Tab Books, Blue Ridge Summit, PA, 1989.

[4] Hartmut Noltemeier. Dynamic partial orders and generalized heaps. *Computing Suppl.*, 7:125–139, 1990.

[5] Eliezer Levy and Abraham Silberschatz. Distributed file systems: Concepts and examples. *ACM Computing Surveys*, 22(4):325ff, December 1990.

[6] Eric Jul, Henry Levy, Norman Hutchinson, and Andrew Black. Fine-grained mobility in the Emerald system. *ACM Transactions on Programming Languages and Systems*, 7:244–269, 1985.

[7] Robert E. Strom, David F. Bacon, Arthur P. Goldberg, Andy Lowry, Daniel M. Yellin, and Shaula Alexander Yemini. *Hermes: A Language for Distributed Computing*. Prentice Hall, Englewood Cliffs, NJ, 1991.

[8] Gerald J. Popek and Bruce J. Walker. *The LOCUS Distributed System Architecture*. MIT Press, Cambridge, MA, 1985.

[9] Guy T. Almes, Andrew P. Black, Edward D. Lazowska, and Jerre D. Noe. The Eden system: A technical review. *IEEE Trans. Softw. Eng.*, SE-11(1), January 1985.

[10] Paul Hudak. Para-functional programming. *Computer*, 19(8):60–71, August 1986.

[11] Paul Hudak. Para-functional languages for parallel and distributed computing. In *Proceedings of Spring COMPCON '87*, pages 331–337. IEEE, February 1987.

[12] Michael H. Coffin. *Parc: An Approach to Architecture-Independent Parallel Programming*. PhD thesis, University of Arizona, Tucson, August 1990. TR 90-28.

[13] Lawrence A. Crowl and Thomas J. LeBlanc. Architectural adaptability in parallel programming via control abstraction. Technical Report TR 359, University of Rochester Computer Science Dept., Rochester, NY, January 1991.

[14] J. P. Gelb. System-managed storage. *IBM Systems Journal*, 28(1):77–103, 1989.

[15] W. B. Harding, C. M. Clark, C. L. Gallo, and H. Tang. Object storage hierarchy management. *IBM Systems Journal*, 29(3):384–397, 1990.

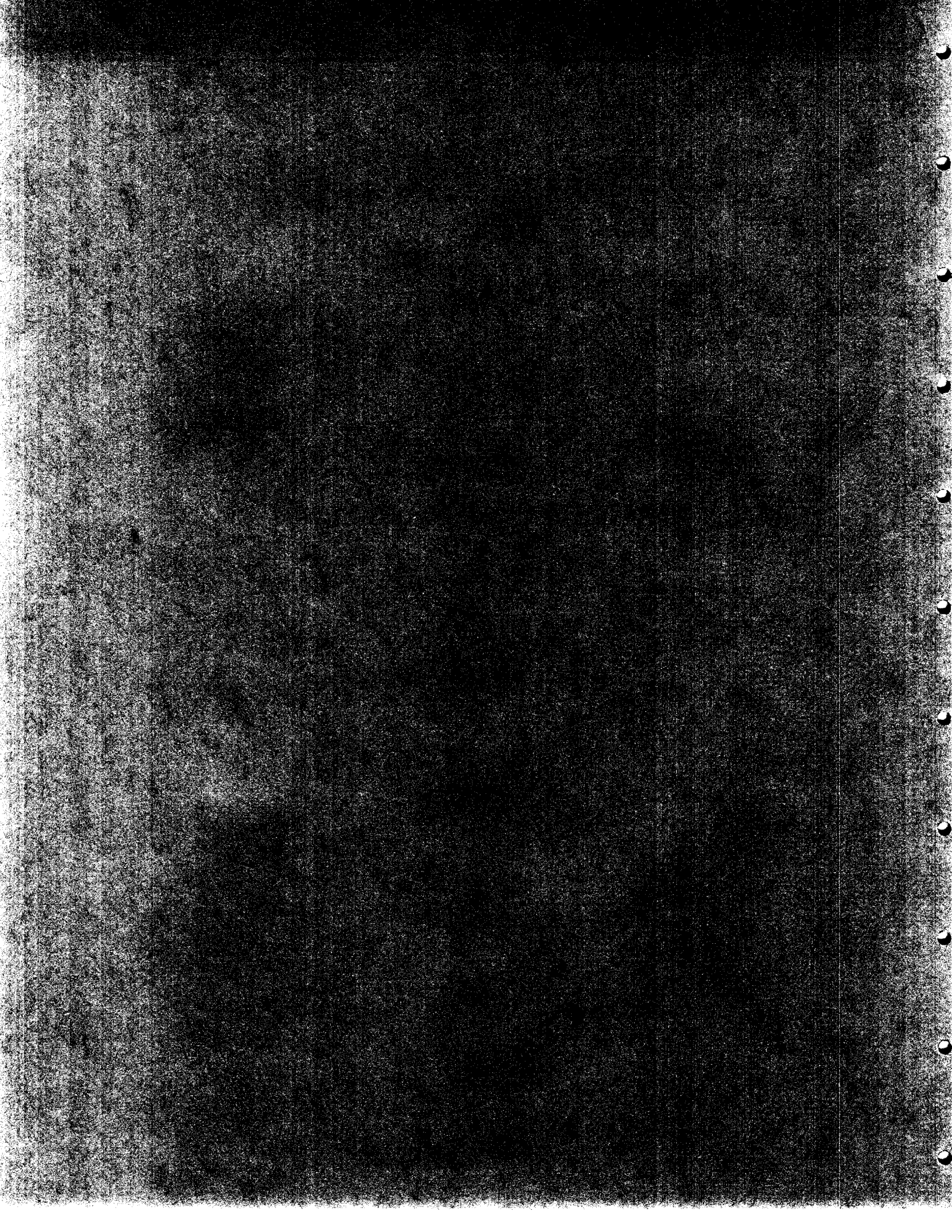
[16] John Wilkes and Raymie Stata. Specifying data availability in multi-device file systems. *ACM Operating Systems Review*, 25(1):56–59, January 1991.

[17] UniTree™ Central File System. DISCOS Division of General Atomics, Inc.

240

"DAFNE Software Requirements"

E. Pace



COMPUTING REQUIREMENTS FOR EXPERIMENTS AT DAΦNE

ELISABETTA PACE
INFN FRASCATI COMPUTER CENTER

DAΦNE

The Frascati Φ factory

- PROJECT APPROVED BY INFN IN JUNE 1990.
- FIRST BEAM WILL BE DELIVERED IN SPRING 1995

e^+e^- storage ring

$E_{BEAM} \doteq 510$ MeV

Bunch dimension.

$.02 \times 2 \times 30$ mm^3

Bunch crossing rate

70 - 350 MHz

Luminosity

$2\ 10^{32} \longrightarrow 10^{33}$ cm^{-2} s^{-1}

It will produce

$> 10^{10}$ ϕ / year

An unique source of

K_S^0 K_L^0 } in a pure quantum state

K^+ K^- }

THE PHYSICS

- CP violation in $K^0_{S,L}$
- CP violation in $K^\pm$
- CPT violation studies
- rare K decays
- chiral perturbation theory checks
- radiative rare ϕ decays
- total cross section measurement

. . - - -

Measurement of the modulus and phase of ε', the parameter that describes the direct CP violation in kaon decay.

$$\frac{BR(K_L \to \pi^+\pi^-)}{BR(K_S \to \pi^+\pi^-)} \Big/ \frac{BR(K_L \to \pi^0\pi^0)}{BR(K_S \to \pi^0\pi^0)} =$$

$$= 1 + 6\, Re\left(\frac{\varepsilon'}{\varepsilon}\right)$$

It is today generally believed that $Re\left(\frac{\varepsilon'}{\varepsilon}\right) \simeq 10^{-3}$

→ experiments searching for direct CP violation should measure $Re\left(\frac{\varepsilon'}{\varepsilon}\right)$ to an absolute accuracy of 10^{-4}.

Being

- $\sigma(e^+e^- \to \phi) \sim 5\,\mu b$
- $BR(\phi \to K_S K_L) = 34\,\%$
- ϕ at rest decays into a $K_S K_L$ pair of ~ 110 MeV/c momentum

→ mean free path for decay of K is

$\langle \ell(K_S) \rangle \simeq .59$ cm

$\langle \ell(K_L) \rangle \simeq \underline{350}$ cm

BIG APPARATUS →
LARGE NUMBER OF CHANNE

248 ≡ PACE

HEPLIB

Branching fraction for $K_L \to \pi^0\pi^0$ is 10^{-3}

The statistical error on $Re\,(\varepsilon'/\varepsilon)$ is

$$\delta\, Re\left(\frac{\varepsilon'}{\varepsilon}\right) = \frac{1}{6}\,\frac{1}{\sqrt{\frac{2}{3}\,N_L^0}}$$

N_L^0 : # of observed $K_L \to \pi^0\pi^0$ decays

$\Rightarrow$ 4.2 10^6 décays ($K_L \to \pi^0\pi^0$) must be observed to achieve the desired statistical accuracy and extracted from $\sim 10^{10}$ accumulated events, which are not CP violating

$\Rightarrow$ 100 days (1 "canonical" year) of running at $\mathcal{L} = 3\ 10^{32}\ cm^{-2}s^{-1}$ with an apparatus of 2 m radius

In order not to spoil $\delta_{STAT}\ Re(\varepsilon'/\varepsilon)$ one needs to keep systematic errors of $5\ 10^{-5}$ in ε'/ε

$\rightarrow$ of $2.3\ 10^{-4}$ in $R^{\pm}/R^{00}$



lot of M.C. simulation both in ~~detector~~ design and in detector evaluation



keep under control

- detection efficiency evaluation
- background subtraction
- geometric acceptance evaluation

The design of a detector for ε'/ε measurement at DAΦNE will substantially rely on the results of a Monte Carlo simulation

For the moment, the approac is traditional: a program based on GEANT is under development (GEANFI).

GEANF1

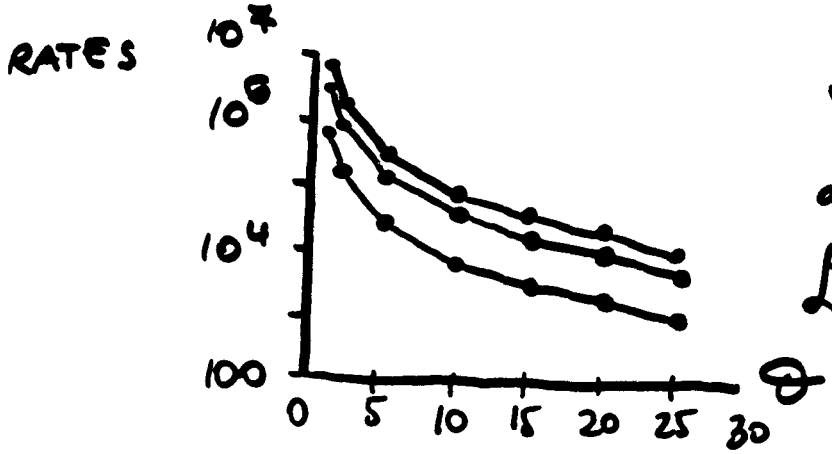
- simulates the generation of the ϕ decay
- simulates tracking of particles through the experimental set up for acceptance studies and simulation of detector response
- graphically represents the set up of the particle trajectories

OPEN PROBLEMS

- code manager is now mandatory : again traditional solution (CMS / expand, CMZ, historian, ...)
- CPU power
- CADD (computer aided detector design) will be really usefull

DATA ACQUISITION

(A. Marini, F. Ronga - Workshop on Physics and Detectors for DAΦNE,



$\mathcal{L} = 10^{33}$
$\mathcal{L} = 5\ 10^{32}$
$\mathcal{L} = 10^{32}$

Rates vs. polar opening of the apparatus integrating the differential cross sections of

$e^+e^- \longrightarrow \phi$

and QED processes

$e^+e^- \rightarrow e^+e^-$

$e^+e^- \rightarrow \mu^+\mu^-$

$e^+e^- \longrightarrow \gamma\gamma$

The total rate from physics
events is largely dominat
by Bhabha scattering in
the angular $25° < \theta < 155°$
for which the cross sections
of ϕ production and Bhaba
scattering are comparable.

Rates

The prescaling of Bhabha events should be performed in the first stages of the data acquisition by using collinearity triggers in the central chamber and/or in the calorimeter

RATES (KHz)

	$\mathcal{L} = 10^{32}$	$5\ 10^{32}$	10^{33}
physics events	> 10	> 50	> 100
logged events (Bhabha scaled)	1	5	10

event length and data throughput

It is possible to guess

2 - 3 kB/event central chamber

1.5 - 15 kB/event calorimeter

→ average event length 5 - 20 kB

→ final data throughput (MB/s) of the data acquisition system

$\mathcal{L}$ =	10^{32}	$5\,10^{32}$	10^{33}
	5 - 20	25 - 100	50 - 200

→ amount of data (in TB) to be collected in 1 year (assuming that the machine will deliver the peak luminosity 50 % of the time)

$\mathcal{L} =$	10^{32}	$5\ 10^{32}$	10^{33}
	75 - 300	375 - 1500	750 - 3000

Obviously data reduction techniques will be used.

By scaling down the CPU time needed to process an event in Aleph the figure obtained is 0.15 s/event on 7.5 MIPS CPU.
The number of units (100 MIF each) needed to process the events generated at DAΦNE

$\mathcal{L}$	10^{32}	5.10^{32}	10^{33}
	12	60	120

OPEN PROBLEMS

- CPU power
  (high speed networks,
  high speed buses
  switched networks)

- mass storage capacity
  and I/O bandwidth.
  It is an "hard" problem
  (1-2 orders of magnitude increment to state of the art experiments)

These problems can be solved by manufacturers.

The "real" open problem is

DATA MANAGEMENT

It could be solved:

- looking for home-made solutions
- in collaboration with manufacturers
- using the IEEE mass storage standard system

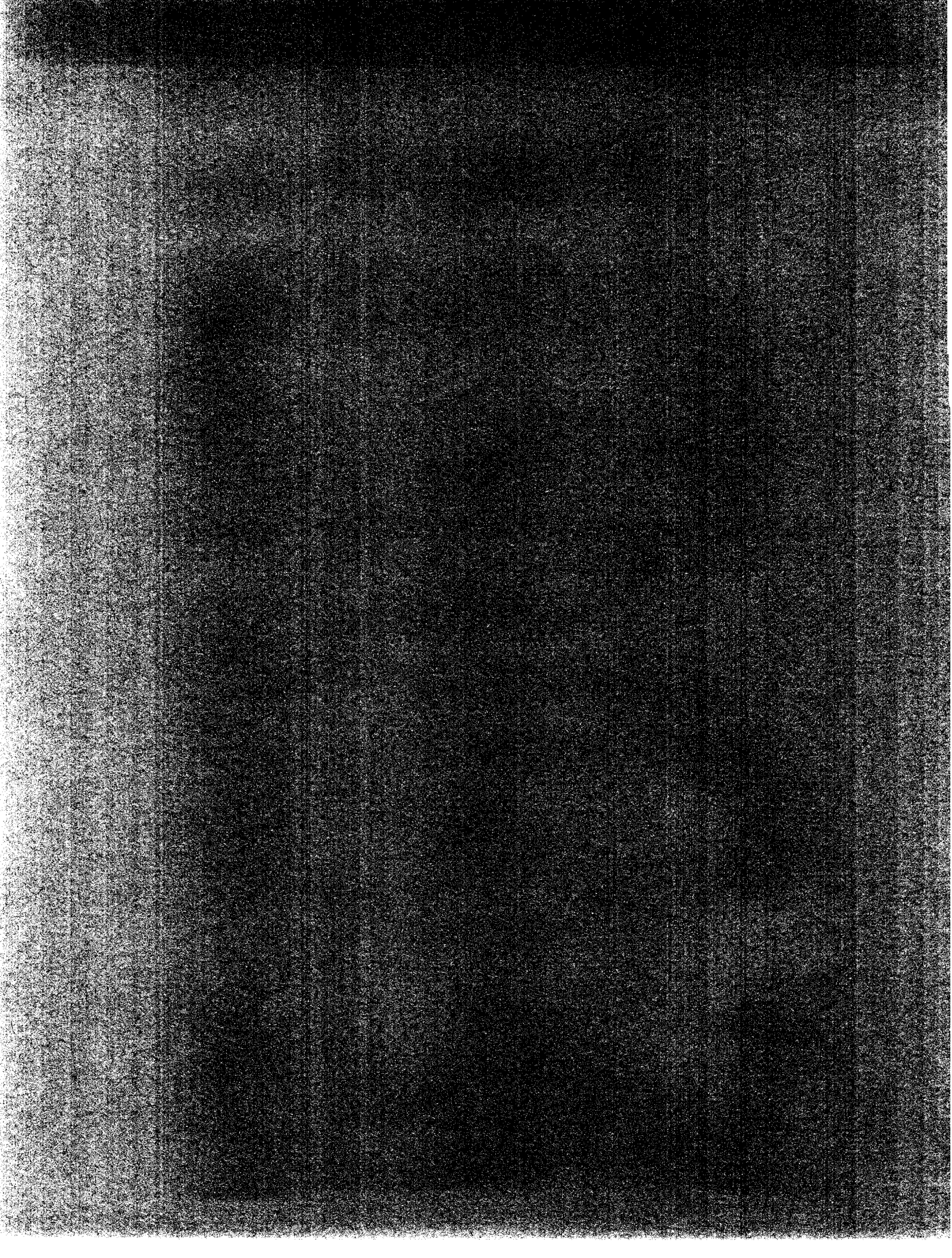
The experiments at DAΦNE will ~~contribute~~ constitute an important step toward the future generation experiments (LHC, SSC)

⇓

experiments at DAΦNE can test new architectural solutions

"Discussion of the Need for Active Promotion of non-HEP Applications of Advances in HEP Computing"

V. Chaloupka



Discussion of the Need for Active Promotion of non–HEP Applications of Advances in HEP Computing

International Users Meeting on High Energy Physics Computing
Dallas, Texas, Sept. 1991

Vladimir Chaloupka
Physics Department
University of Washington, Seattle

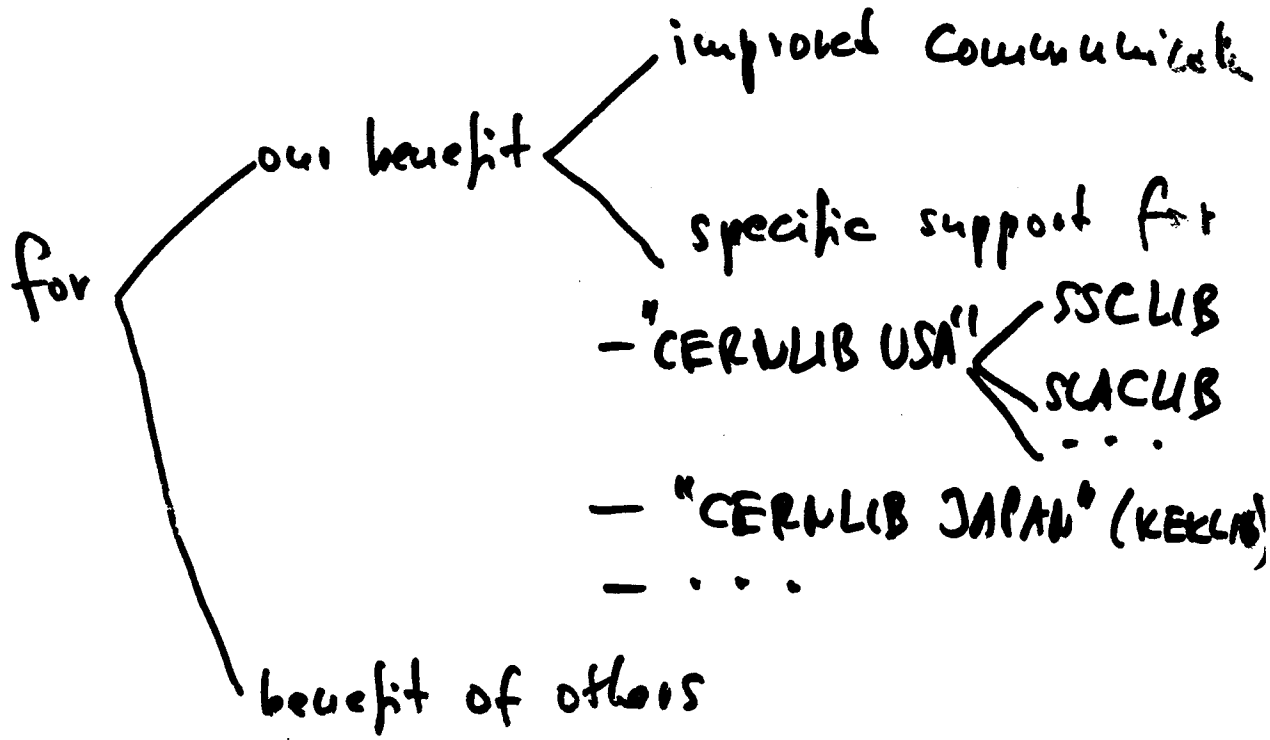
- why HEP should have as many spinoffs as possible:
  - because it is part of our propaganda (especially SSC)
  - also, it is good
- why many advances in HEP computing should have wider applications:
  - extreme demands (speed, data rates, info processing) drive progress
  - inherent parallelism
  - inherent modularity
- examples:
  - statistical packages
  - theory–motivated parallel architectures (lattice)
  - experiment–motivated parallel architectures (events)
- the benefits of active promotion:
  - enhancement of HEP byproducts and spinoffs, with favorable effort/outcome ratio
  - increased HEP visibility
  - reverse flow: neural networks, information processing (hypertext etc.)
  - improved contacts with industry
- proposals:
  - appointment of responsibility within the HEPLIB world users group for non–HEP application promotion
  - contact-person at each major HEP lab (plus SCRI etc.)
  - issue a (quarterly?) Newsletter on General Applications of Advances in High Energy Physics Computing.

V. Chaloupka

"what is the problem ?"

there is no problem

just recognition that we are a Special Interest Group, and therefore, we should organize



"Initial Experience Porting CDF Software to RS6000"

R. Hollebeek

Porting CDF Code to UNIX and the IBM RS6000

R. Hollebeek, University of Pennsylvania, HEPLIB'91

Motivation for UNIX

Increased General interest in UNIX as a solution to Heterogeneous computing.

Increased usage within the Universities of UNIX especially within engineering support groups

Front-end trigger farm upgrade (from ACP to MIPS)

Desire for increased file connectivity

UNIX advantages

Availability on large number of platforms each has a particular reason for being attractive

- DEC - due to historical relationship with HEP
- SUN - one of the first widely available UNIX platforms, and common in engineering application
- IBM - connectivity to mainframes
- HP - code needs to be benchmarked to tell how many mips it really is
- Silicon Graphics - in CDF, front-end data acquisition machines, some of which are S.G.

Extensive networking capabilities including

rlogin, rexec, -display=...

Networking - NFS

Advantages (cont.)

Rapid increase in available CPU power

Graphical User Interface (GUI) support in X

Reasonable help facilities - man, info, CD rom documentation etc.

Graphics tools for PAW etc.

UNIX disadvantages

Small processor I/O bandwidth

Large Increase in system administration load

System administration difficulties

Rapid evolution of operating system versions

Retraining - yet another new operating system for the user to learn

"Unusual" command names

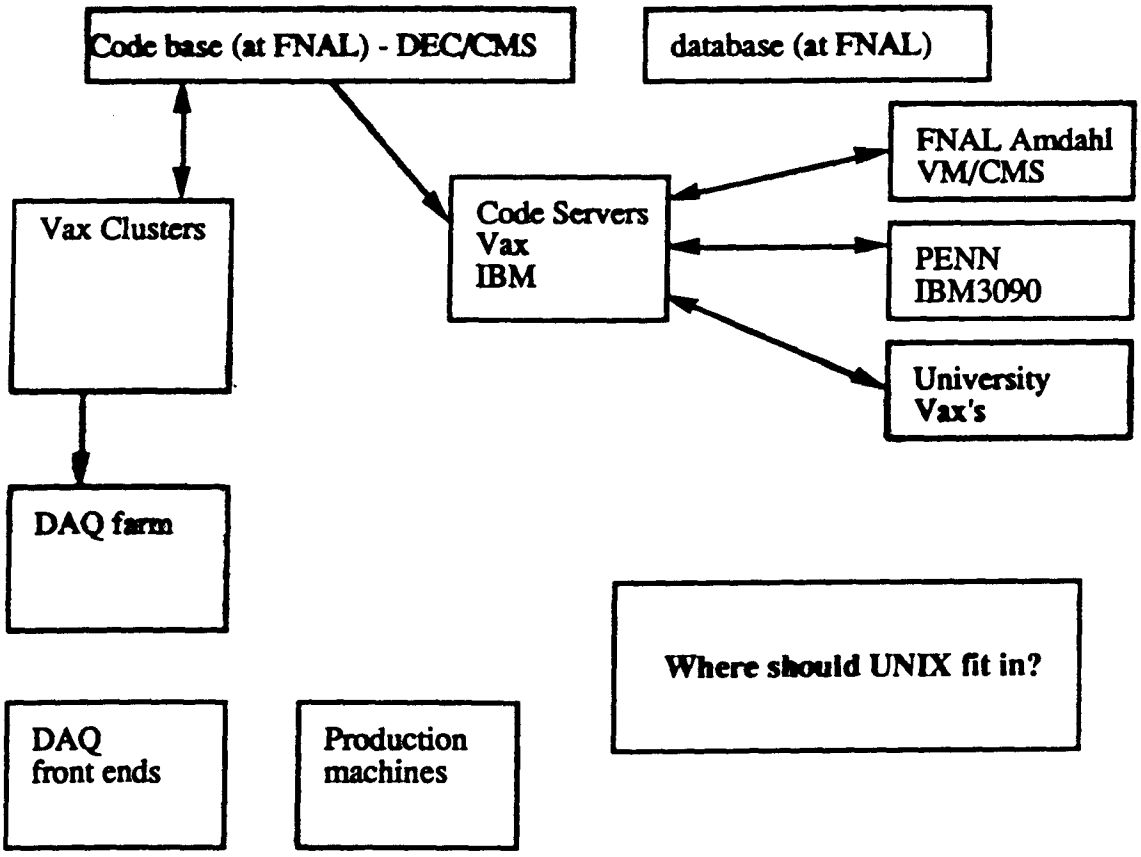
Large amount of network traffic required by cross-NFS mounts

Overall balance

My own opinion is that the advantages vastly outweigh the disadvantages.

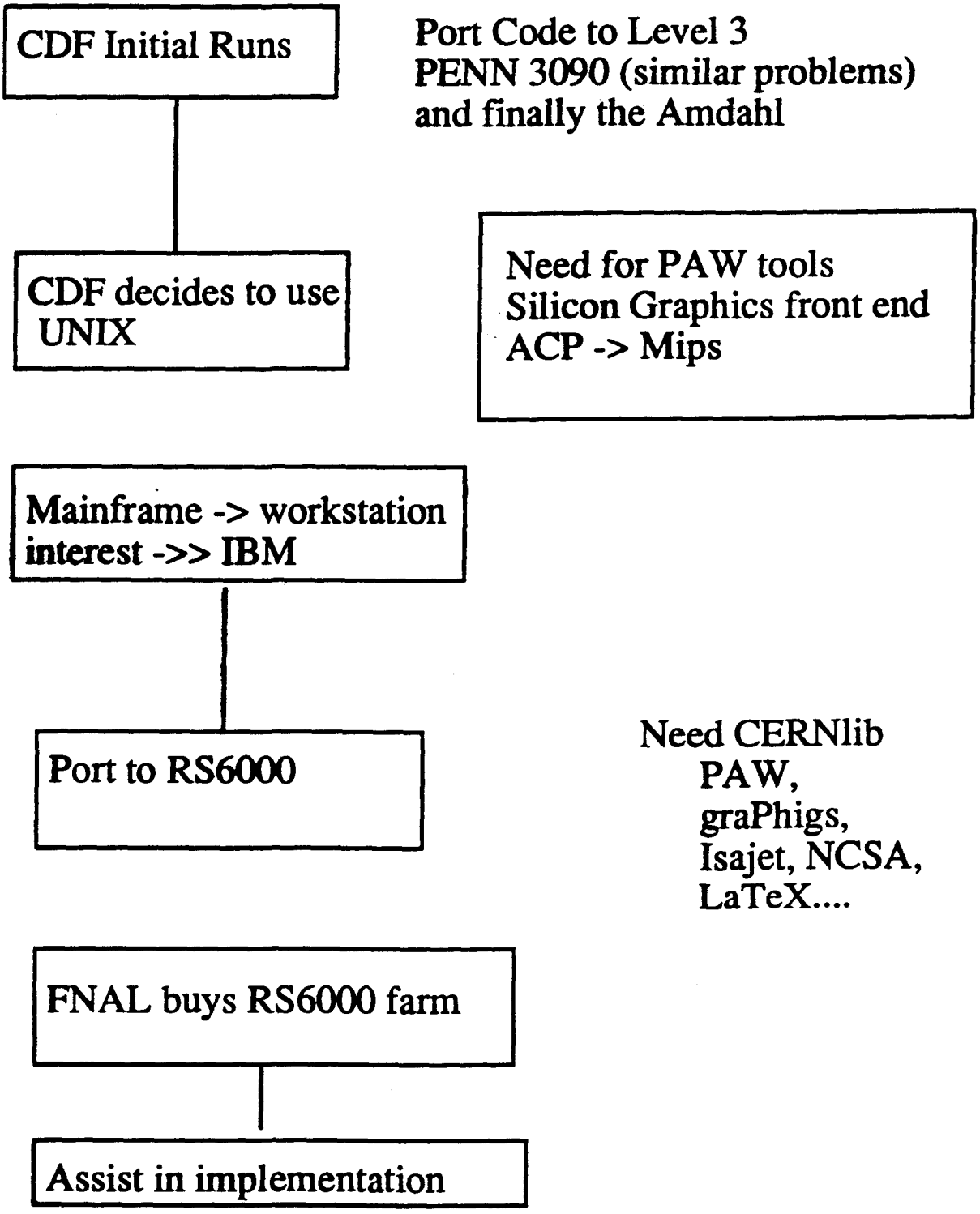
The learning curve is probably 6 months, and the major expense will be hardware replacement.

CDF Environment

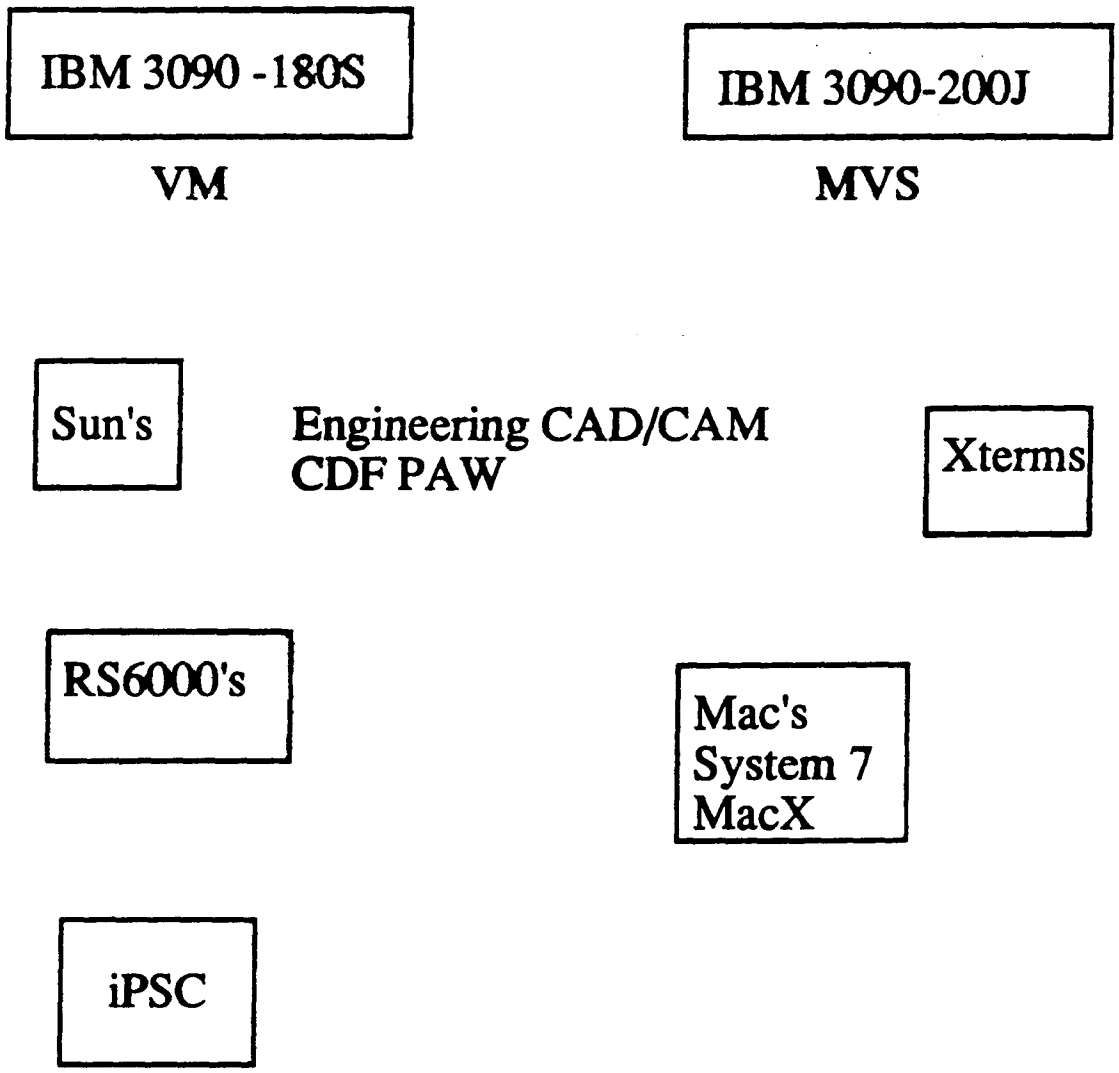


The widely spread CDF community provides a micro example of the problems of code distribution for a "development" environment, which is quite different from that of "package" distribution.

Motivation for UNIX



PENN Environment



CDF software primarily 3090 base with code maintained by servers to FNAL Vax cluster. Complete mapping of VMS file structure to VM.

Initial Experience

Machine Setup
- simplified by SMIT

Problem report and database in IBM and user inquiry database including keyword search mechanism

get CDF code snapshot

Isolate the problems
- compile the code
- $$If ....
- $$Else...

List of Problems
- code incompatibility
- syntax
- OP sys glitch

Problem reporting
- 6x...
- 7x...

Have a simple user interface

Set up (and edit) the user feedback

Provide for response mechanism?

Need a means of keeping machine dependent code obvious

Outstanding problems should be documented and tracked

Strategy

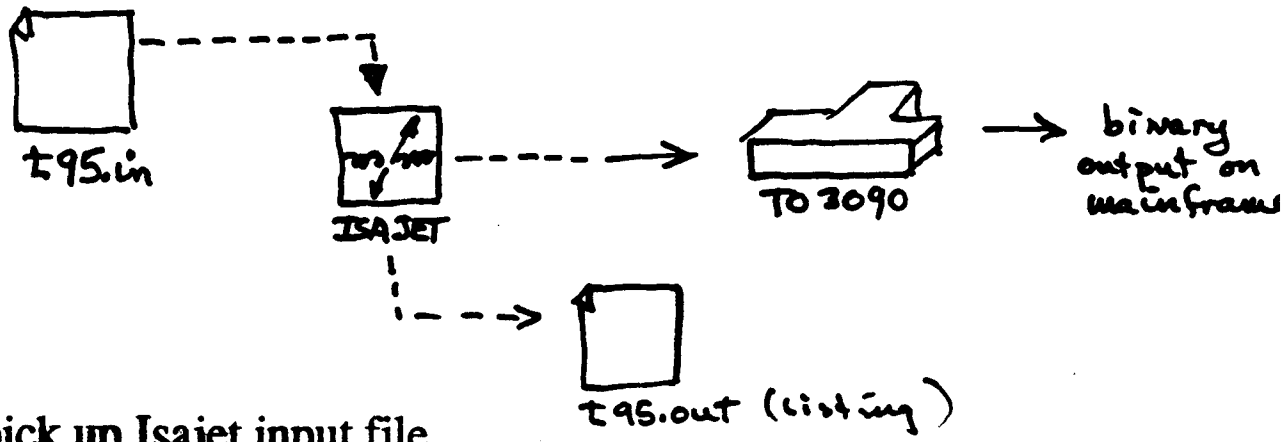
Isajet 6.36

file open and close interface needed to be changed

optimizer bugs

Isaplt -> PAW -> display

Desktop tools



pick up Isajet input file

drop icon on Isajet icon

output (listing) appears in parent (input) folder

output (binary) appears on mainframe disk (via FTP)

Similar icon approach to analysis

icons are also UNIX shells so can be constructed easily

icons can be c programs

Porting CDF Code to UNIX and the IBM RS6000

Motivation for UNIX

Increased General interest in UNIX as a solution to Heterogeneous computing.

Increased usage within the Universities of UNIX especially within engineering support groups

Front-end trigger farm upgrade (from ACP to MIPS)

Desire for increased file connectivity

UNIX advantages

Availability on large number of platforms each has a particular reason for being attractive

DEC - due to historical relationship with HEP

SUN - one of the first widely available UNIX platforms, and common in engineering application

IBM - connectivity to mainframes

HP - code needs to be benchmarked to tell how many mips it really is

Silicon Graphics - in CDF, front-end data acquisition machines, some of which are S.G.

Extensive networking capabilities including

rlogin, rexec, -display=...

Networking - NFS

Check FTP binary transfer of CDF data files

YBOS allows machine independent transfer

actually a bug was discovered at this point. Data is written in VAX format and was being written incorrectly (but was inverted with a similar bug when read.)

Machine independent data format is important for collaborations with heterogeneous computing base.

ana -> isajet -> ybos output

CDF like most collaborations has had to do the work of making standard MC output for interface to the simulation. All generators have a translation routine to CDF format.

Expand - to keep machine dependent code\

misc. bugs and need to specify more platforms

shell problems - CDF uses c shell

need all CDF Utilities

YBOS

UIPACK

MENU

A_C

Read raw CDF DST (disk)

READ 8mm Tape

Fortran sequential I/O problems

tape label processing

Some of the problems in the utilities included

bytes vs words in LRECL

filename problems

getenv

READONLY problems

/dev/rmt0 variable length problems

256 character limitation of library search path

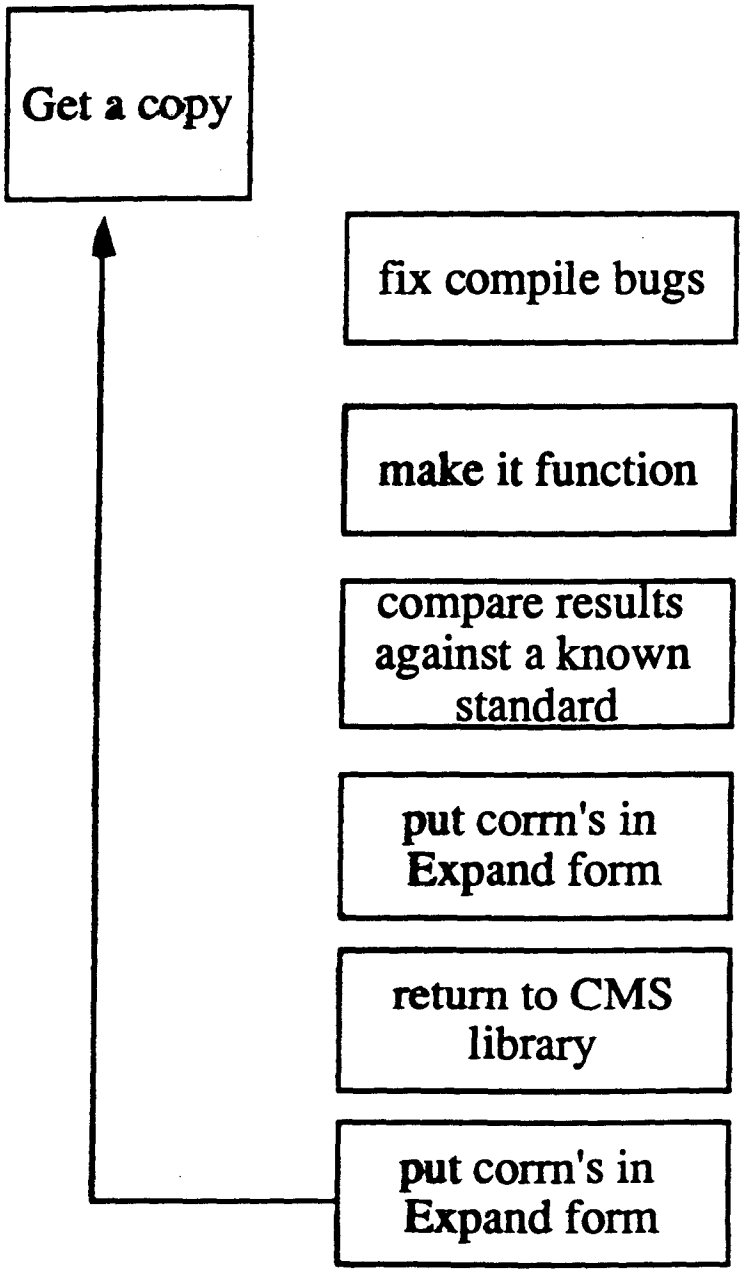
difficulty in setting up shared libraries

Bring up offline display

DI3000 -> graPHIGs emulator developed in joint study with IBM

Setup UPD to distribute to IBM Kingston

Porting Map


Get a copy
fix compile bugs
make it function
compare results against a known standard
put corrn's in Expand form
return to CMS library
put corrn's in Expand form

Packages (partially) ported

CERN products

UPS, UPD - FNAL code transfer

Cdfdb - CDF data base code

Expand - machine dependency preprocessor

C$INC- include file library

Error_logger

Isajet

Bank_dump

C$Part, partprop - particle database

jet, mbs, vtx, sac, trs, gmu, mon, sim, ele, cas, cal

c$gen - MC generator modifications

generator$source

C$dsp - event display

C$trk - tracking library

cdfgeom

menu_source

physics_utilities

random_number

.....

UPS/UPD, xupd

- very nice package for UNIX distribution

CDFCP

- first attempt to solve development server problem
- construct makefiles from CMS/MMS

AFS

- we need to be able to share files between sites transparently

NQS

- we need to be able to use multinode architectures

Motif and Desktop Interfaces

- desktop components are command files or c programs
- user areas are organized with makefiles

Replacement for the ana exec

- need a simple interface for running the program
- must run interactively and from command file

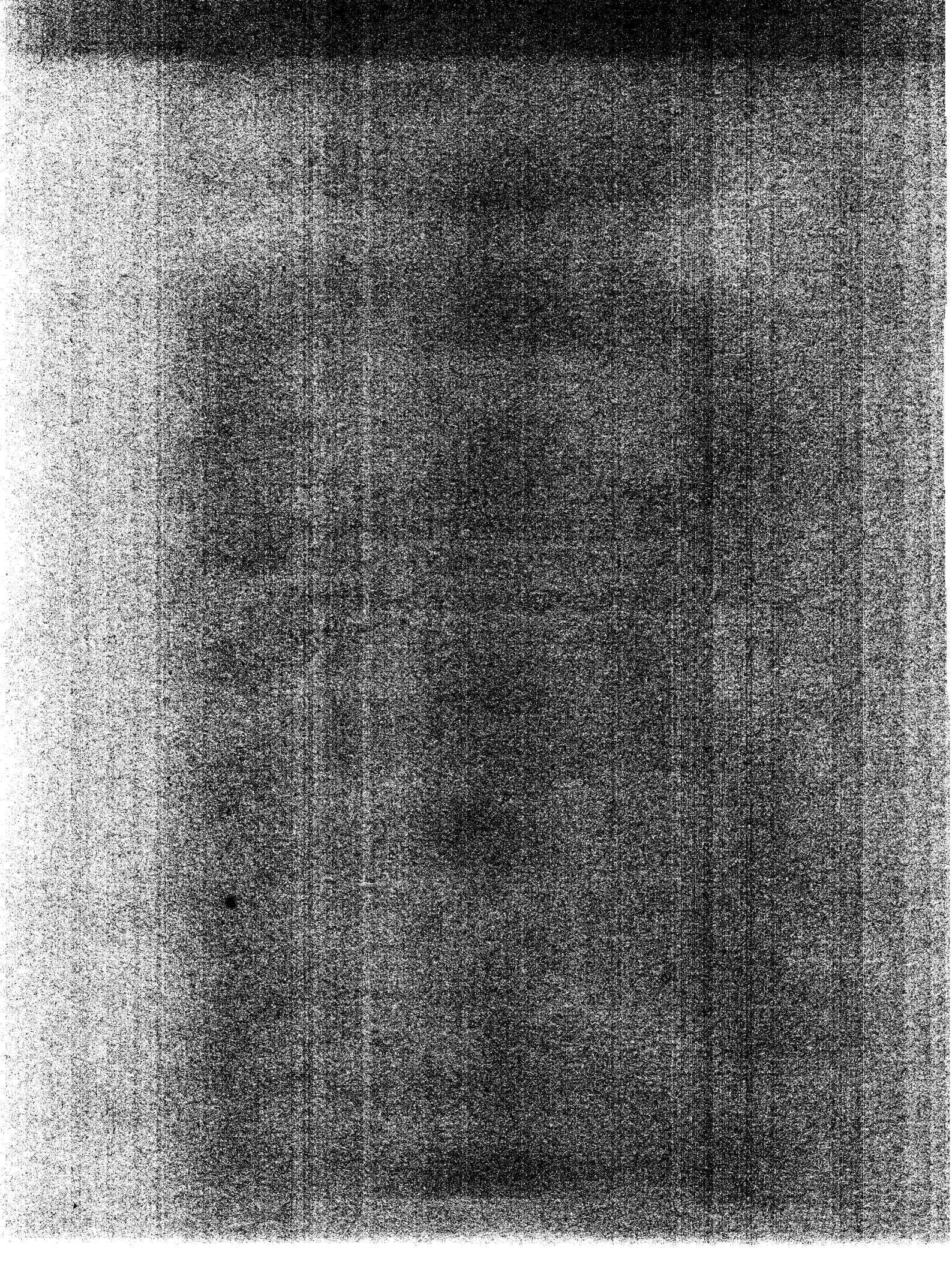
Summary

and some comments for HEPLIB

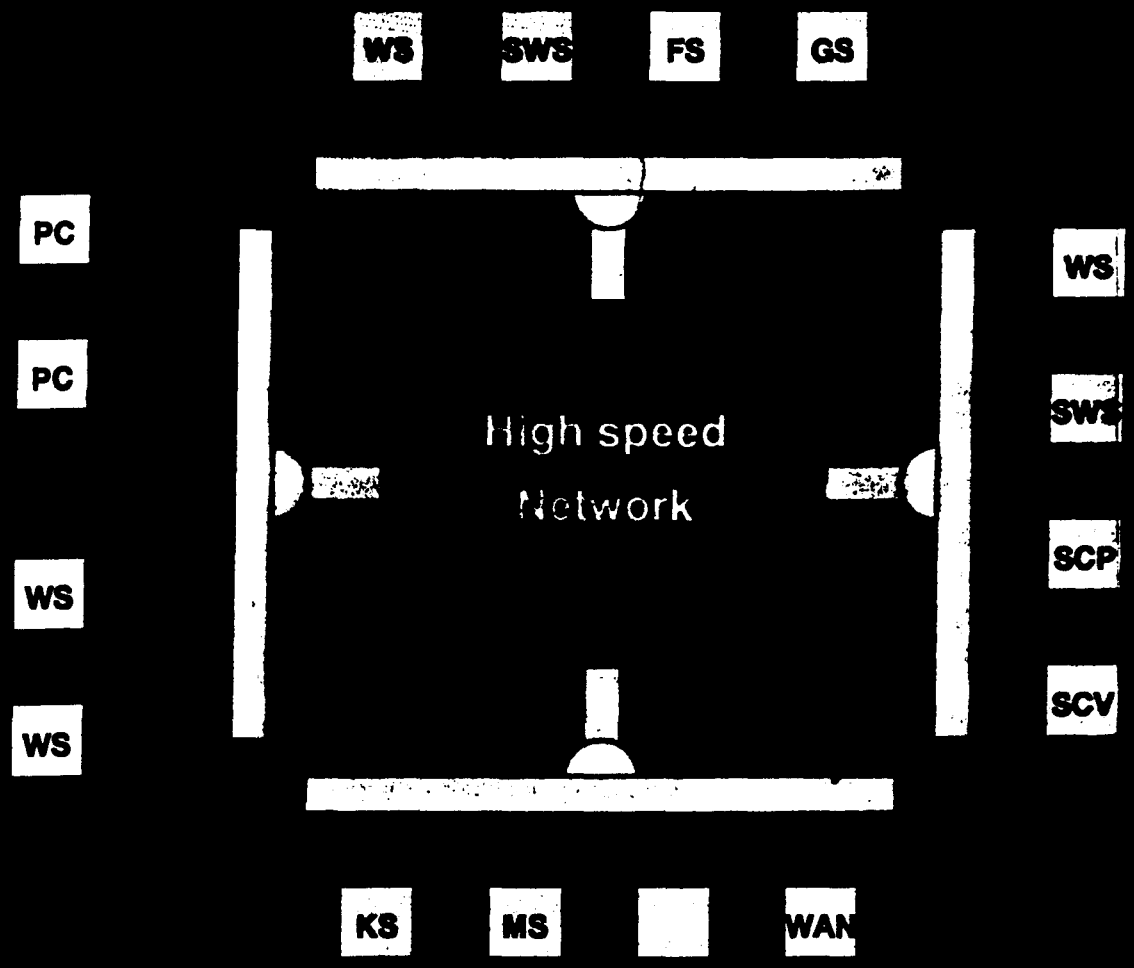
- the problem of the developer is more complex than just receiving a new copy of the latest library
  - need development version as well as package distribution tools
- the following three functions are not necessarily best handled with the same tool
  - code management
  - machine dependency storage
  - code distribution
- many of the current UNIX tools are better than some of our old habits
  - makefiles are a good example (MMS)
  - c structures is another though more controversial (Zebra)
  - motif and other graphic interfaces
- availability of code is the key to growth
  - make it as easy to get/update as possible

"Distributed Computing at CERN"
"The SHIFT/HOPE Expansion"
"Experience with ISIS and Distributed GEANT"
"The Simulation Farm Project"
"The Visualization Farm Proposal"

R. Brun



Towards a COMPUNET



Work Station

Super Work Station

File Server

Graphics Server

Kerberos Server

Mail server

Super Computer (eg. CM5)

Super Computer (eg. Y-MP)

Simulation Farm

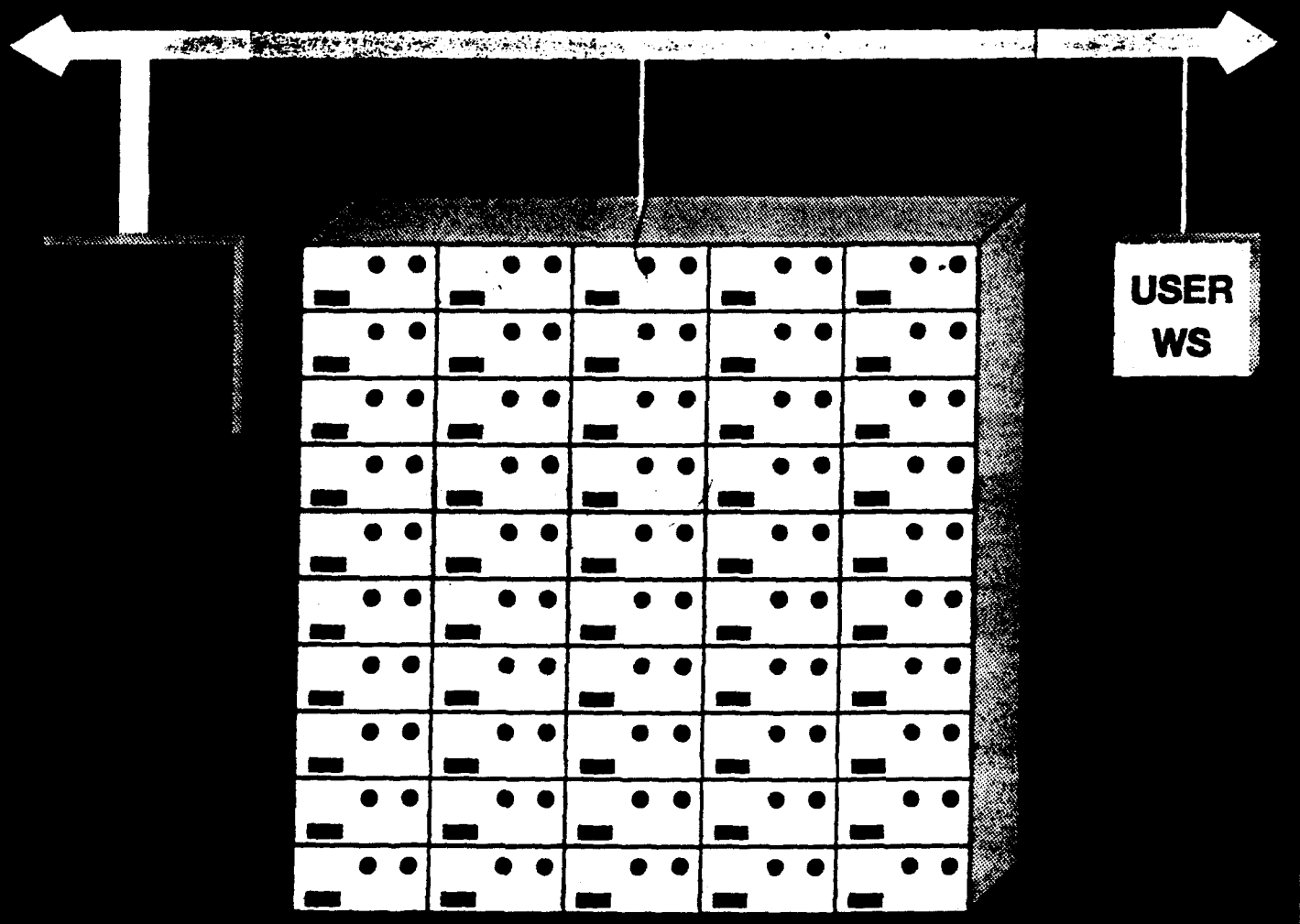
LEP

- *Time to simulate one event: 30s < T < 120s with full shower development*
- *Requirement for 1,000,000 events per month to control systematics at one per cent level*
- *Want to run long jobs all the time*
- *Need robustness and file staging*
- *Manpower available from experiments*

LHC

- *Time to simulate one event: 500s < T < 5000s with parametrization or full showers*
- *Interactive development : a few hundred events*
- *Real-Time problem*
- *From time to time long batch jobs*
- *Interactive histograming (limited I/O)*
- *Manpower ? (ECP, PPE)*

Simulation Farm



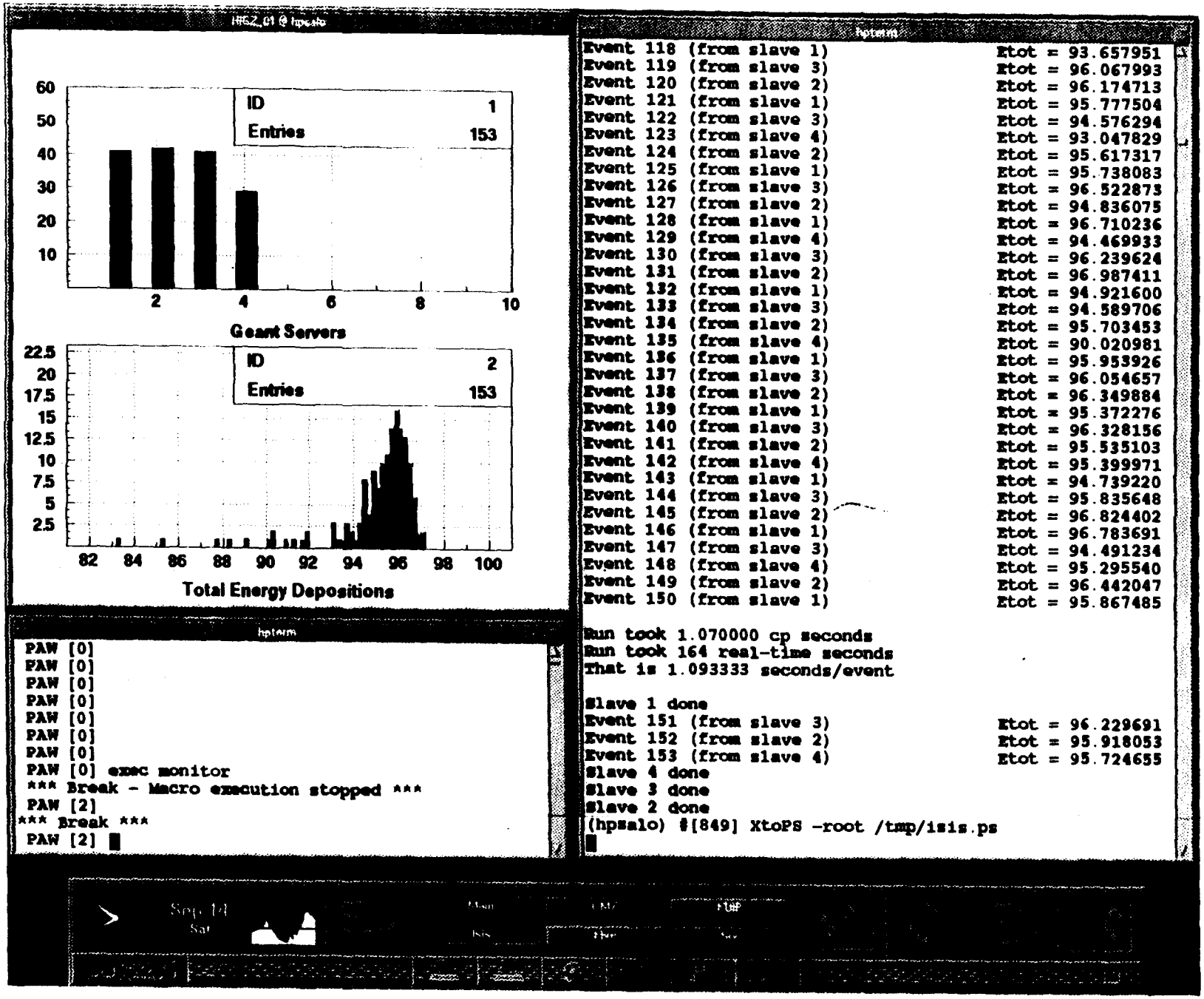
A possible configuration : 50 x HP720 (16 Mb + 400Mb)

= 14 CRAY XMP/48 or 8 IBM 3090/600 J

= 4 x Computer centre

COST = [redacted]

FARM R.Brun POW91


ID 1
Entries 153
Geant Servers
ID 2
Entries 153
Total Energy Depositions
hplerm
PAW [0]
PAW [0]
PAW [0]
PAW [0]
PAW [0]
PAW [0]
PAW [0]
PAW [0] exec monitor
*** Break - Macro execution stopped ***
PAW [2]
*** Break ***
PAW [2]
hplerm
Event 118 (from slave 1) Etot = 93.657951
Event 119 (from slave 3) Etot = 96.067993
Event 120 (from slave 2) Etot = 96.174713
Event 121 (from slave 1) Etot = 95.777504
Event 122 (from slave 3) Etot = 94.576294
Event 123 (from slave 4) Etot = 93.047829
Event 124 (from slave 2) Etot = 95.617317
Event 125 (from slave 1) Etot = 95.738083
Event 126 (from slave 3) Etot = 96.522873
Event 127 (from slave 2) Etot = 94.836075
Event 128 (from slave 1) Etot = 96.710236
Event 129 (from slave 4) Etot = 94.469933
Event 130 (from slave 3) Etot = 96.239624
Event 131 (from slave 2) Etot = 96.987411
Event 132 (from slave 1) Etot = 94.921600
Event 133 (from slave 3) Etot = 94.589706
Event 134 (from slave 2) Etot = 95.703453
Event 135 (from slave 4) Etot = 90.020981
Event 136 (from slave 1) Etot = 95.953926
Event 137 (from slave 3) Etot = 96.054657
Event 138 (from slave 2) Etot = 96.349884
Event 139 (from slave 1) Etot = 95.372276
Event 140 (from slave 3) Etot = 96.328156
Event 141 (from slave 2) Etot = 95.535103
Event 142 (from slave 4) Etot = 95.399971
Event 143 (from slave 1) Etot = 94.739220
Event 144 (from slave 3) Etot = 95.835648
Event 145 (from slave 2) Etot = 96.824402
Event 146 (from slave 1) Etot = 96.783691
Event 147 (from slave 3) Etot = 94.491234
Event 148 (from slave 4) Etot = 95.295540
Event 149 (from slave 2) Etot = 96.442047
Event 150 (from slave 1) Etot = 95.867485
Run took 1.070000 cp seconds
Run took 164 real-time seconds
That is 1.093333 seconds/event
Slave 1 done
Event 151 (from slave 3) Etot = 96.229691
Event 152 (from slave 2) Etot = 95.918053
Event 153 (from slave 4) Etot = 95.724655
Slave 4 done
Slave 3 done
Slave 2 done
(hpsalo) #[849] XtoPS -root /tmp/isis.ps

PAW

- Interactive data analysis and presentation
- Multi dimensional statistical analysis. Minimisation

Today

- *1 process on 1 processor (mainframe/workstation)*
- *Data sets < 10 Mbytes*
- *UIMS + conventional procedural languages (Fortran, C)*

Tomorrow

- *1 process on a workstation + many concurrent processes on CPU and file servers*
- *Graphical programing*
- *Real-time 3D/4D dynamic graphics over LANs and WANs*
- *Very large data base (> 1 Terabyte) (Data sets > 1 Gbyte)*
- *Requirements : LANs > 1 Gbit/s*
  WANs > 40 Mbit/s
  CPU > 1 Teraflop per active user

Visualization will become the main consumer of computer resources

Visualization Farm

Client running PAW on WS

PAW > Histo/File 1 //shift/user/brun/demo.hist

PAW > Ntuple/load 10

*PAW > Ntuple/plot 10.var x**2+y**2 < var*

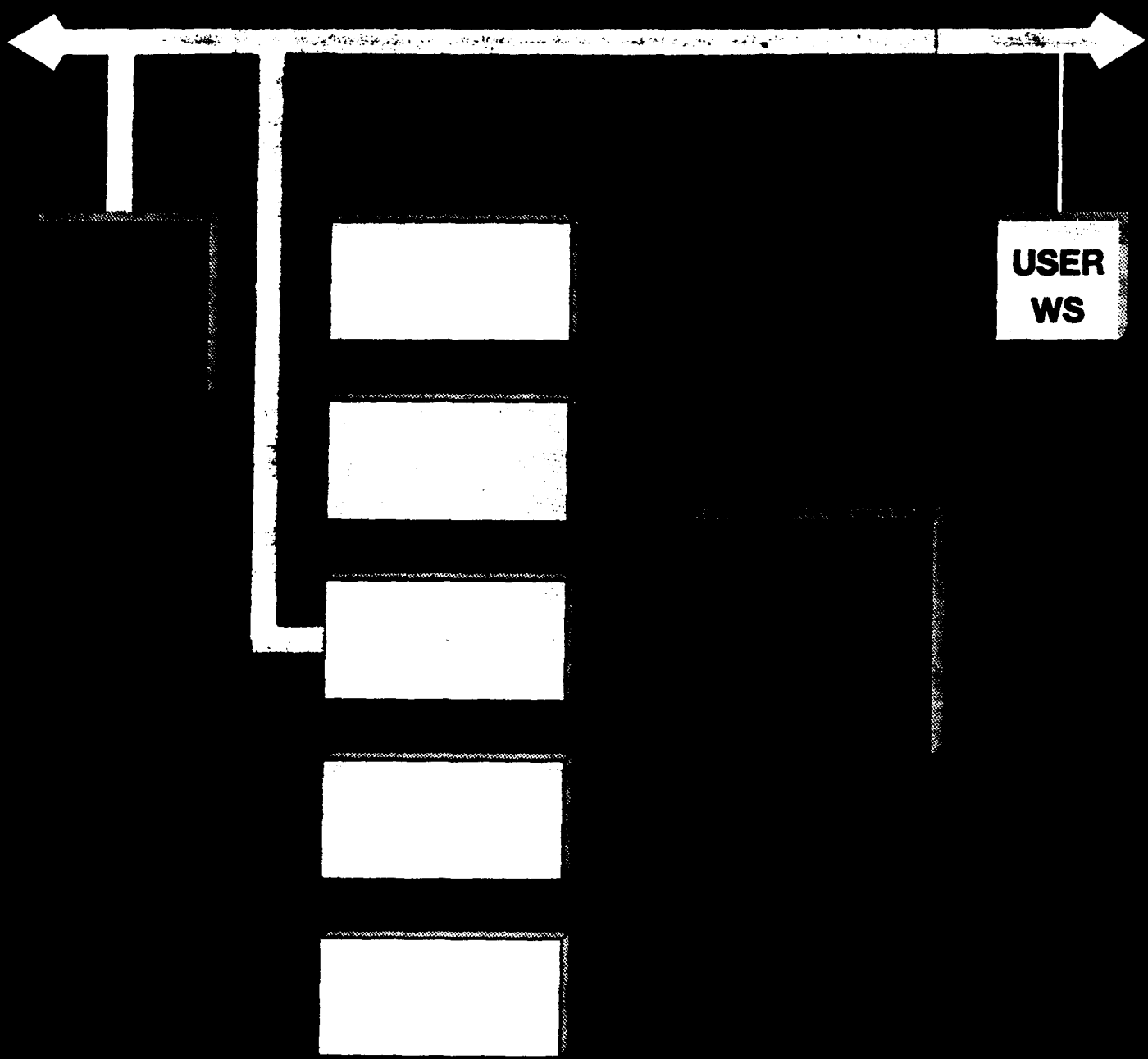
PAW Servers : Farm

- *Ntuple ID=10 is read from file demo.hist*
- *Events are equally distributed to servers*
- *The compilation of the selection algorithm is distributed to servers*
- *Histogram(s) are filled on individual servers*
- *Histograms are sent to client, added and displayed*
- *Multiple clients per server*

*'One server (RS6000/550 or HP720) can process'**

1 Mbyte/s in average

Visualization Farm



A possible configuration : 5 x RS6000/550 (128 Mb)

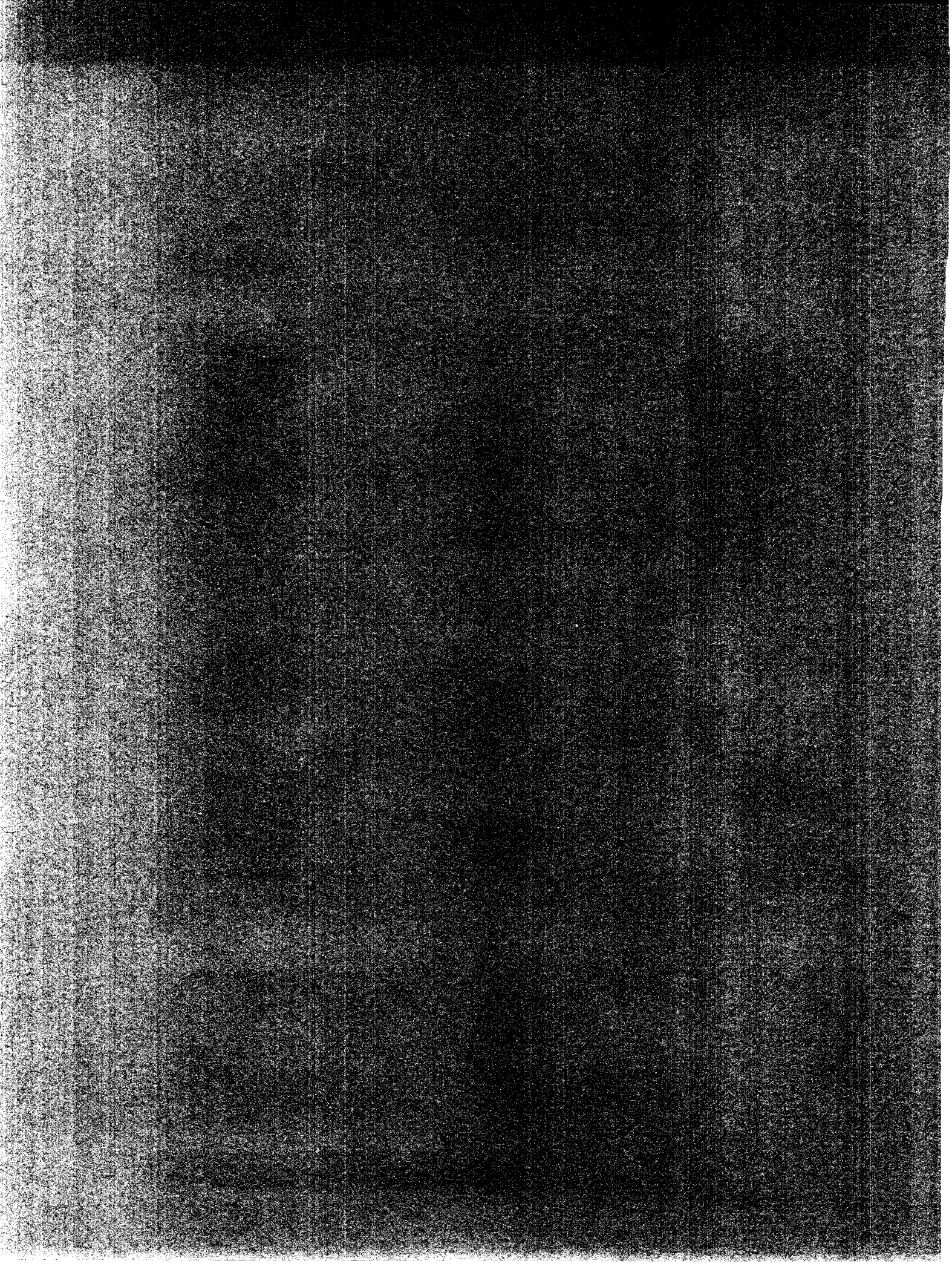
\+ FCS + Disk Array 50 Gbyte

VFARM

R Brun POW91

"The SDC SHELL Project"

S. Kunori



THE SDC SHELL project

20-Sep-1991

S.Kunori
U. of Maryland

Goal of the project:

- **Provide common environment to simulation work for detector design in SDC.**

simulation
-> reconstruction
-> analysis

The project started about a year ago. and codes have been developed at many places in US and Japan.

SDCSIM

Requirements:

- **single framework to support application programs**
- **machine independent (UNIX, VAX, ...)**
- **portability of program system**
- **possibility of future upgrade**

Solution:

- **simple system**
- **avoid commitment to particular product (software and hardware)**
- modular program.

Choices for the system

Language: F77
with almost no extension.

no direct call for OPEN/CLOSE
(use interface routine)

template

Data structure: Zebra with user interface
(SZ package)

Control files: KUIP (Kit for User Interface
package from CERN)

Code management: PATCHY
with minimum fanctionalities.
correction is done deck by deck

CERNLIB

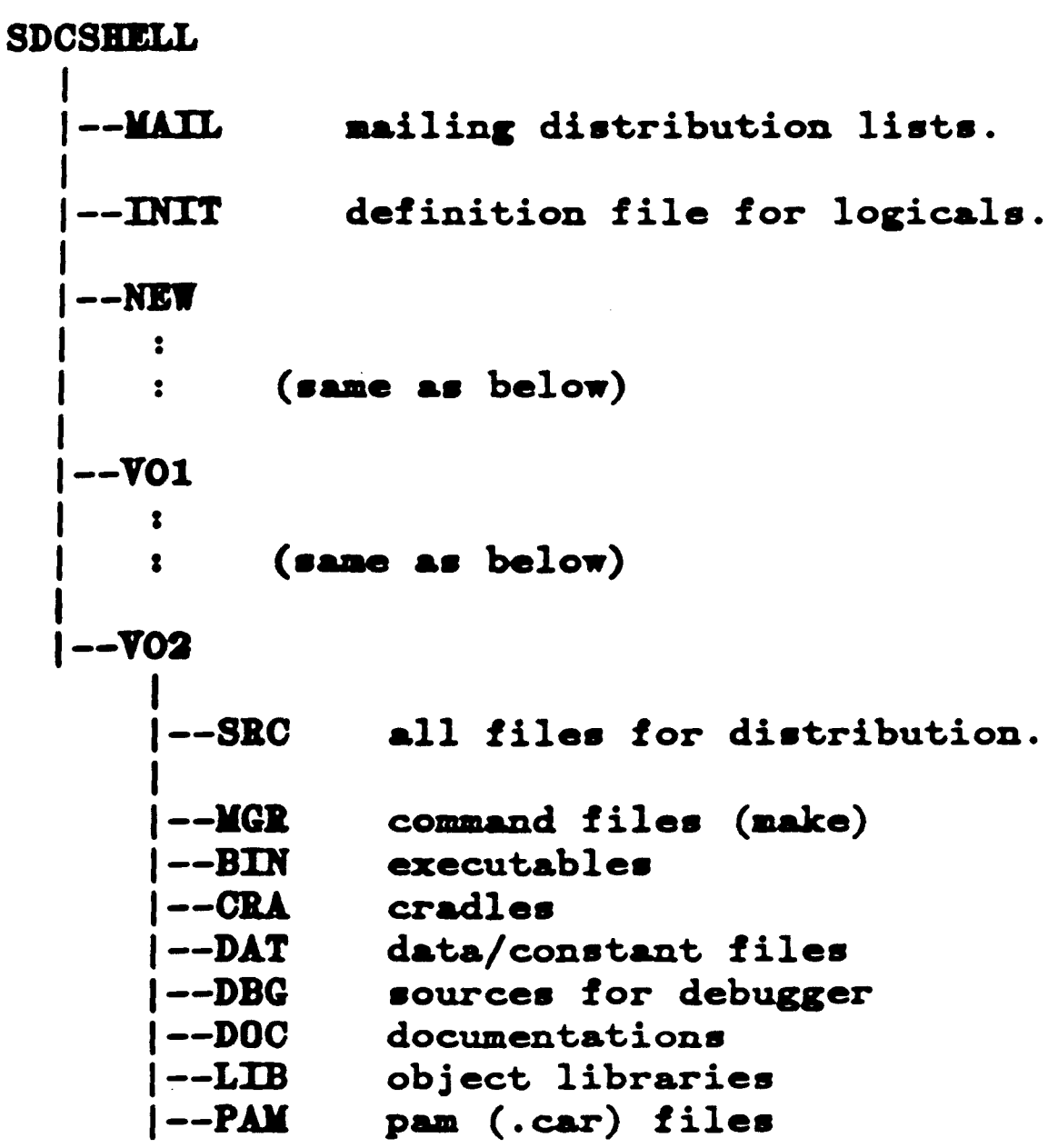
Organization of program packages:
divide to small packages and
assign two letters to each
package- package name,
suroutine and common names.
(first two letters)

--> already same two letters was found in CERNLIB

Distribution: No automatic distribution

```
+DECK,EXAMPL.
      SUBROUTINE EXAMPL
CA)==========================================(begin of short description)
C
CB)------------------------------------------(begin of long description)
C
C     Purpose and methods :
C
C     Called by:
C
C     Input arguments :
C
C     Input/output arguments :
C
C     Output arguments:
C
C     Side effects (other variables/banks modified, etc.)
C
C     Created:   dd-mmm-yyyy   author's name
C     Modified:  dd-mmm-yyyy   editor's name
C
CC)------------------------------------------(end of long description)
+SEQ,SHIMPNON.
CD)---------------------------------(declare common blocks, variables)
C  Common blocks
C+SEQ,seqname.
C  Variables in argument list
C  Save variables
C  Local variables
C  External references
C  Statement functions
C  Local data statements
CE)==========================================(end of declarations)
C
      RETURN
      END
```

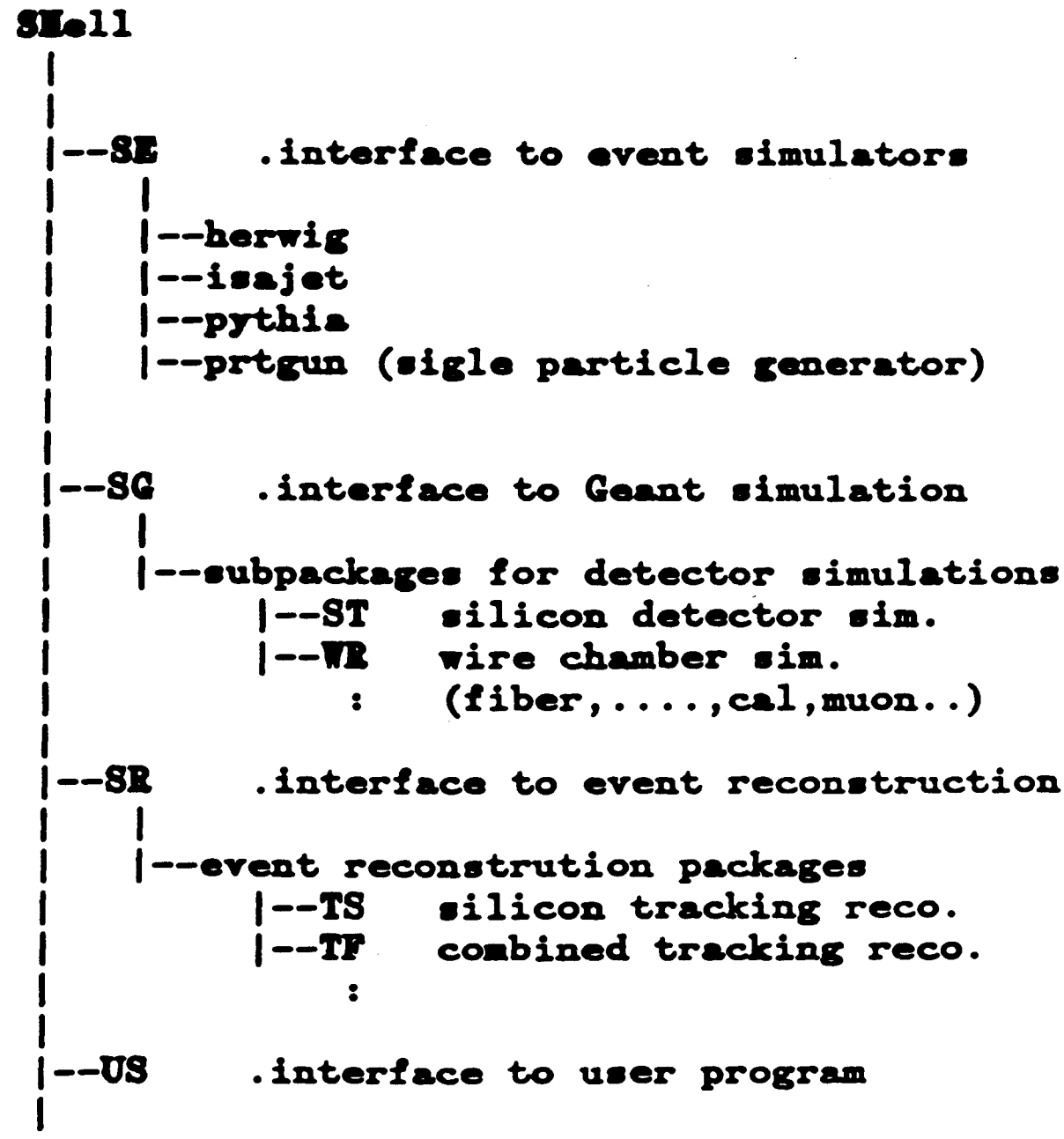
The SDC SHELL library on SSCVX1



Passive distribution: Each machine needs to copy files in SRC and split to other subdir.

Structure of SDCSIM

(combined packages under SHell)



- One can run only one of those packages by using switches at compile time or run time.
- SHell also supports event i/o using exchange mode of the Zebra FZ package.

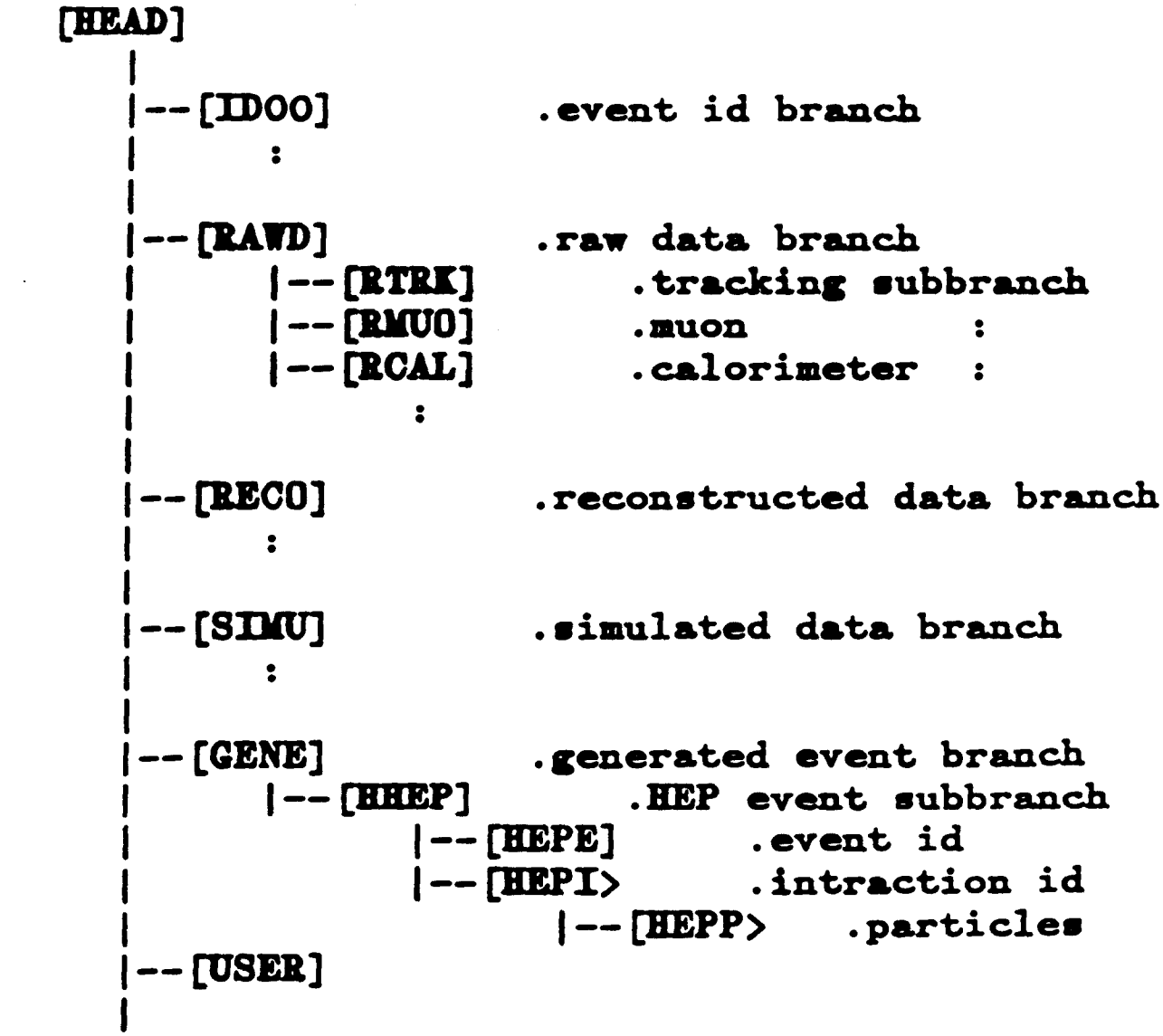
→ machine dependent codes are only in CERNLIB & SHell.

Data Structure for SDCSIM

Communication between program packages is done only through well defined data structure.

--> enforce the modular structure.

tree structure:

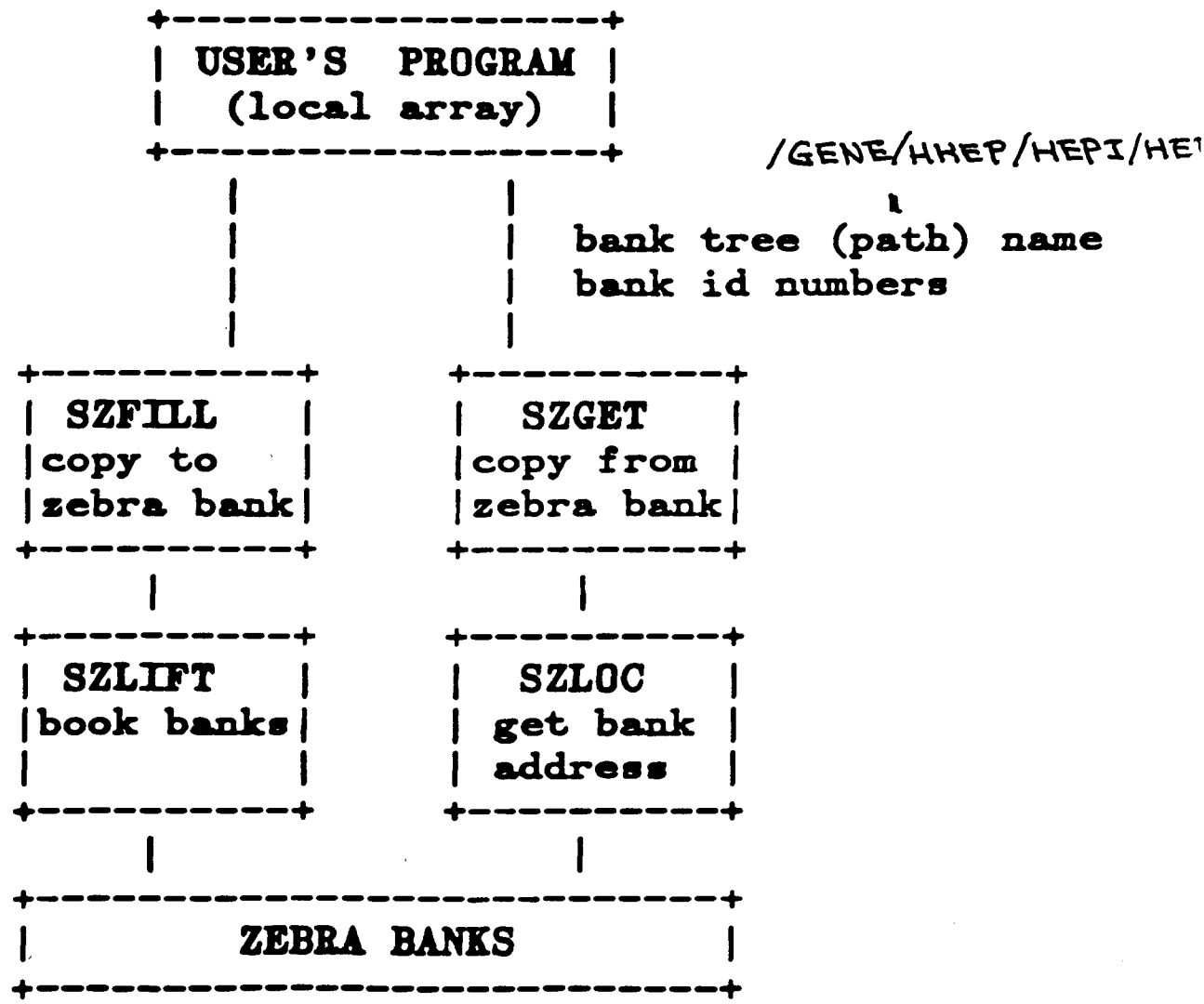


SDC SZ-package: Inteface to Zebra

1. Data structure is defined in a ASCII data file. (KUIP format)

   --> zebctl file

2. Data are always accessed through interface routines.



Bank Definition file (KUIP format)

```
*************************************************************
*          Definition of the SDC data structure             *
*                                                           *
*    history                                                *
*   ---------                                               *
*                                                           *
*************************************************************
SH/BKDIR  HEAD
*****************************************************************************
*              bank tree name              link num      F/V ver wrd format   *
*****************************************************************************
SH/BKTREE '-' '/IDOO'                      '/1'          'F' 1    0 '-I'
SH/BKTREE '-' '/IDOO/IDHH'                 '/1/1'        'F' 1    0 '-I'
SH/BKTREE '-' '/IDOO/IDHH/EVID'            '/1/1/1'      'F' 1   20 '-I'
SH/BKTREE '-' '/RAWD'                      '/2'          'F' 1    0 '-I'
SH/BKTREE '-' '/RAWD/RTRK'                 '/2/3'        'F' 1    0 '-I'
*
SH/BKTREE '-' '/RECO'                      '/3'          'F' 1    0 '-I'
SH/BKTREE '-' '/SIMU'                      '/4'          'F' 1    0 '-I'
SH/BKTREE '-' '/GENE'                      '/5'          'F' 1    0 '-I'
SH/BKTREE '-' '/USER'                      '/6'          'F' 1    0 '-I'
SH/BKTREE '-' '/GENE/HHEP'                 '/5/1'        'F' 1    0 '-I'
SH/BKTREE '-' '/GENE/HHEP/HEPE'            '/5/1/1'      'F' 1   20 '-I'
SH/BKTREE '-' '/GENE/HHEP/HEPI'            '/5/1/2'      'F' 1   20 '10I-F'
SH/BKTREE '-' '/GENE/HHEP/HEPI/HEPP'       '/5/1/2/1'    'F' 1   20 '10I-F'
*****************************************************************************
*  SHSP : Constants/Parameters
*****************************************************************************
SH/BKDIR  SHSP
SH/BKTREE '-' '/SHSP'                      '/1'          'F' 1    0 '-I'
*****************************************************************************
*  TEMP : Temporary data structure
*****************************************************************************
SH/BKDIR  TEMP
SH/BKTREE '-' '/TEMP'                      '/1'          'F' 1    0 '-I'
SH/BKCLOSE
```

3. Work space for local array is provided by 'malloc' in C run time libarary.

SHTSCR(bankid,base,index,size,retcode)
allocate temporary space

SHTSDL(bankid,base,index,retcode)
desallocate temporary space

Other Utilities in SHell

1. SUERR: error report and recovery

2. SUOPEN file open
   SUCLOS file close

SUOPEN(CHKEY,IRUN,CHTYP,CHSTAT,LUN,ISTAT)

Input arguments :

CHKEY	C*	key word in file server
IRUN	I	current run number
CHTYP	C*	type. ASCII,FZ,BINARY
CHSTAT	C*	status. OLD,NEW,READONLY

Output arguments:

LUN	I	logical unit number
ISTAT	I	status word

File is defined in control file(s):

```
SH/FILEDEF  KEY  zebra.ctl  000000 99999
```

3. Job control on pdsf at SSCL.

   supporting interactive/bckground
   batch/parallel modes.

Current Status of SDCSIM

1. Four event generators, herwig, isajet pythia and prtgun (single particle gen.) have been installed.

2. Detector simulation based on LOI design (>1/2 year old) has been working with Geant. Need to update the geometry.

3. Current major effors are developing reconstruction programs and study of perfomance of the detector.

4. The program is running on VAX and UNIX machines at many sites in SDC.

Future of the SDC SHell

1. Upgrade to full simulation-reconstruction program.

2. test bench for future computing and future software.

   --> evaluation of new code management system.

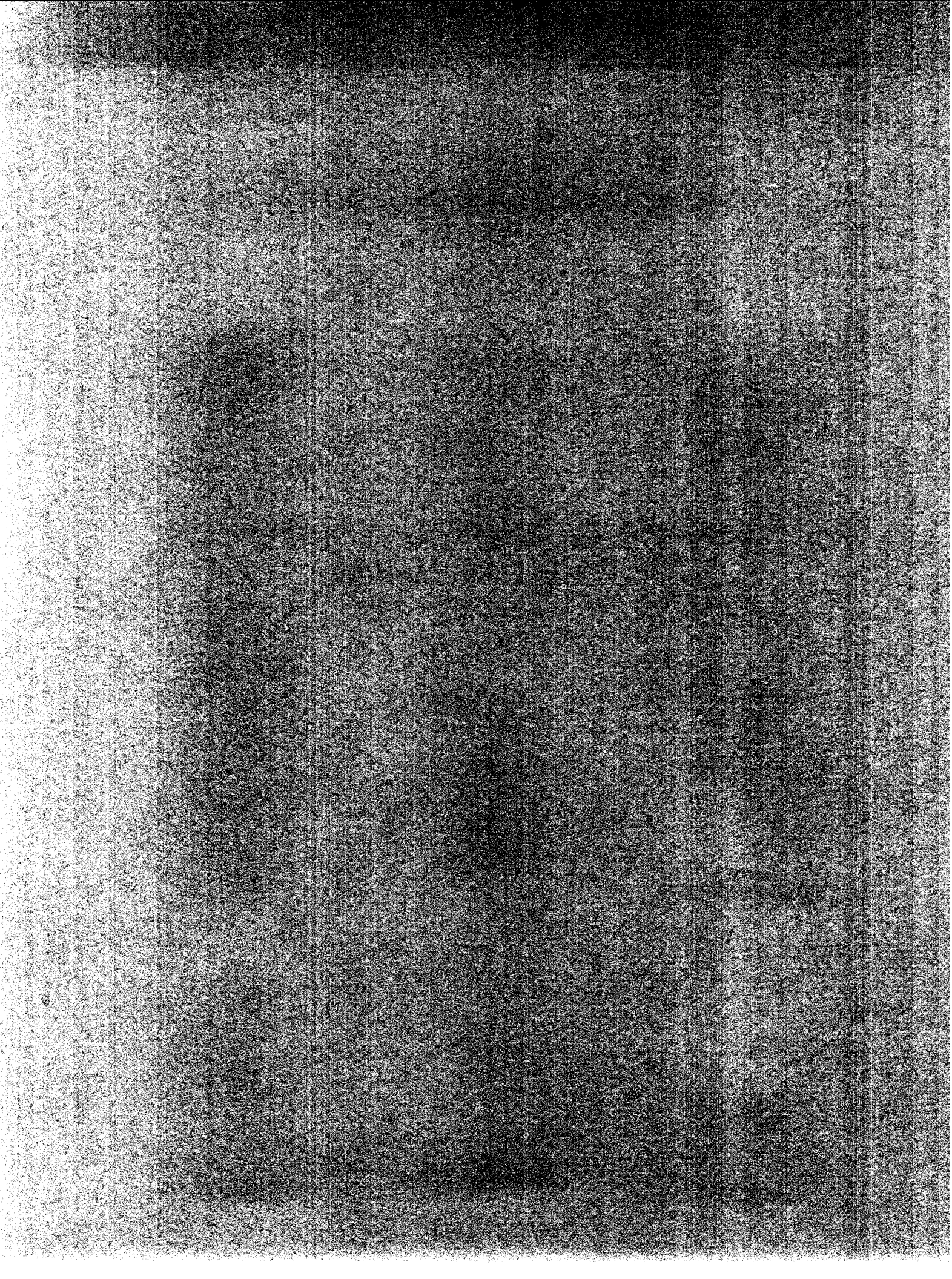
   --> parallel shell.

3. get experience on software development in widely distributed environment in terms of manpower and cpu's.

- Communication
- Standard
- modular

"Parallelization Software on PDSF"
"The PSHELL User Interface"

B. Traversat



Parallel/Distributing Methods Applied to the SDC Shell Simulation Code

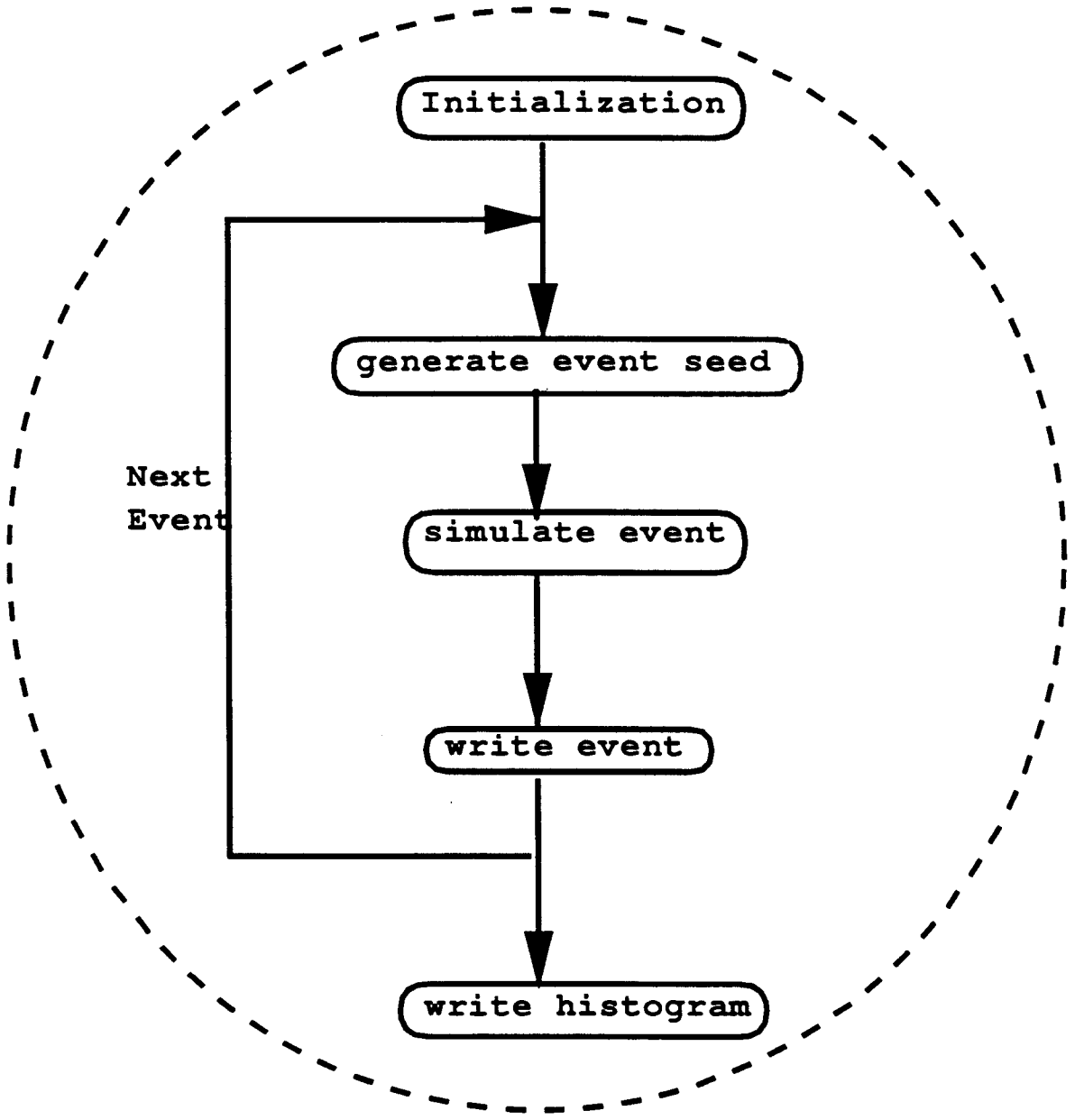
Bernard Traversat
Systems Development Group
Physics Research Division
SSC Laboratory

HEPLIB'91

Organization

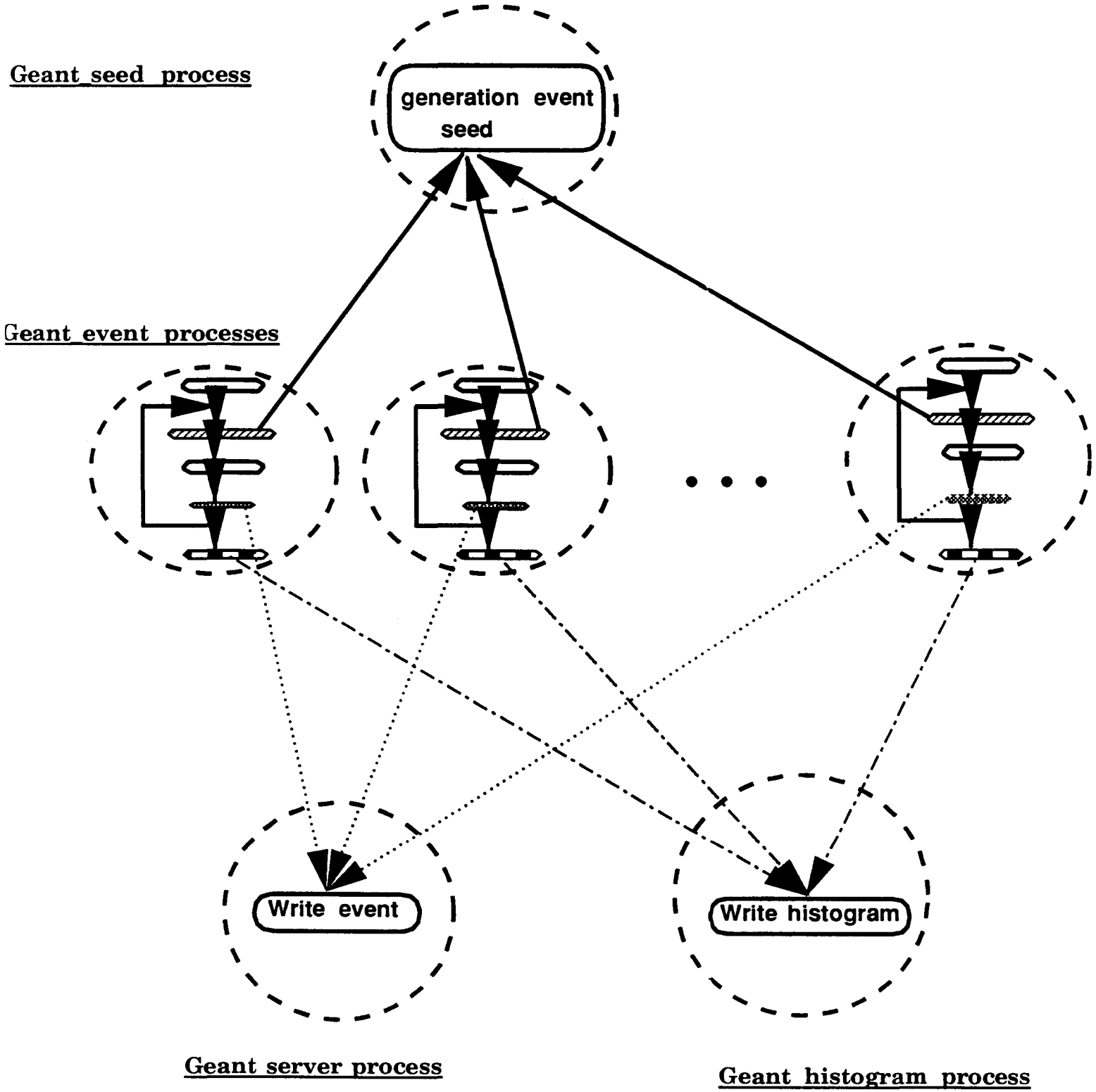
* "Event-Level" Parallelism for the SDC Shell
* What distributed/Parallel tools ?
* How to support parallel version ?
* What next ?

SDC Shell Simulation

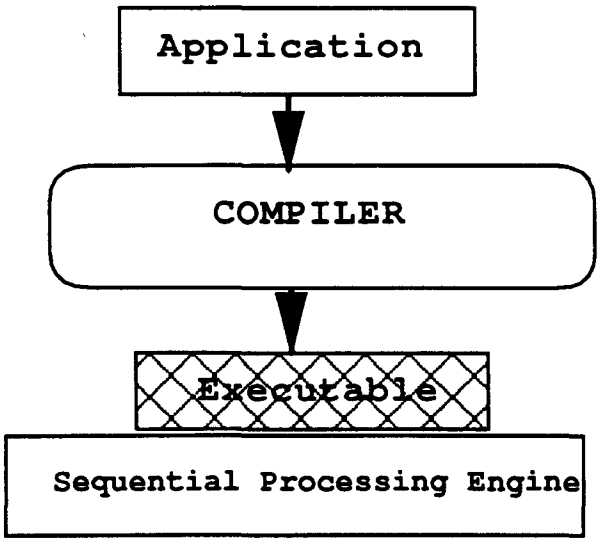


Process

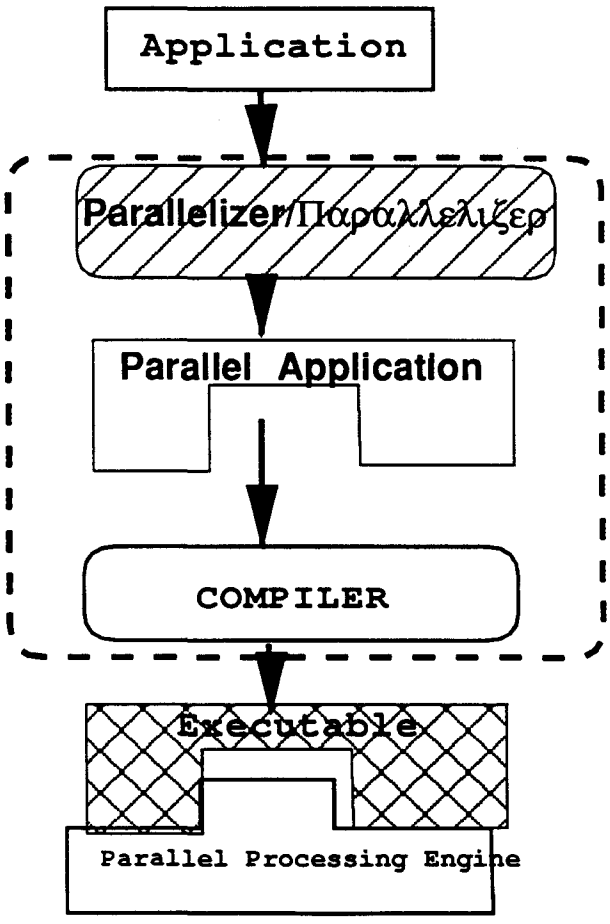
"Event-Level" Parallelism



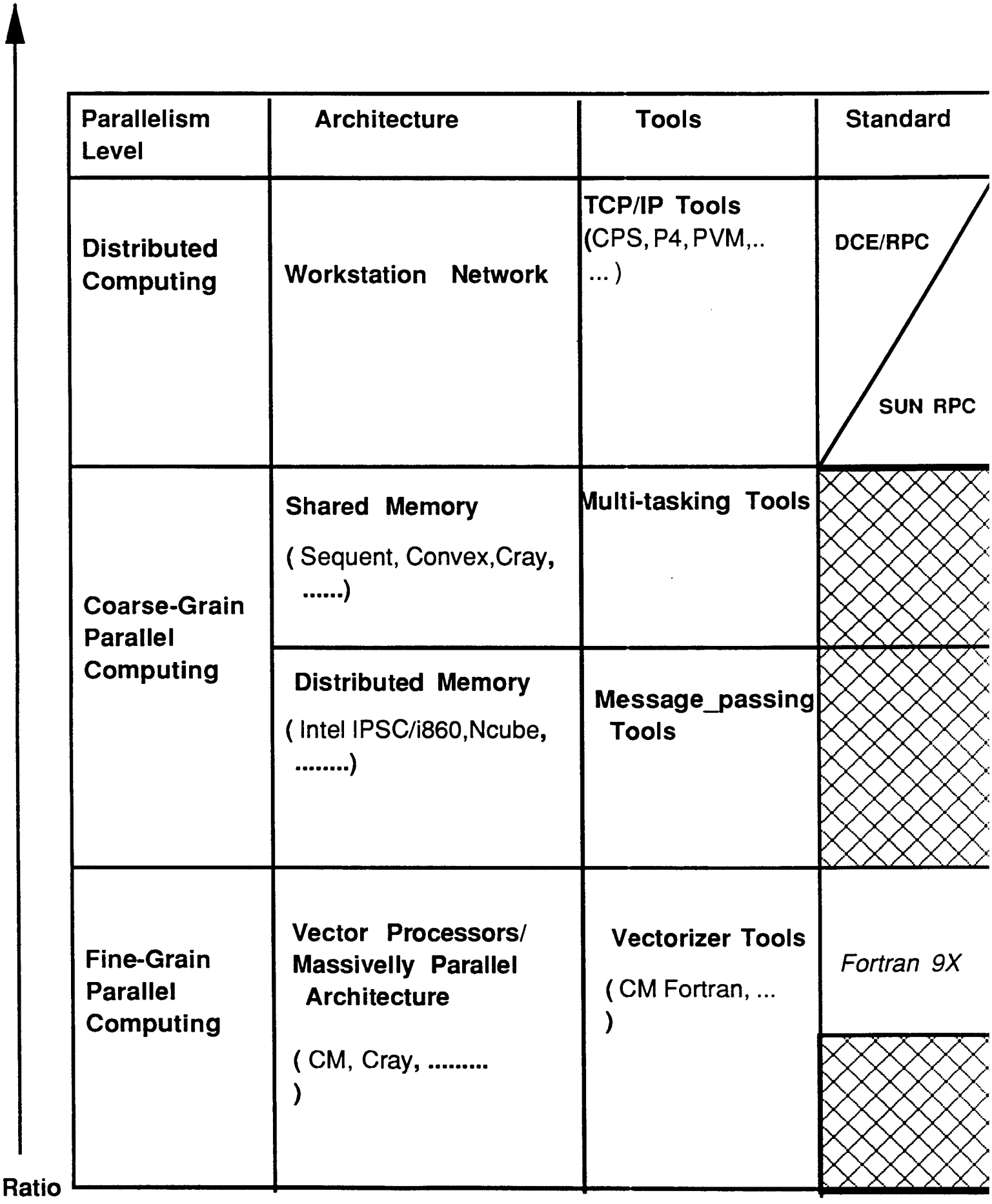
The "FUN" Sequential Computing



The "ΦΥΝ" Parallel Computing



What tools ?


Parallelism Level
Architecture
Tools
Standard
Distributed Computing
Workstation Network
TCP/IP Tools
(CPS, P4, PVM,..
...)
DCE/RPC
SUN RPC
Coarse-Grain Parallel Computing
Shared Memory
(Sequent, Convex,Cray, ...)
Multi-tasking Tools
Distributed Memory
(Intel IPSC/i860,Ncube, ...)
Message_passing Tools
Fine-Grain Parallel Computing
Vector Processors/ Massivelly Parallel Architecture
(CM, Cray, ...
)
Vectorizer Tools
(CM Fortran, ...
)
Fortran 9X
Ratio
Computing per Communications

EVENT LEVEL PARALLELISM -> DISTRIBUTED COMPUTING

What we have used:

* CPS : 1.1 .. 2.x

  work fine for (<15 processes)
  but too many whistles and bells after

  "nice" job manager

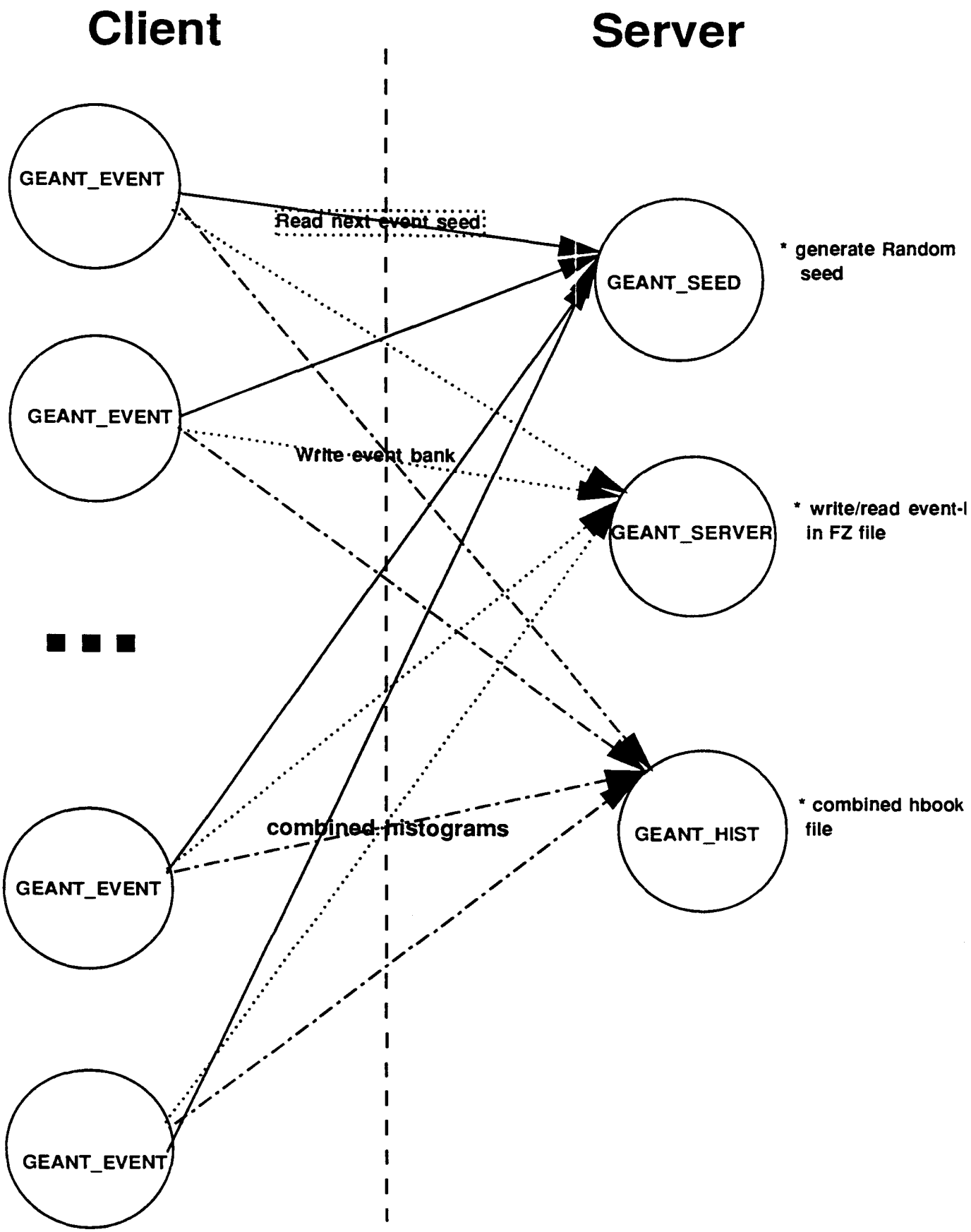
* RPC: (Sun)

  Preprocessor (rpcgen)

  minimum functionality for Distributing Computing

  Emerging standard for distributed computing

RPC Implementation



```
   Created :  16-aug-1991   B.traversat   First release

*/

const MAXSEED = 2;

struct seed {
  int status;
  int event_nb;
  int seed[MAXSEED];
};


program SEEDPROG {
   version SEEDVERS {
      int  RPCGEANT_PID(int) = 1;
      seed RPCGEANT_SEED(int) = 2;
      int  RPCKILL_SEED(int)  = 3;
   }= 1;
} = 0x20000010;



const MAXSEM = 5;                    /* max number of semap
const MAXEVENT = 1000000;            /* max size of a bank

struct sem {
  int sema[MAXSEM];
};

struct event {
  opaque bank<MAXEVENT>;
};

program SERVERPROG {
   version SERVERVERS {
      int  RPCSERVER_POST(int) = 1;
      int  RPCSERVER_WAIT(int) = 2;
      int  RPCSERVER_WRITE(event) = 3;
      int  RPCKILL_FZSERVER(int)  = 4;
   }= 1;
} = 0x20000020;
```

```
#include <rpc/rpc.h>
#include <sys/time.h>
#include "geant_seed.h"

/* Default timeout can be changed using clnt_control() */
static struct timeval TIMEOUT = { 25, 0 };

int *
rpcgeant_pid_1(argp, clnt)
        int *argp;
        CLIENT *clnt;
{
        static int res;

        bzero((char *)&res, sizeof(res));
        if (clnt_call(clnt, RPCGEANT_PID, xdr_int, argp, xdr_int, &res, TIMEOUT) !
                return (NULL);
        }
        return (&res);
}


seed *
rpcgeant_seed_1(argp, clnt)
        int *argp;
        CLIENT *clnt;
{
        static seed res;

        bzero((char *)&res, sizeof(res));
        if (clnt_call(clnt, RPCGEANT_SEED, xdr_int, argp, xdr_seed, &res, TIMEOUT)
                return (NULL);
        }
        return (&res);
}


int *
rpckill_seed_1(argp, clnt)
        int *argp;
        CLIENT *clnt;
{
        static int res;

        bzero((char *)&res, sizeof(res));
        if (clnt_call(clnt, RPCKILL_SEED, xdr_int, argp, xdr_int, &res, TIMEOUT) !
                return (NULL);
        }
        return (&res);
}
```

```
        case RPCKILL_SEED:
                xdr_argument = xdr_int;
                xdr_result = xdr_int;
                local = (char *(*)()) rpckill_seed_1;
                break;

        default:
                svcerr_noproc(transp);
                return;
        }
        bzero((char *)&argument, sizeof(argument));
        if (!svc_getargs(transp, xdr_argument, &argument)) {
                svcerr_decode(transp);
                return;
        }
        result = (*local)(&argument, rqstp);
        if (result != NULL && !svc_sendreply(transp, xdr_result, result)) {
                svcerr_systemerr(transp);
        }
        if (!svc_freeargs(transp, xdr_argument, &argument)) {
                (void)fprintf(stderr, "unable to free arguments\n");
                exit(1);
        }
}
```

```
#include <stdio.h>
#include <rpc/rpc.h>
#include "geant_seed.h"

static void seedprog_1();

main()
{
        SVCXPRT *transp;

        (void)pmap_unset(SEEDPROG, SEEDVERS);

        transp = svcudp_create(RPC_ANYSOCK);
        if (transp == NULL) {
                (void)fprintf(stderr, "cannot create udp service.\n");
                exit(1);
        }
        if (!svc_register(transp, SEEDPROG, SEEDVERS, seedprog_1, IPPROTO_UDP)) {
                (void)fprintf(stderr, "unable to register (SEEDPROG, SEEDVERS, udp
                exit(1);
        }

        transp = svctcp_create(RPC_ANYSOCK, 0, 0);
        if (transp == NULL) {
                (void)fprintf(stderr, "cannot create tcp service.\n");
                exit(1);
        }
        if (!svc_register(transp, SEEDPROG, SEEDVERS, seedprog_1, IPPROTO_TCP)) {
                (void)fprintf(stderr, "unable to register (SEEDPROG, SEEDVERS, tcp
                exit(1);
        }
        svc_run();
        (void)fprintf(stderr, "svc_run returned\n");
        exit(1);
}

static void
seedprog_1(rqstp, transp)
        struct svc_req *rqstp;
        SVCXPRT *transp;
{
        union {
                int rpcgeant_pid_1_arg;
                int rpcgeant_seed_1_arg;
                int rpckill_seed_1_arg;
        } argument;
        char *result;
        bool_t (*xdr_argument)(), (*xdr_result)();
        char *(*local)();

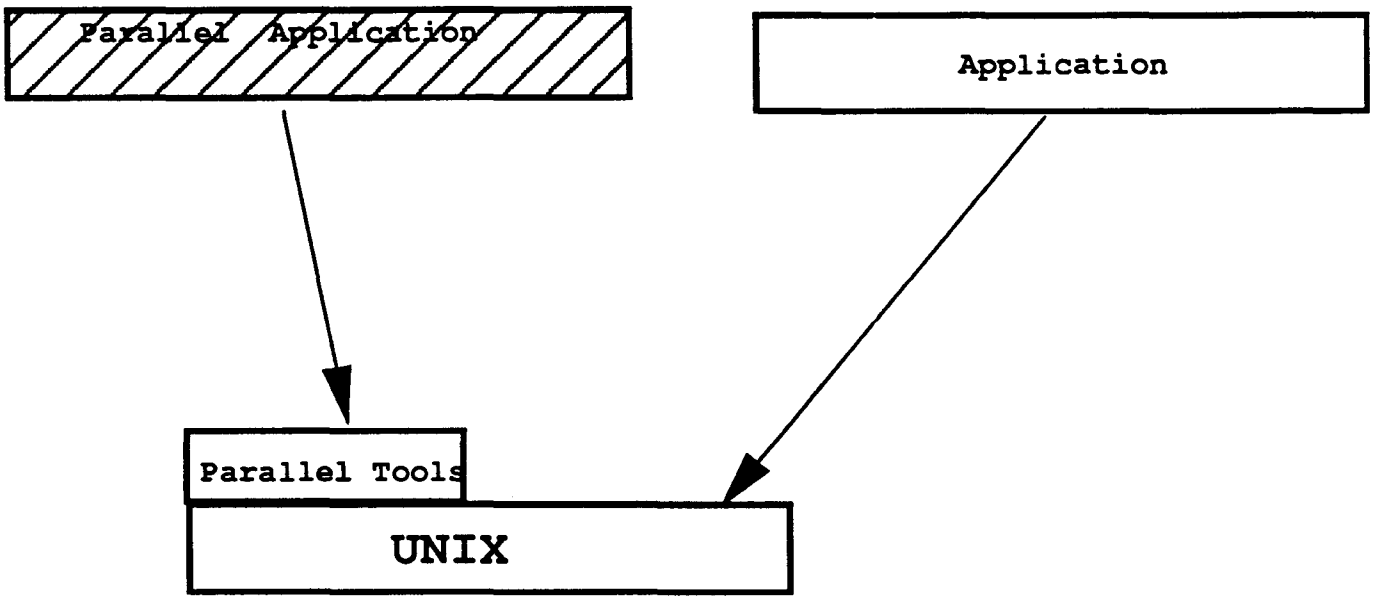
        switch (rqstp->rq_proc) {
        case NULLPROC:
                (void)svc_sendreply(transp, xdr_void, (char *)NULL);
                return;

        case RPCGEANT_PID:
                xdr_argument = xdr_int;
                xdr_result = xdr_int;
                local = (char *(*)()) rpcgeant_pid_1;
                break;

        case RPCGEANT_SEED:
                xdr_argument = xdr_int;
                xdr_result = xdr_seed;
                local = (char *(*)()) rpcgeant_seed_1;
                break;
```

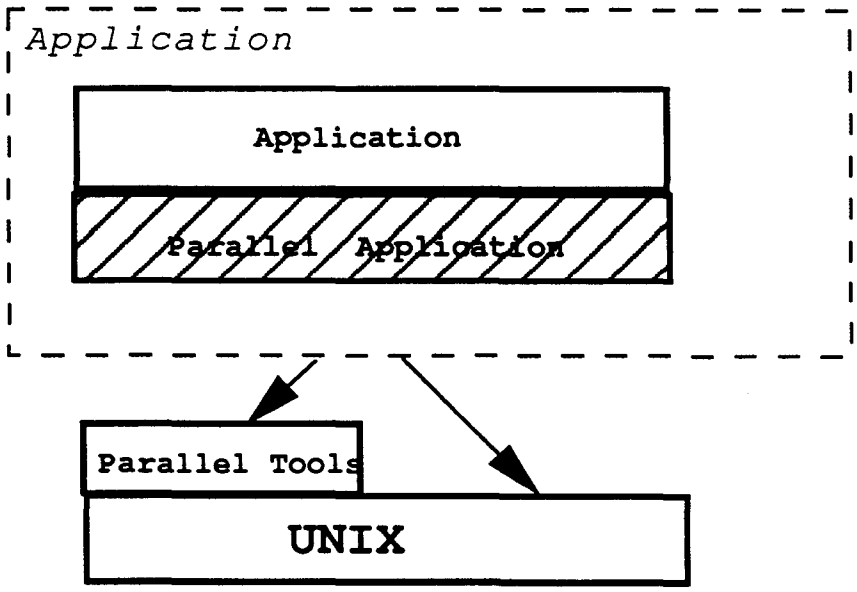
What support ?

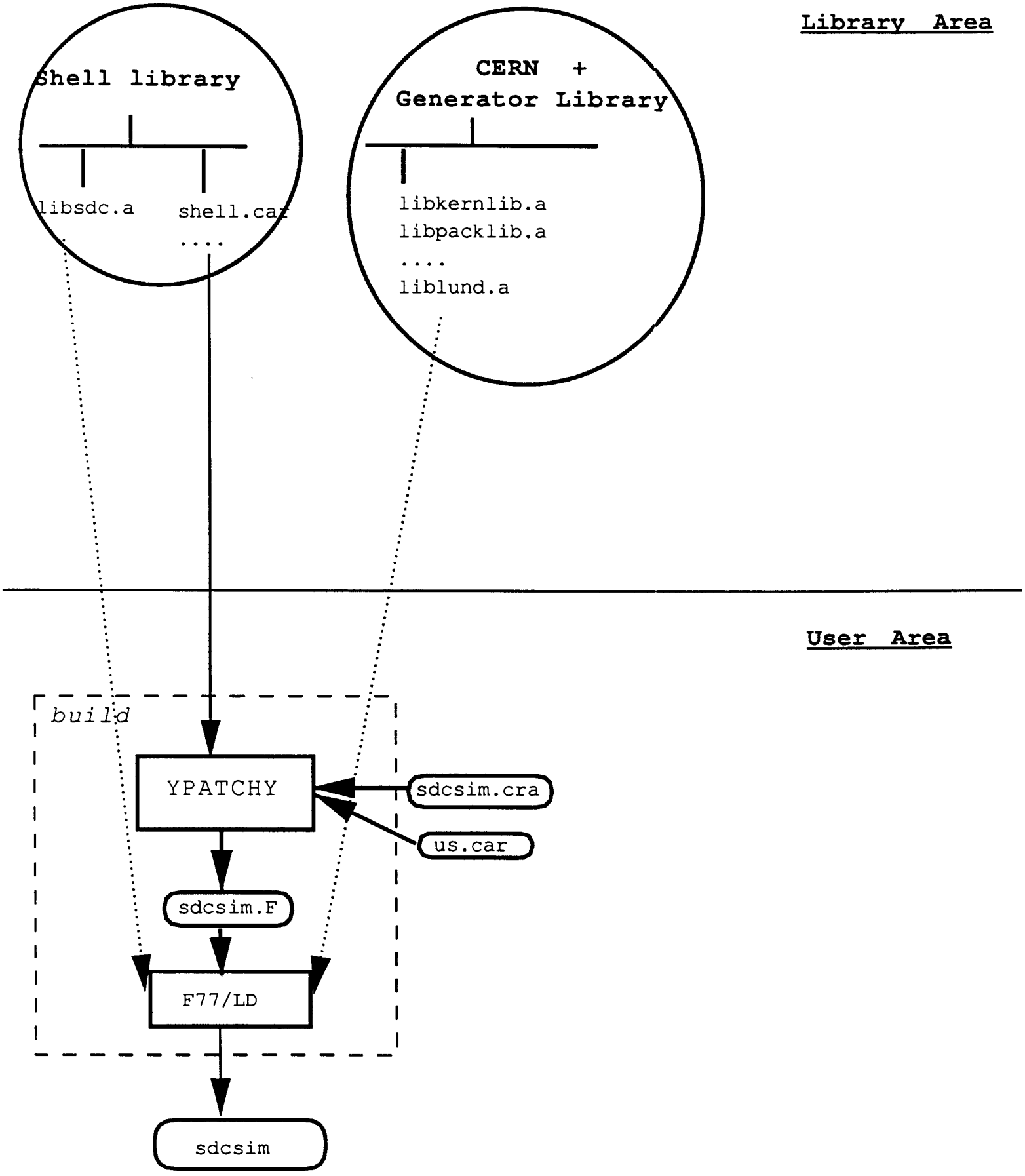
1) "Dual" Support

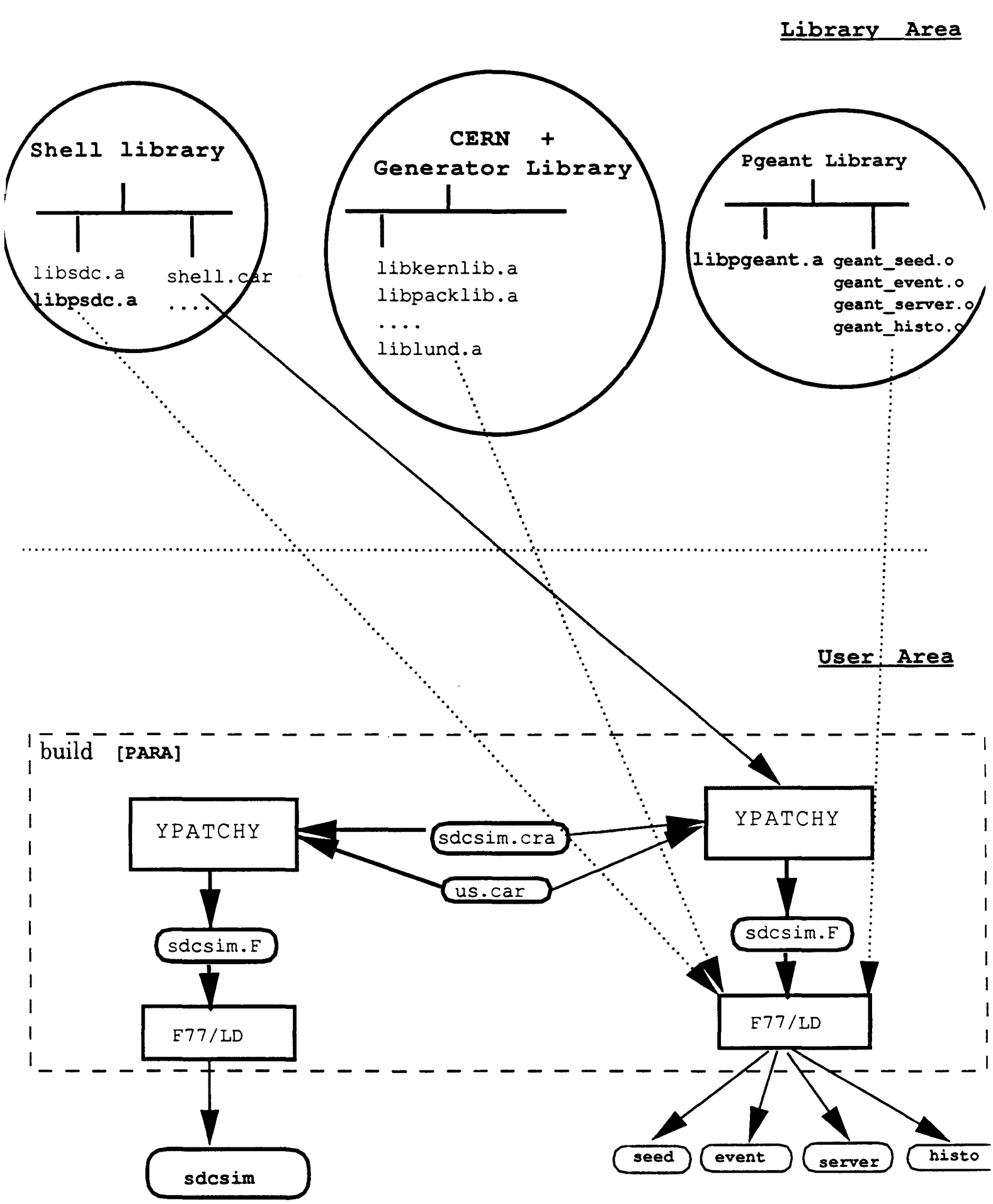


---

2) "Transparent" Support




Library Area
Shell library
libsdc.a
shell.car
....
CERN +
Generator Library
libkernlib.a
libpacklib.a
....
liblund.a
User Area
build
YPATCHY
sdcsim.cra
us.car
sdcsim.F
F77/LD
sdcsim


Library Area
Shell library
libsdc.a
libpsdc.a
shell.car
....
CERN +
Generator Library
libkernlib.a
libpacklib.a
....
liblund.a
Pgeant Library
libpgeant.a
geant_seed.o
geant_event.o
geant_server.o
geant_histo.o
User Area
build [PARA]
YPATCHY
sdcsim.cra
us.car
YPATCHY
sdcsim.F
F77/LD
sdcsim.F
F77/LD
sdcsim
seed
event
server
histo

```
+-------------------------------------------------------------+
|              SDC   Shell Script utility                     |
+-------------------------------------------------------------+
|                                                             |
| Enter <quit> at any prompt to stop this command file.       |
|                                                             |
+-------------------------------------------------------------+
The current directory is :   /tmp_mnt/home/fs0/physics/bernard
This directory will be used to execute the Shell program. You must
have previously installed the Shell on that directory.

Please enter the program name                    [sdcsim]:
Do you which to build the program?  (yes/no)        [no]:
Do you wish to run the program?    (yes/no)        [yes]:
Do you wish to have a log listing   (yes/no)        [no]:
Select job mode  (inter/back/batch/para)         [inter]: para
Due to current CPS instabilities the process limit is fixed to  [4]:
The SHELL program will be executed with the arguments:
Program name                                = sdcsim
The program is built                (y/n) = no
Run program                         (y/n) = yes
Keep all listings (verbose mode)    (y/n) = no
Job mode   (inter/back/batch/para)        = para
Number of Processors                      = 4

Are you sure you want to continue?  (cont/quit)  [cont]:
```

What next ?

1) distributed -> parallel computing

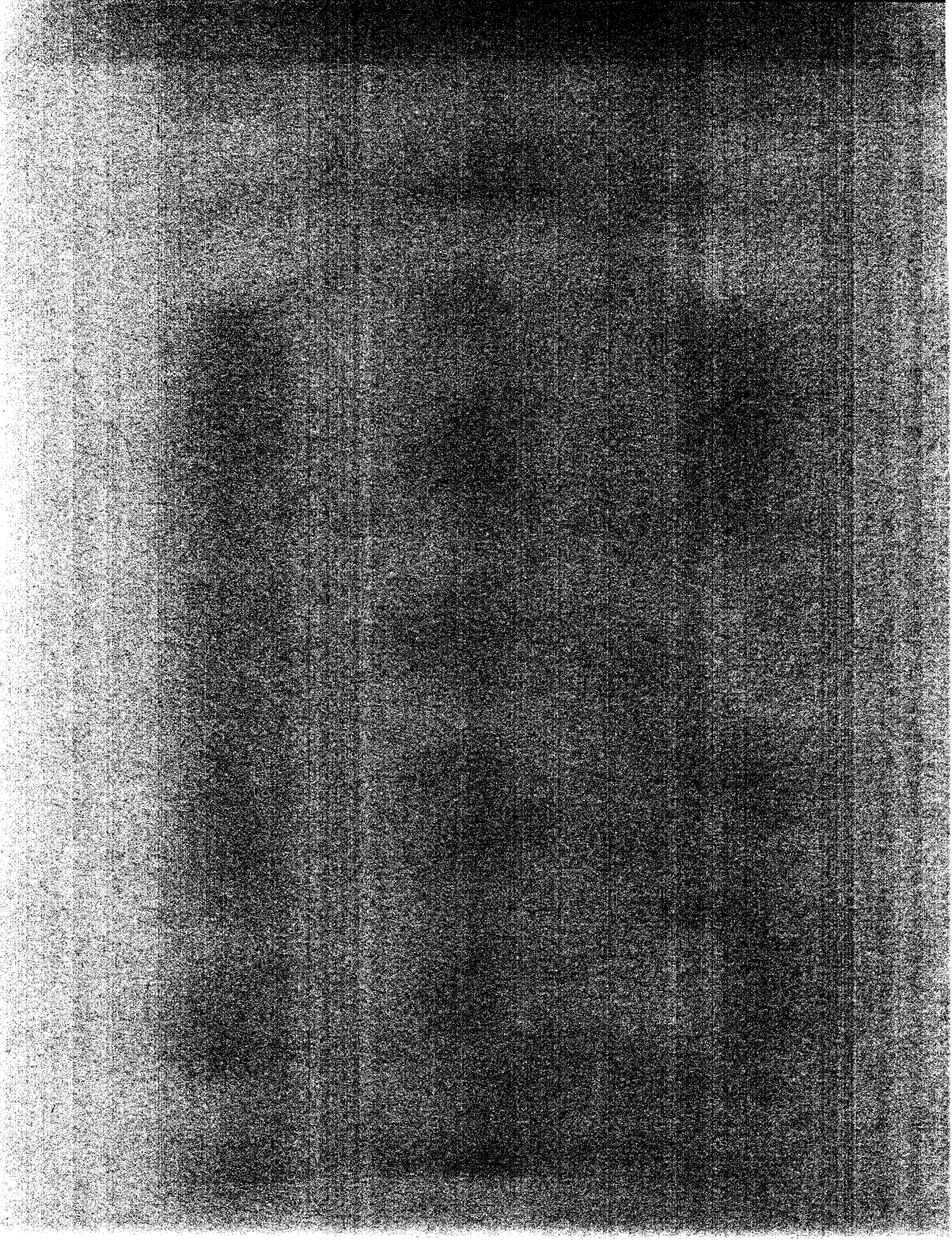
X events vs. 1 event ?

2) General or particular "schema" for Event-Level Parallelism

pshell vs pevent_level ?

"A Parallel Version of GEANT"

P. Rulon
S. Linn

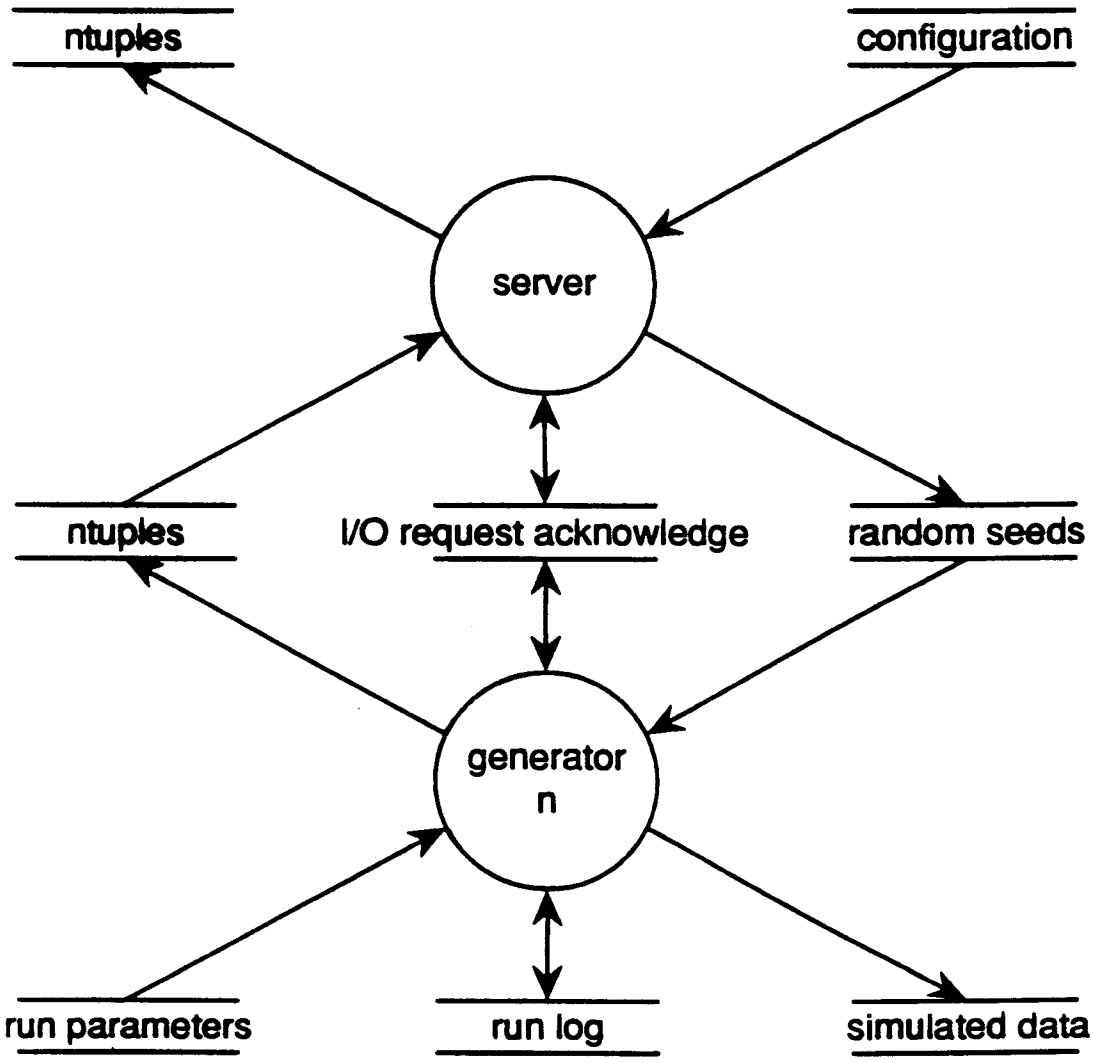


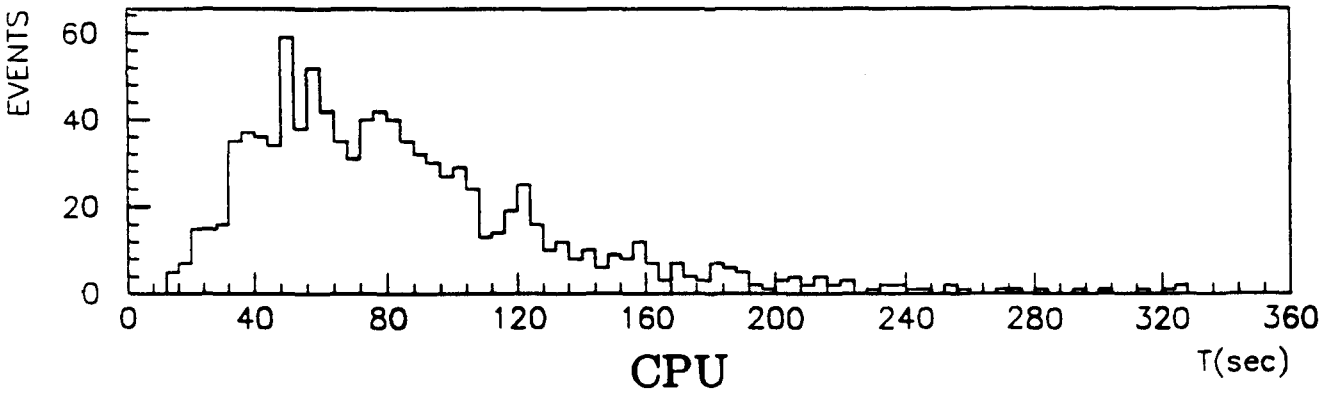
Performance of a Distributed Geant Application

Stephan L. Linn and Phillip J. Rulon

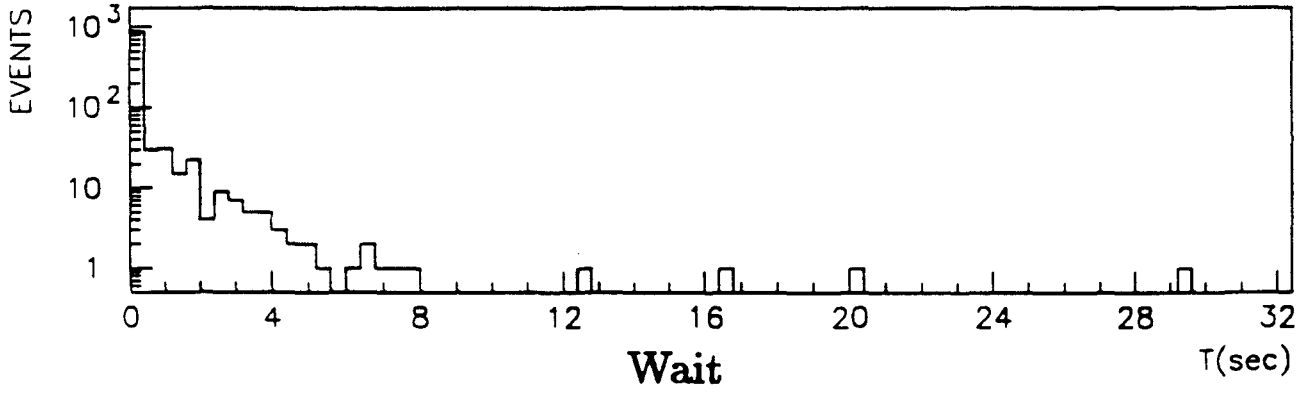
Supercomputer Computations Research Institute
Florida State University
Tallahassee

Implementation

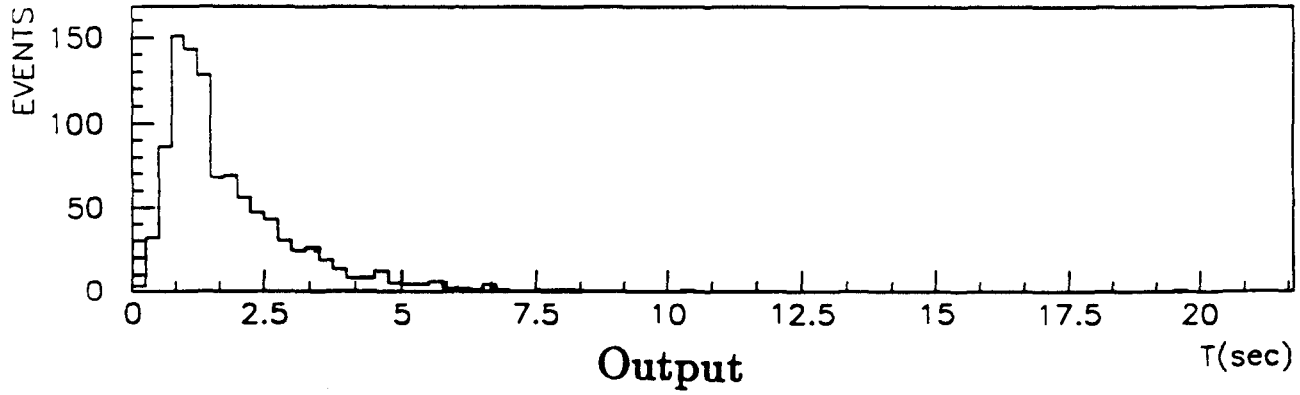




CPU



Wait



Output

8 generators, 100 K/event

Definitions

Average time per event can be described by:

- (t'_e): cpu-time per event
- (t_e): real-time per event
- (t_o): real-time for output
- (t_q): real-time waiting to do output

Average total time per event is:

$$t = t_e + t_o + t_q,$$

efficiency(η) is

$$\eta = \frac{t_e + t_o}{t_e + t_o + t_q}.$$

$$Speedup = \eta N$$

Model

The probability that a process has to wait is

$$p = \frac{(N-1)t_o}{t}.$$

Maximum useful nodes

$$N_{\max} = \frac{t_o + t_e}{t_o}.$$

Average waiting time is

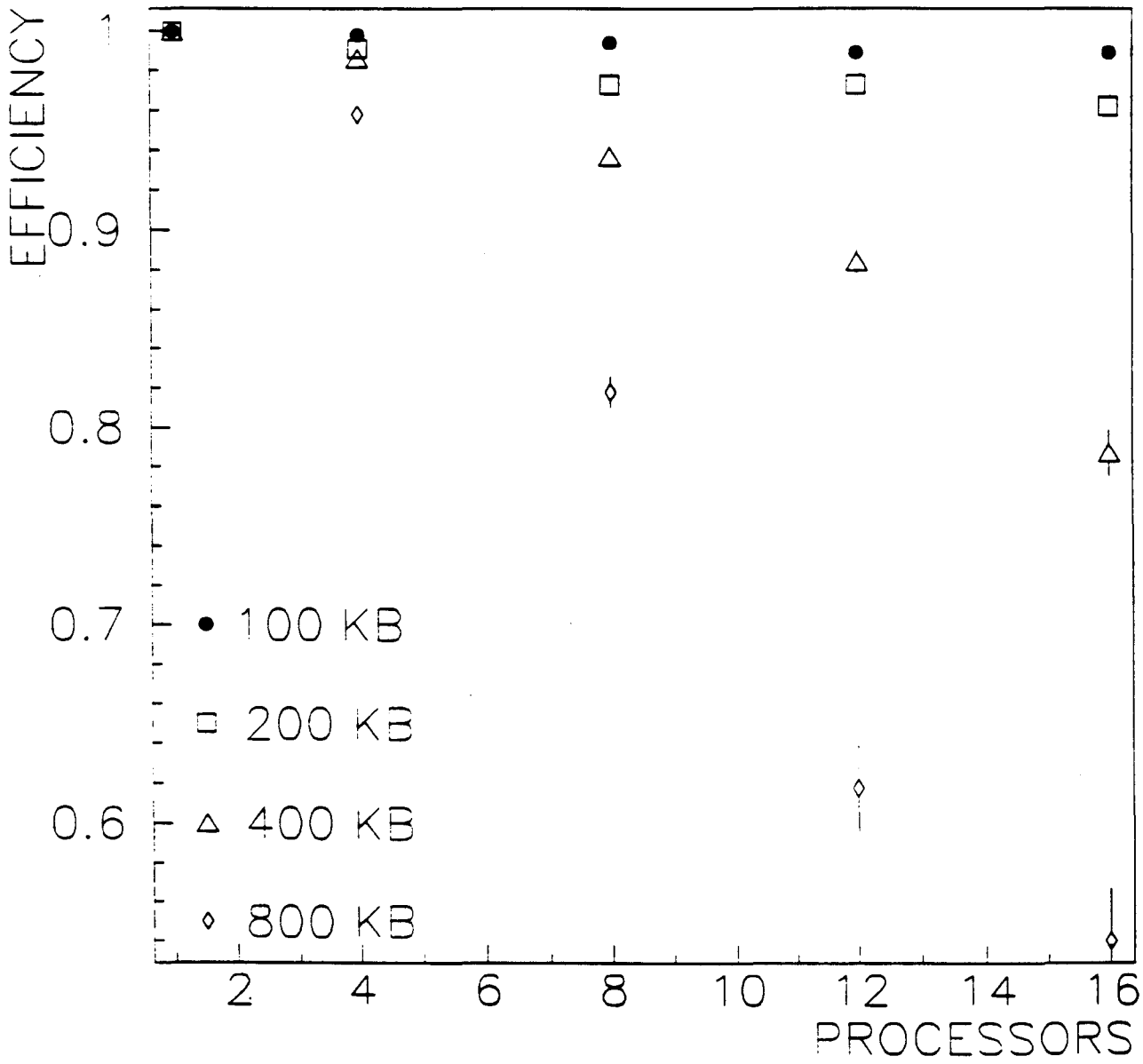
$$t_q = \frac{t_o}{2} \times \frac{p}{1-p}.$$

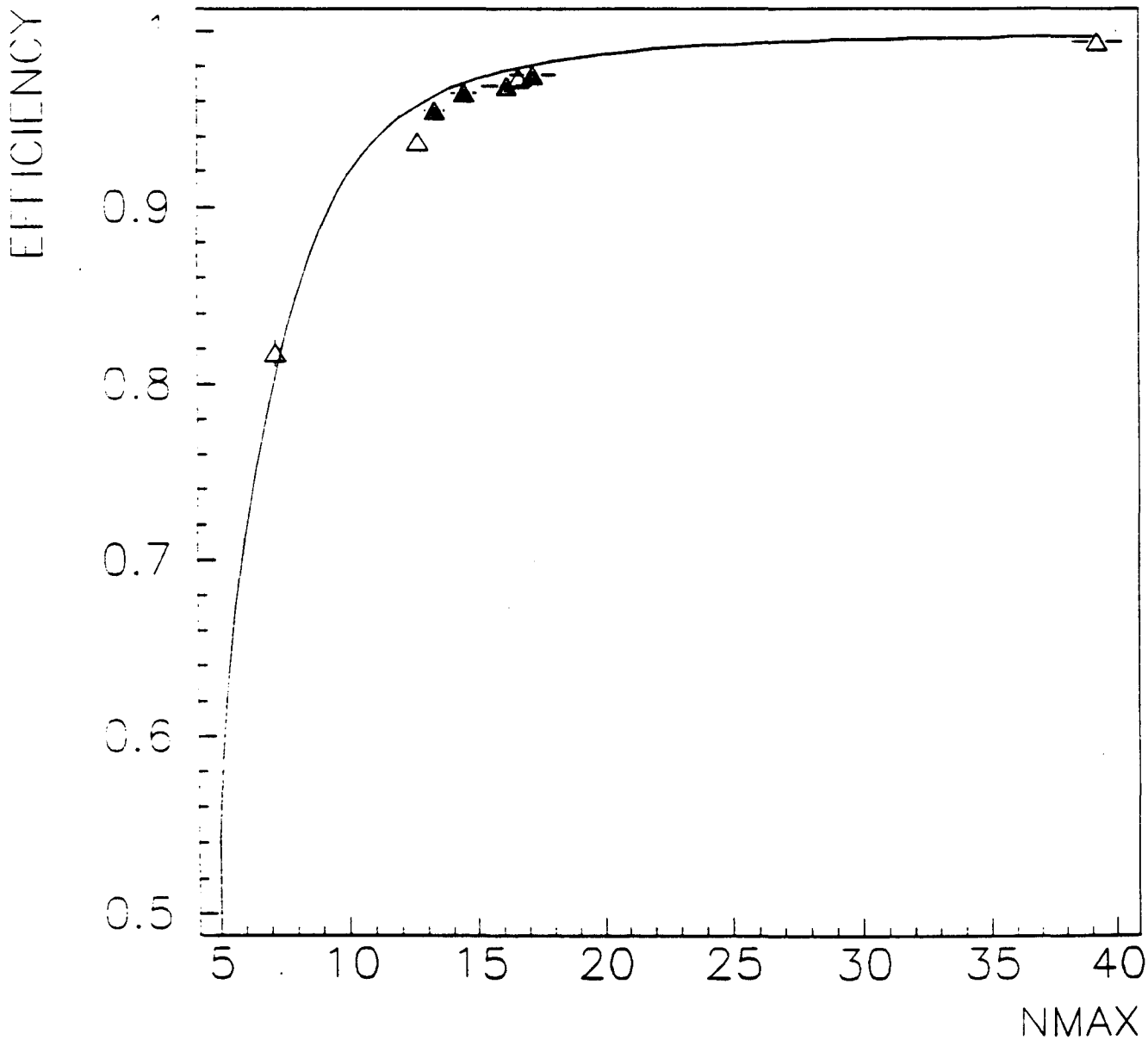
where $t_o/2$ is the average waiting time for one other process. Combining these with the definition of t and η

$$\eta^2 \frac{N-1}{2}(2N_{\max} - 1) - \eta N_{\max}(N - 1 - N_{\max}) + N_{\max}^2 = 0.$$

Note that

$$t_o/t_e = \frac{\text{instructions/second}}{\text{bytes/second}} \times \frac{\text{bytes/event}}{\text{instructions/event}}$$


EFFICIENCY
1
0.9
0.8
0.7
0.6
100 KB
200 KB
400 KB
800 KB
2
4
6
8
10
12
14
16
PROCESSORS


EFFICIENCY
0.9
0.8
0.7
0.6
0.5
5
10
15
20
25
30
35
40
NMAX

Observations

Need for low level I/O.

Parametric message handling.

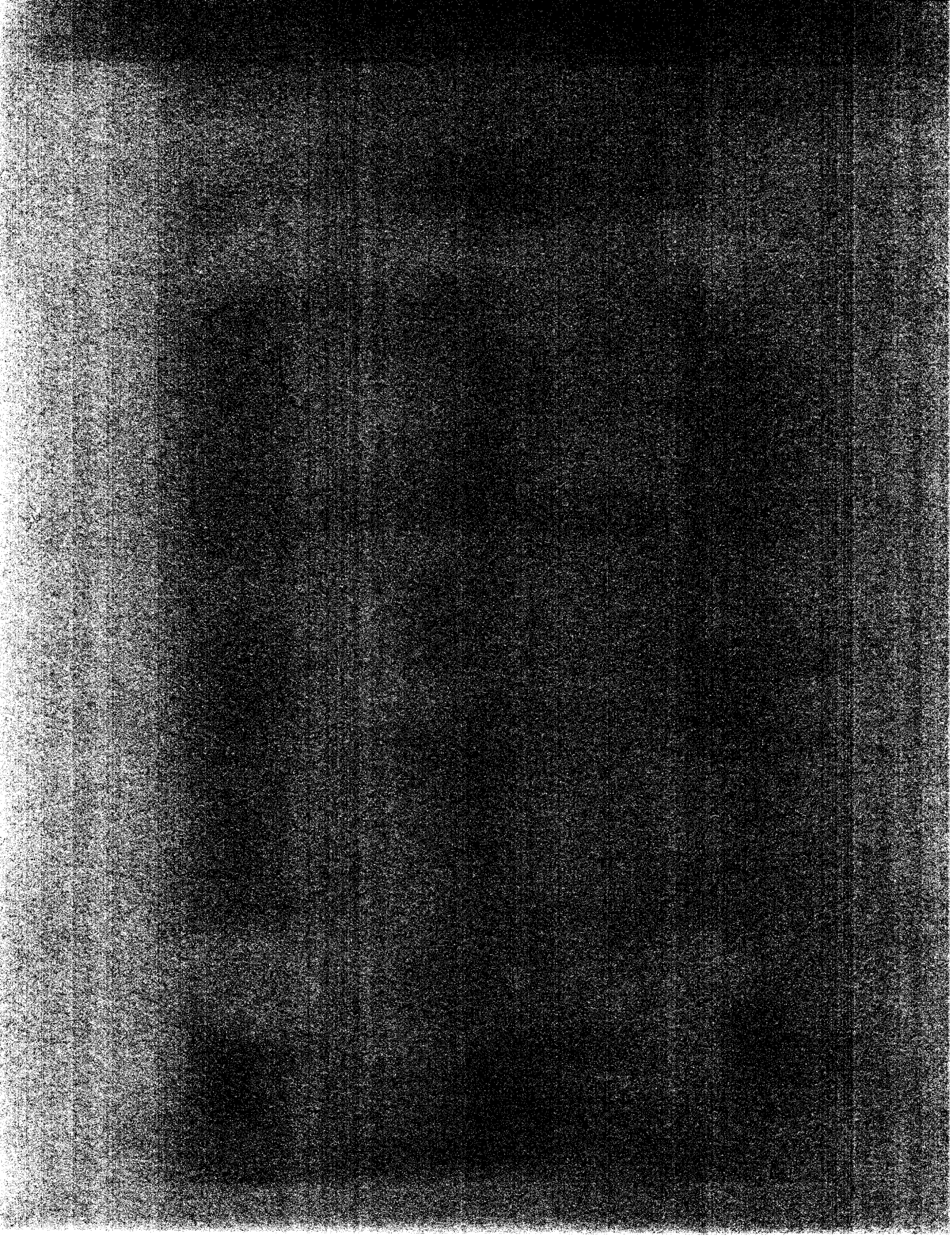
SDC tracking scales to $N_{max} \approx 250$ @ $\eta = .9$.

Integrated solution to distribution.

Error recovery.

"Calor on PDSF and Intel"

T. Gabriel
T. Handler
B. Moore



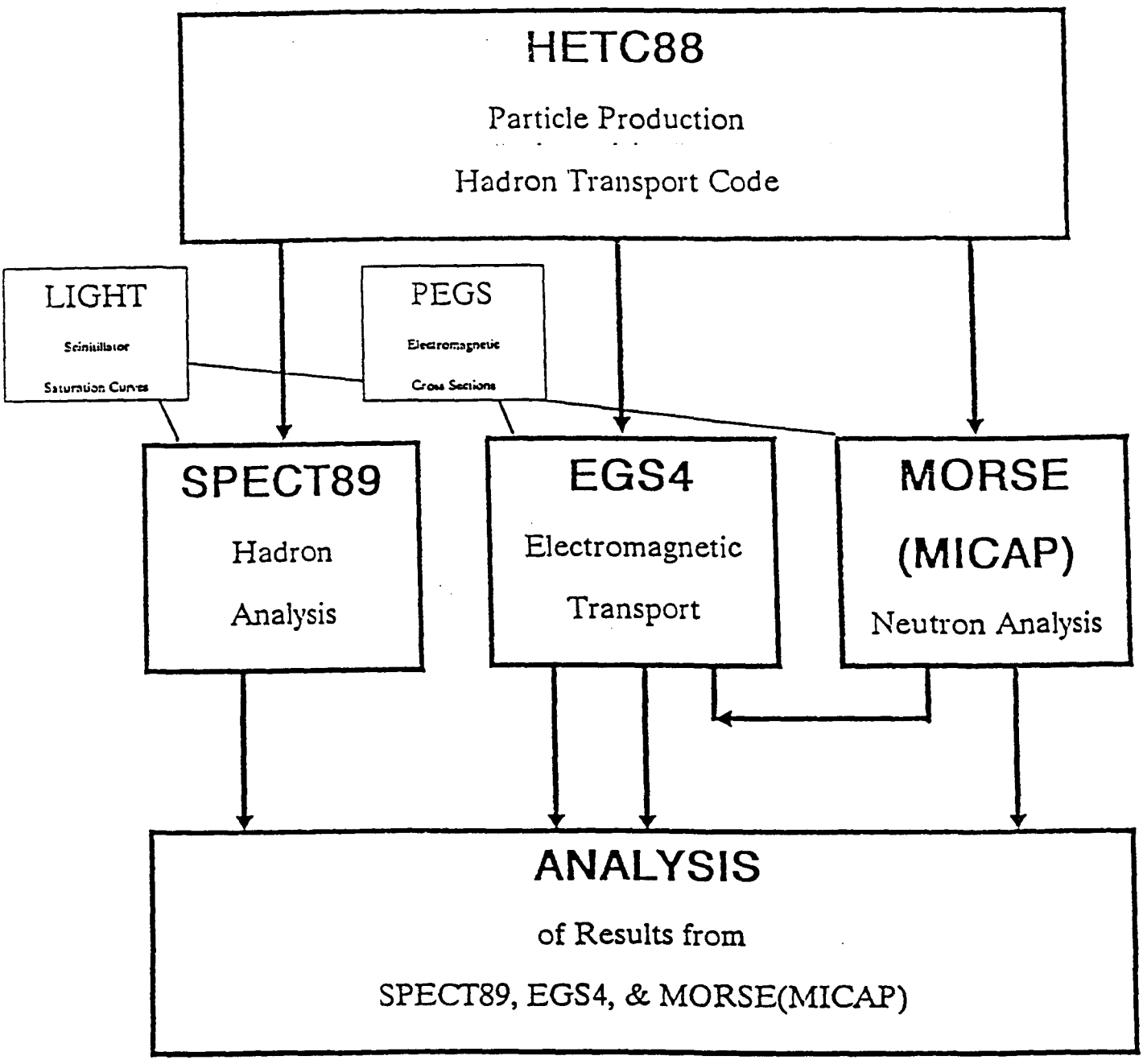
Installation of the CALOR System on the SSC Farm and Intel Machines

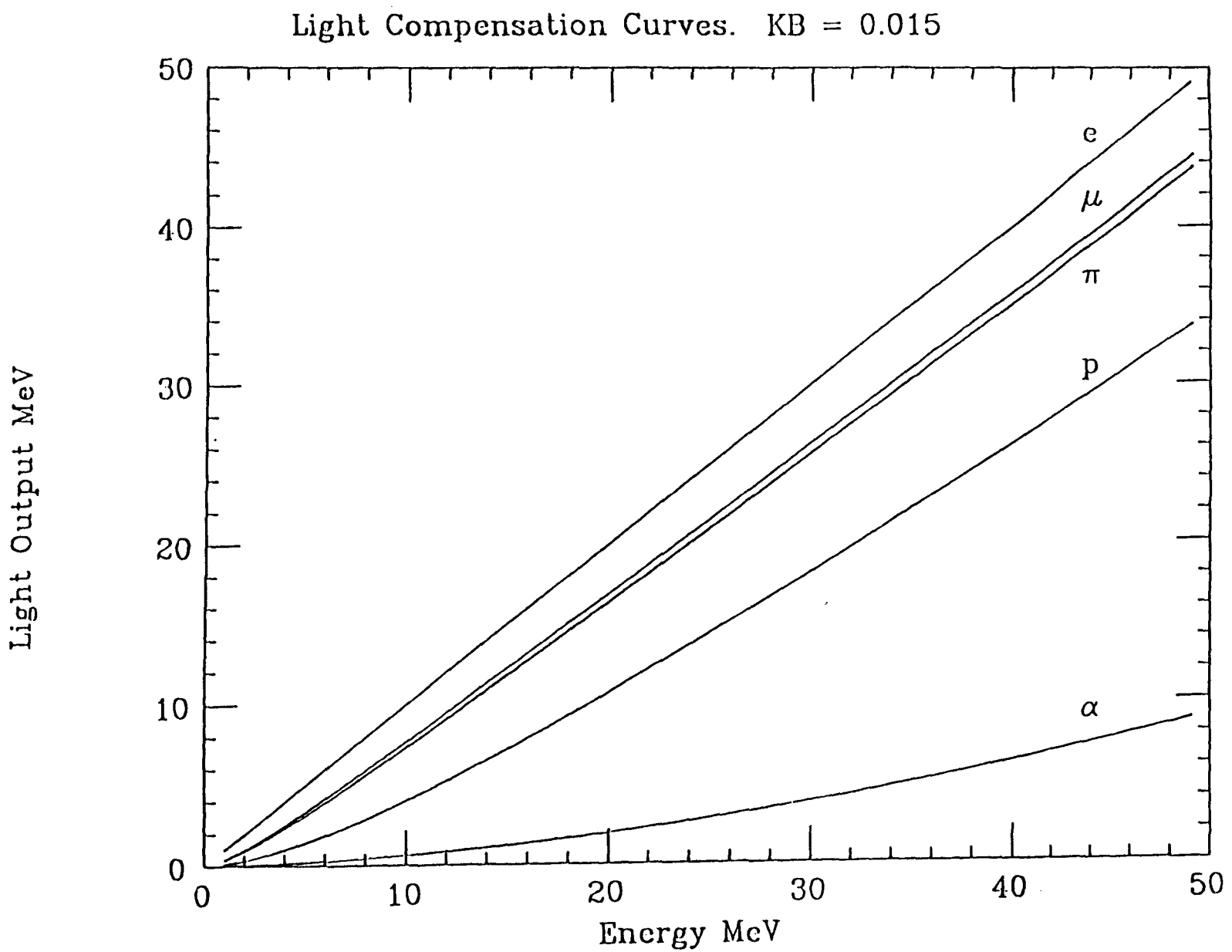
Tony Gabriel
Oak Ridge National Laboratory

Tom Handler
University of Tennessee

Brent Moore
University of Mississippi

CALOR89


HETC88
Particle Production
Hadron Transport Code
LIGHT
Scintillator
Saturation Curves
PEGS
Electromagnetic
Cross Sections
SPECT89
Hadron
Analysis
EGS4
Electromagnetic
Transport
MORSE
(MICAP)
Neutron Analysis
ANALYSIS
of Results from
SPECT89, EGS4, & MORSE(MICAP)


Light Compensation Curves. KB = 0.015
Light Output MeV
Energy MeV
0
10
20
30
40
50
e
μ
π
p
α

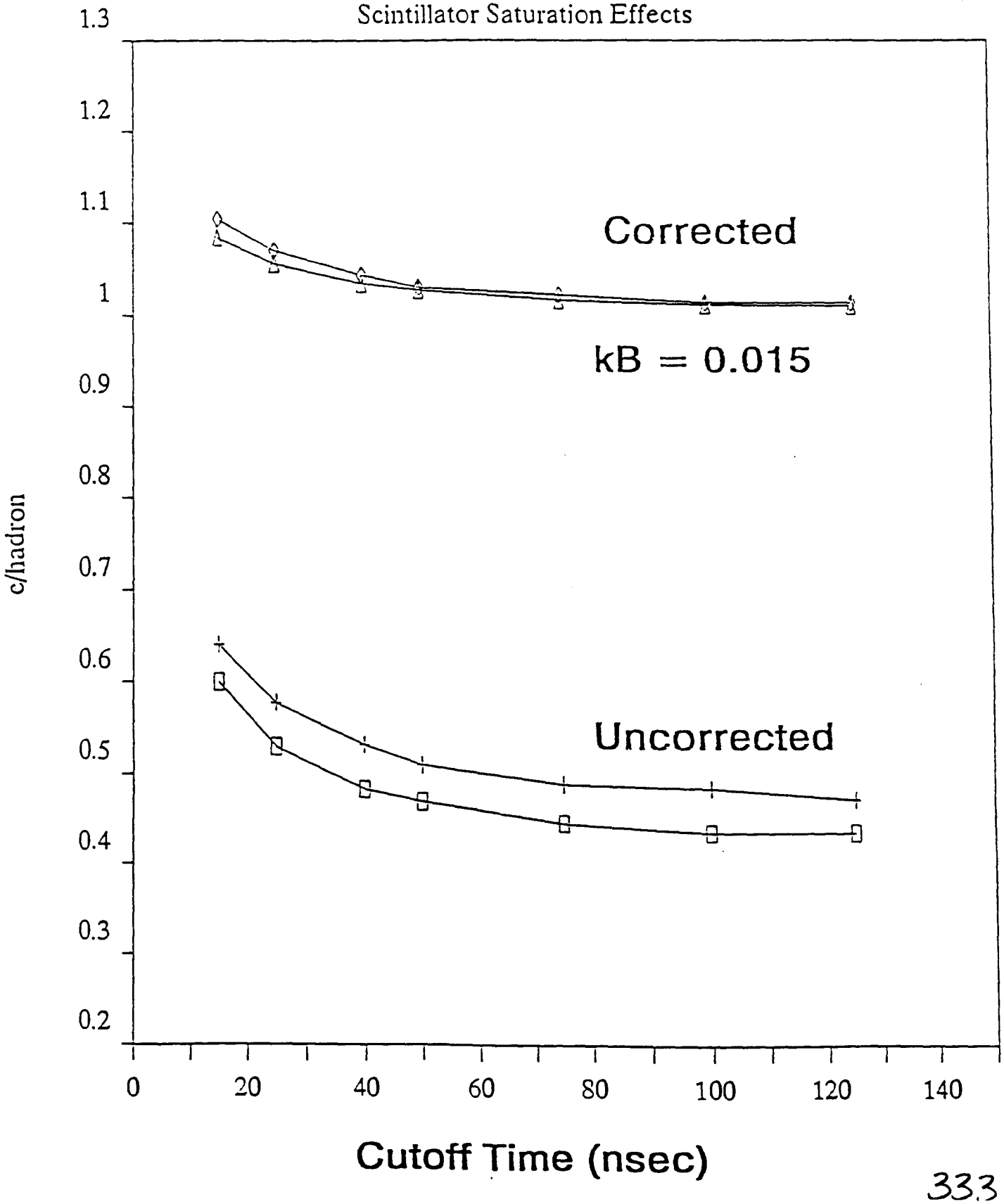
— Some of the most important things discovered thru the application of CALOR include:

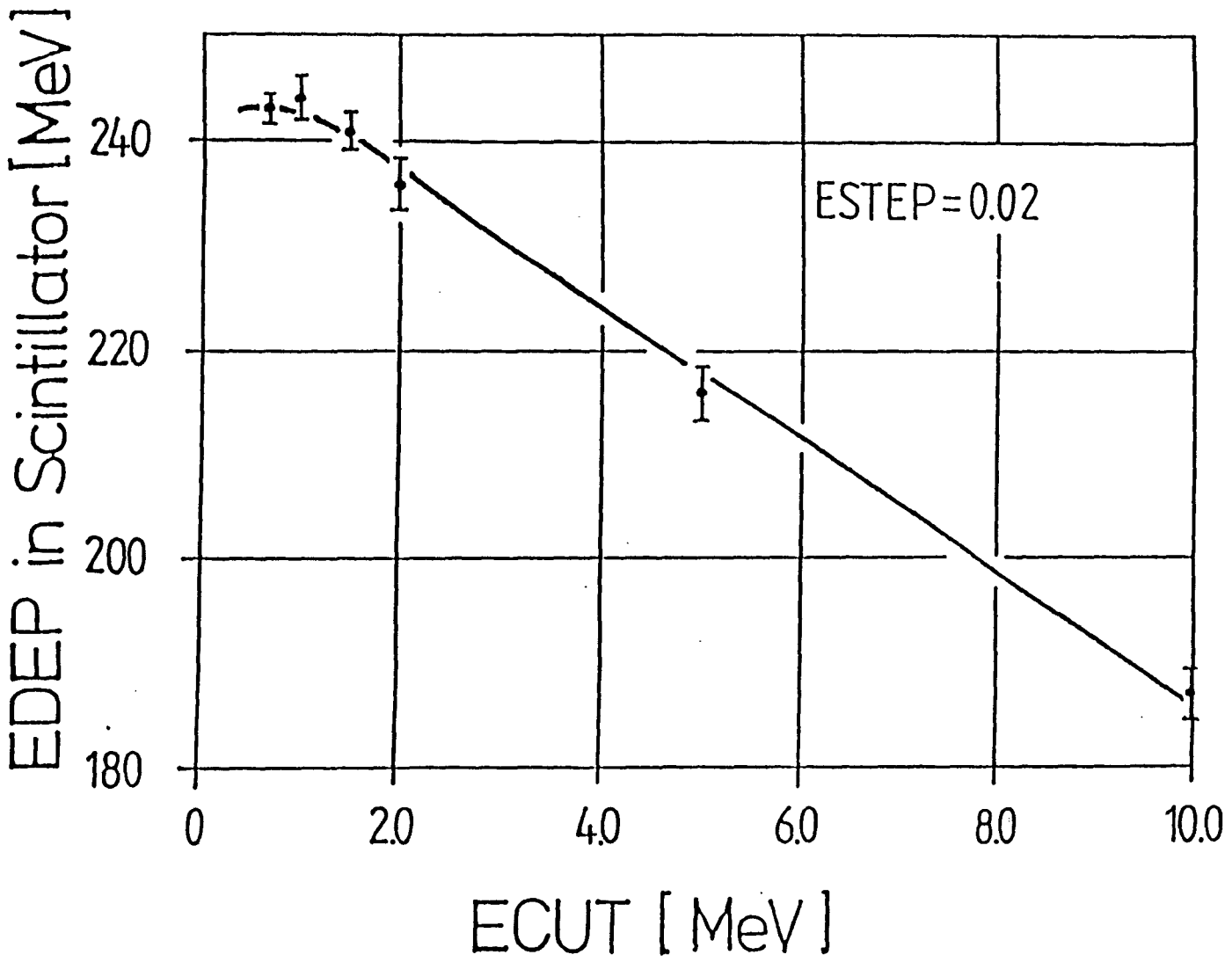
- The reason liq. Ar does not compensate
- The importance of hydrogen in the active media, especially when used high ≡ passive material
- The importance of the electromagnetic sampling itself, and the physics behind it, and
- That a Pb/plastic calorimeter should be able to compensate

— Today several subsystem collaborations are using this code system to design and analyze their calorimeter systems.

- Liq. Ar
- Scintillating Plate
- Liq. Scintillator

Different Neutron Cutoff Times




EDEP in Scintillator [MeV]
240
220
200
180
ESTEP=0.02
0
2.0
4.0
6.0
8.0
10.0
ECUT [MeV]

CAVEATS

CODE Analysis does NOT incorporate

- Non-uniformity of light collection
- Electronic noise
- Material noise - Uranium
- Pedestal cuts

These can be added during analysis

Comparison with experimental results are also complicated by simulation of full geometry and materials that are present

Tom Handler
Univ of Tenn

Tony Gabriel
ORNL

Brent Moore
Univ of Mississippi

Porting of CALOR 89

CALOR89 on Various Computing Platforms

CALOR89 is currently running on various platforms around the country.

IBM 3090 (Univ of Tenn)
Silicon Graphics 4D/240GTX (Univ of Tenn)
IBM 3084 (Argonne)
CRAY X-MP (Lawrence Livermore)
Plus others

Code has been benchmarked on all the machines on which it is running.

Transported code from IBM 3090 to Silicon Graphics 4D/240GTX.

Change from MVS JCL to UNIX

Code transported with relative ease.

- Standard changes:
  - File assignments
  - Random number routines
- Problems:
  - Common block alignments
    - Especially true for older codes
  - Local variables

Initially run in a linear mode, i.e.

HETC
→ SPECT
→ EGS
→ MORSE
→ EGS

All input files for all the programs edited BEFORE run begins as program flow is controlled by a script file.

As the 4D/240GTX has 4 processors, a new script file was written to take advantage of this. CALOR89 now runs in a pseudo parallel mode:

$\rightarrow$ SPECT.

HETC $\rightarrow$ EGS

$\rightarrow$ MORSE $\rightarrow$ EGS

Cuts processing time by 1/3.

Rapid turnaround for a given problem

This still leaves one processor free for other things.

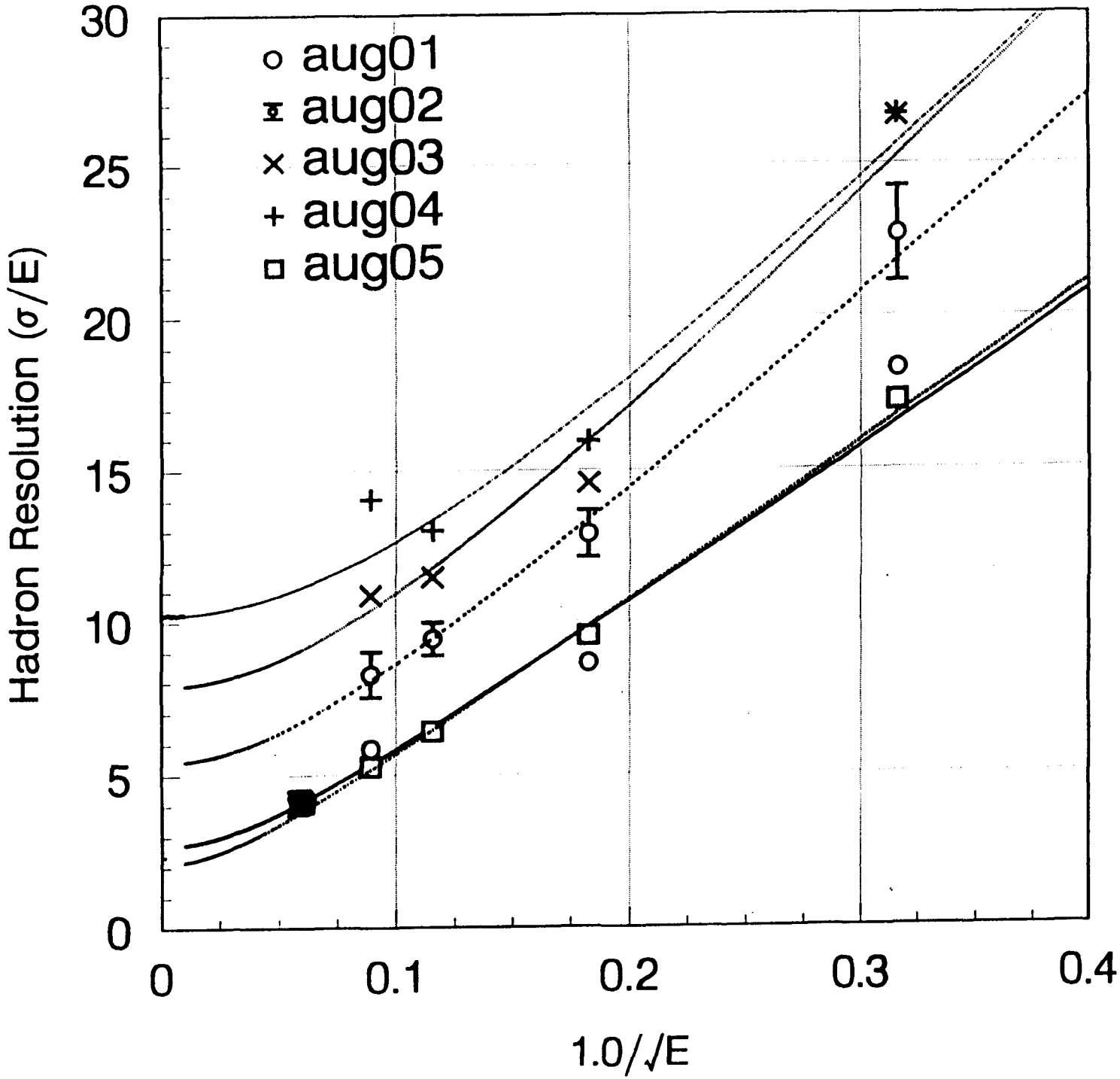
Running times on Silicon Graphics 40/240 GT

200 histories of electrons @ 125 GeV ~ 16 hrs

200 histories of hadrons @ 125 GeV ~ 8 hrs
(Parallel mode)

---

21 data points ~ 1 week running
(not total CPU Time)


aug01
aug02
aug03
aug04
aug05
Hadron Resolution (σ/E)
1.0/√E
0
5
10
15
20
25
30
0
0.1
0.2
0.3
0.4

Work is under way to transport the CALOR89 code system to the Intel Hypercube at Oak Ridge National Laboratory

Similar problems are being encountered in porting the code to this machine as was in porting from the IBM 3090 to the Silicon Graphics

EGS4 on IPSC/2 (386) Machine

Reads source info from common data set.

# Processors	Time/Proc	#hist/Proc	Eff
1	216.6 min	256	1
2	108.8 min	128	0.995
4	55.3 min	64	0.979
8	28.7 min	32	0.940
16	16.8 min	16	0.810

$$Eff = \frac{216.1}{(\#proc) * (time/process)}$$

Code is currently being set up on Intel i860 (48
with 128 nodes

Work being done by B. Kirk at ORNL

UNIX VERSION OF CALOR89

GENERAL GEOMETRY

BRENT MOORE, JIANPING ZHANG,
LUCIEN CREMALDI, JAMES REIDY
UNIV. OF MISSISSIPPI

TONY GABRIEL, FRAN ALSMILLER,
R.G. ALSMILLER, BARBARA BISHOP,
PETER FU
ORNL

TOM HANDLER
UNIV. OF TENNESSEE

LEE ROBERTS (P.K. JOB)
SSCL

YUWU YU, JERRY BUSENITZ
UNIV. OF ALABAMA

UNIX VERSION OF CALOR89

GENERAL GEOMETRY

UNIVERSITY OF MISSISSIPPI

SILICON GRAPHICS

DEC STATION 5000

UNIVERSITY OF ALABAMA

APOLLO DN10000

PDSF AT SSCL

SILICON GRAPHICS

SUN SPARC2

PDSF Implementation

Compiled Executables:

- HETC
- PEGS
- EGSPREP
- LIGHT

User Compiled:

- SPECT
- MORSE
- EGS4
- FINAL

active media > 1

BENCHMARKS

SCINTILLATING PLATE

IRON/PLASTIC SCINTILLATOR

10 GeV URANIUM/PLASTIC SCINTILLATOR

LEAD/PLASTIC SCINTILLATOR

HANGING FILE CALORIMETER

LEAD/IRON/PLASTIC SCINTILLATOR

20, 100, 225 GeV

PDSF TIME SCALE:

1-4 WEEKS

FUTURE

COMBINATORIAL GEOMETRY

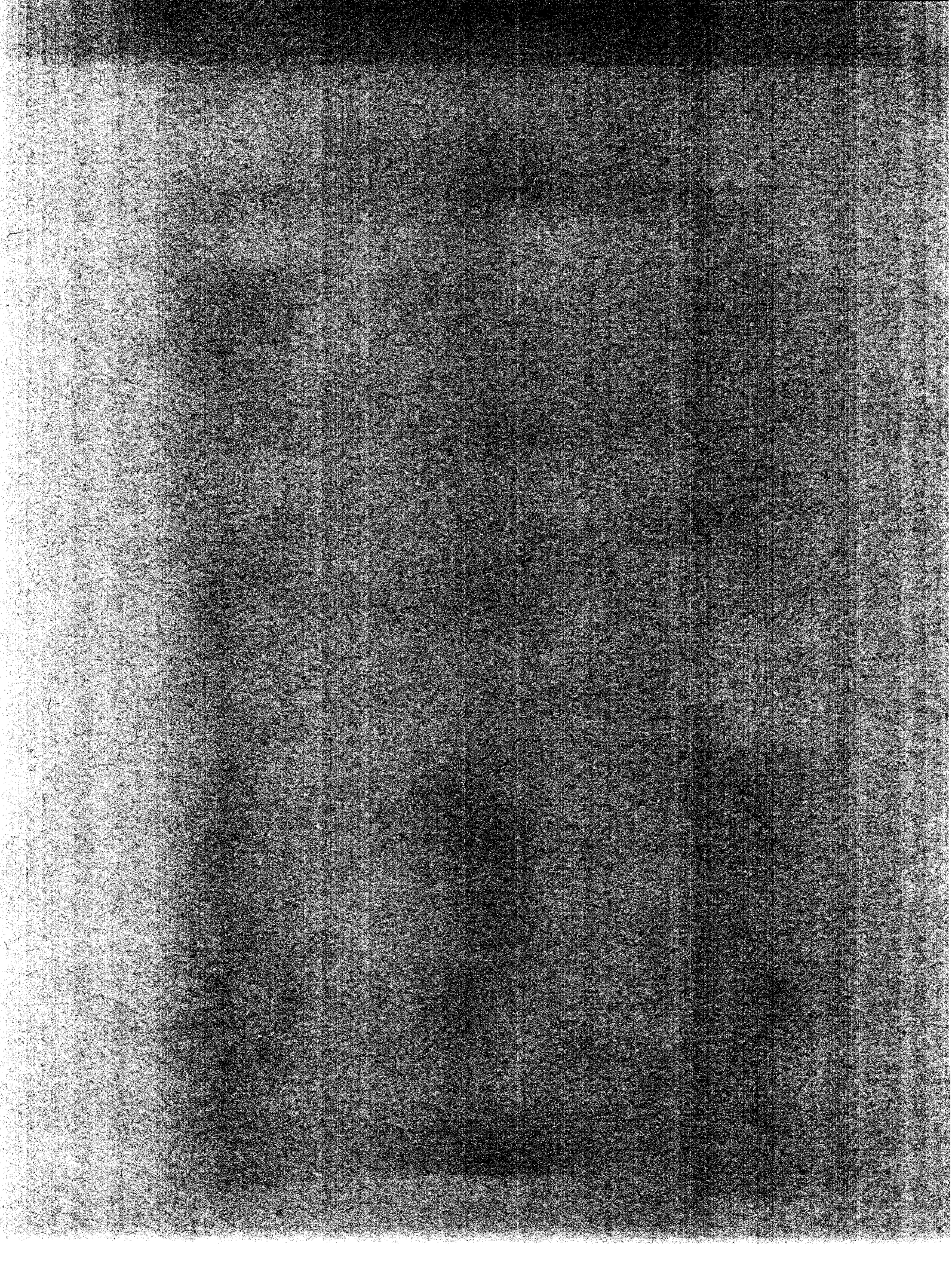
"STREAMLINED" NO-COMPILE VERSION

ENERGY DEPOSITION AS A FUNCTION OF DEPTH, RADIUS, AND TIME GATE SET IN INPUT FILES.

CALOR-GEANT INTEGRATION

"Physics Analysis Tools on UNIX"
"RISC Workstations"

F .Rouse



Physics Analysis Tools for RISC Workstations

Outline

I) Introduction

II) Objects

III) Tools

IV) Communications System

V) Current Status

VI) Conclusions

Physics Analysis Tools

Current Problems of HEP analysis code

A) Redundant code

Trackers, Histogram packages, fitting routines,

analysis environments for example.

B) Available computer resources

University groups have a hard time competing with laboratories

C) Documentation

CERNLIB (PAW, GEANT etc) is well documented...

anything else?

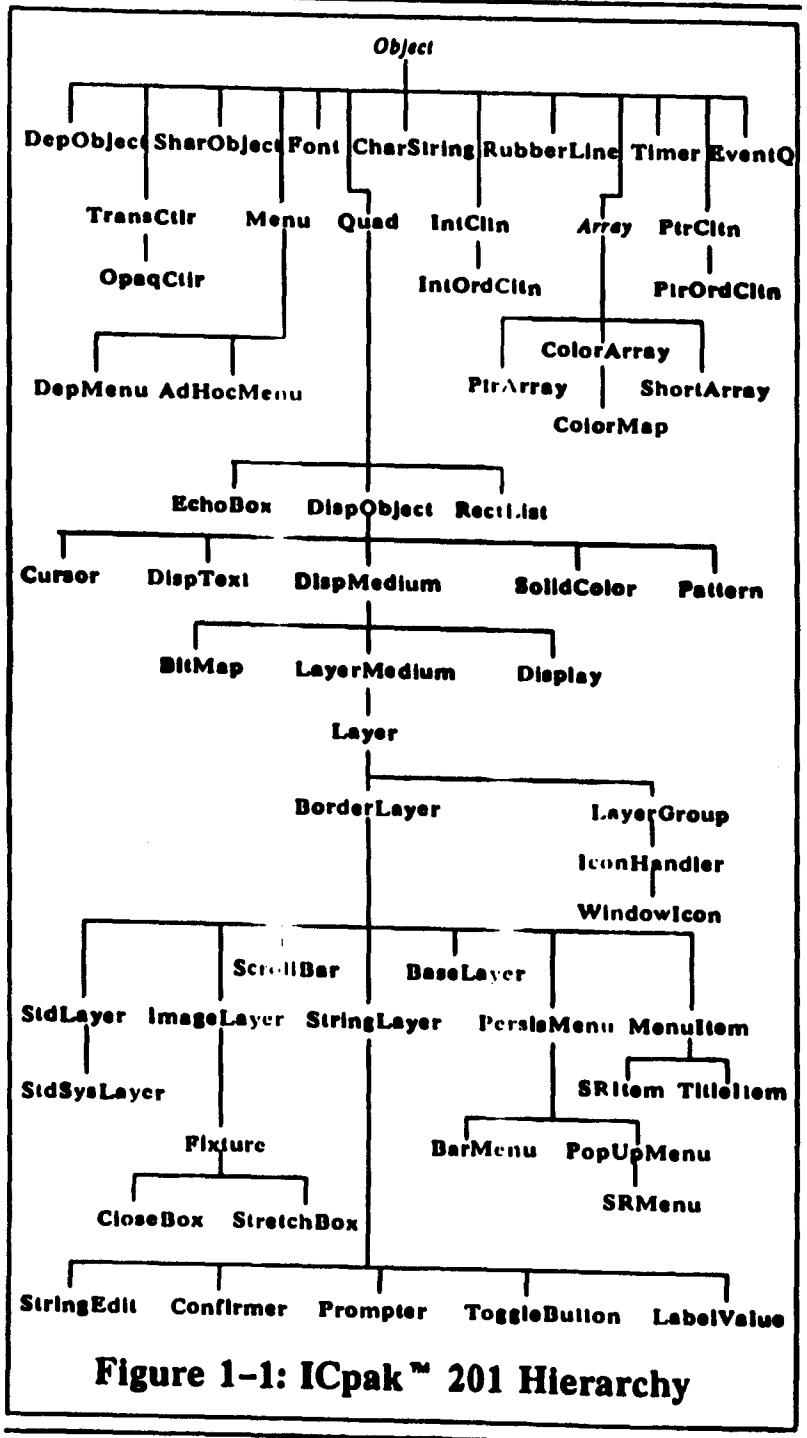


Figure 1-1: ICpak™ 201 Hierarchy

(c) Stepstone 1988

Class Inheritance Hierarchy

For ease of reference, class specification sheets are arranged in this manual in alphabetical order. However, it is recommended that the user initially review the classes in order of inheritance as illustrated in Figure 1-1.

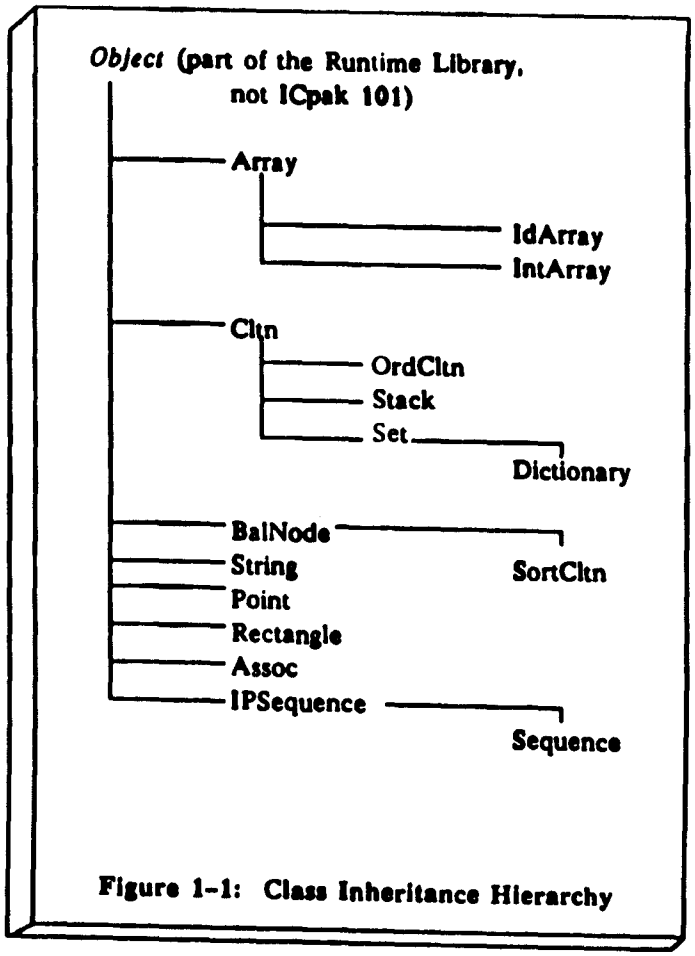


Figure 1-1: Class Inheritance Hierarchy

(c) Stepstone 1988

Physics Analysis Tools

What is a Tool?

A tool is an object that has specific inputs of information and produces specific output. The tool can be a member of a specific user written subclass of the tool class. A special class handles user written subroutines (or existing packages). Here, a tool is specified by an external address and the calling sequence (at first, only FORTRAN compatible).

This Defines the Interface [redacted]
of Tools (read subroutines/modules/packages
programs/...)
to one another.

Advantages: → Uniformity of interface
allows uniformity
of functions
(documentation, testing, distribu
,...)

→ Interconnect different
languages

→ Build Systems out of primitive

→ Distributed processing

→ A visual program self docum

Physics Analysis Tools

The structure of the Tool:

Many Methods that control the appearance of the tool. Most users will not care about these method.

Methods that control the one time initialization, the per run initialization, the execution of the tool, the per run exit and what happens when the tool is deleted.

Methods that control special display (histograms for instance would logically put up the histogram in this method) and status of tool.

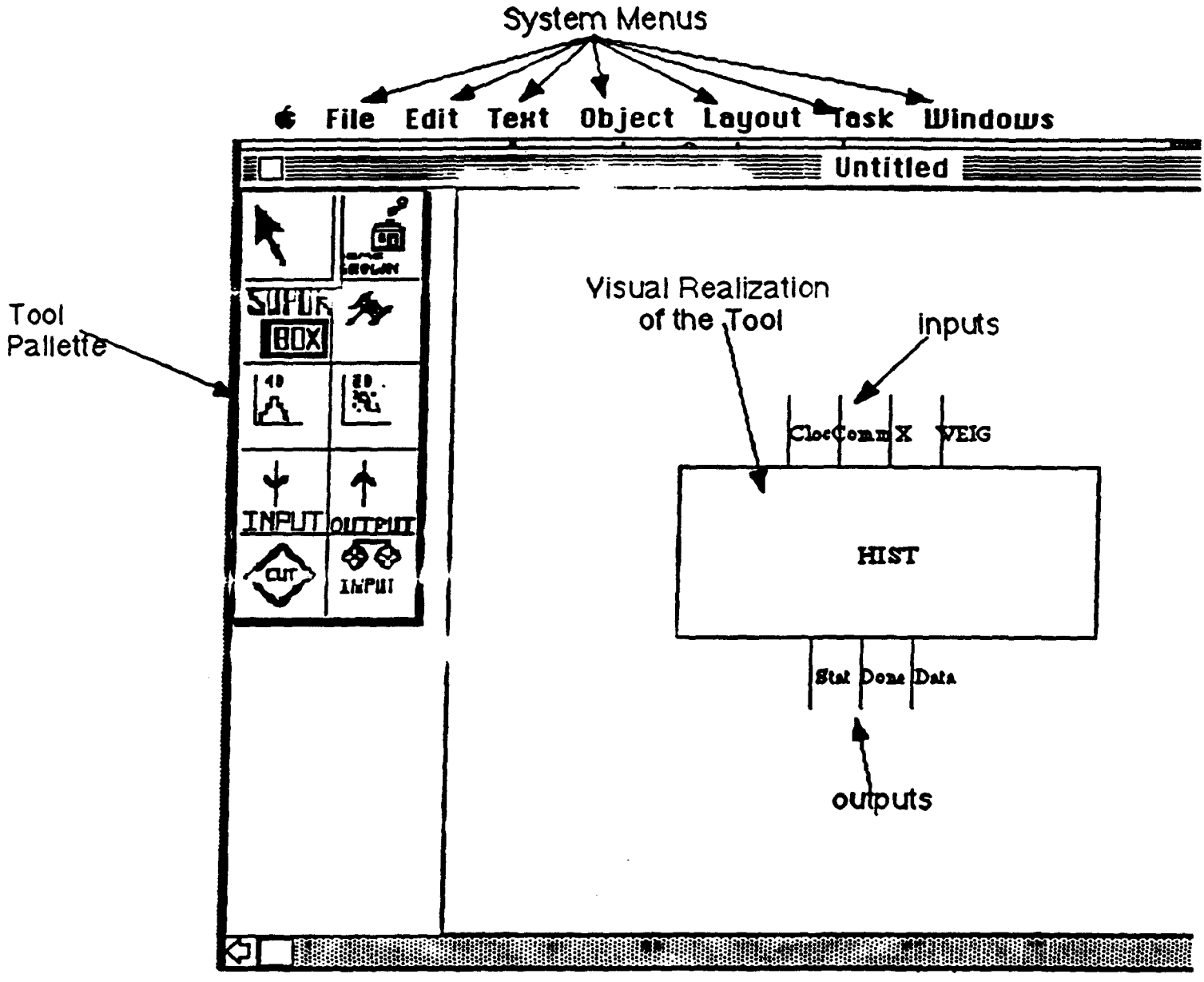
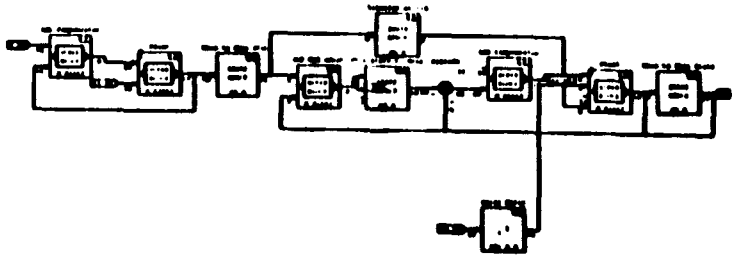


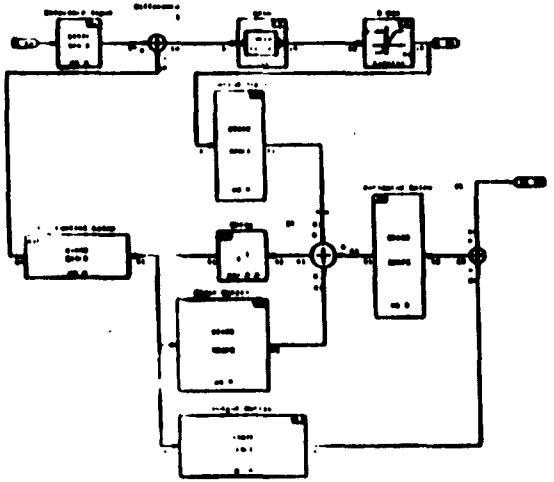
Figure 1: An example of the proposed Graphical User Interface. This example is from the running prototype on the Macintosh™ computer.

358

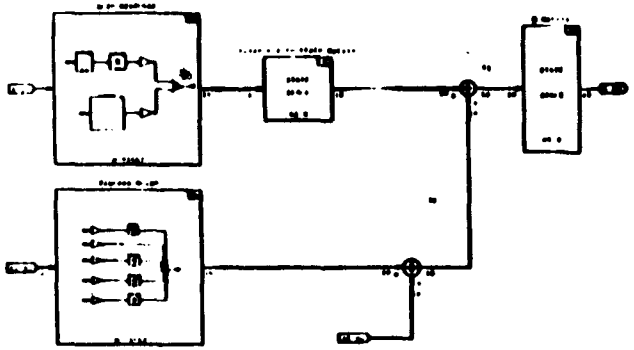
Top level diagram for adaptive controller



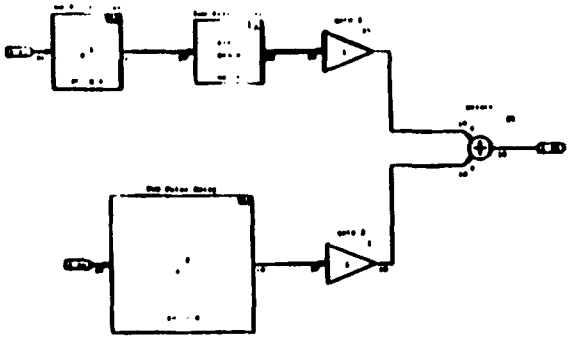
LQG Compensator



Plant



Actu_response



With Thanks to
Tom Himel

Physics Analysis Tools

Since the tool itself is a drawable object, it can be manipulated by the graphics system.

On workstations, the class library makes drawable objects appear under X windows. If one uses the same class library on another graphics system (SUNVIEW, DECWindows, etc), the code will work the same as parent class objects interact with the graphics system.

The tool system is self documenting: The flow chart is the program. Additionally, when a tool is defined in the system, help will be requested.

Tools that have similar function can be grouped together onto the same palette. Envision a statistical palette, a graphics palette, a histogram palette, etc.

Physics Analysis Tools

Communication system binds tools together

Basically, when a tool has all of it's inputs, it fires.

Lines between tools define dataflow.

Details to be decided like:

- **Pass by address only?**
- **In limited space, how to update invalid communication objects?**

3761

Physics Analysis Tools

This naturally lends itself to Distributed Processing:

Connections between objects are virtual. Therefore, information could cross between processors without the user having to worry about it.

Proviso: All information that an individual tool needs is either static or is provided as input to the tool. Ie global information like common blocks presents a problem. This however, is application specific and the individual user can take care of this.

Under UNIX there is a host of inter-task, and inter-process communication tools such as streams.

Physics Analysis Tools

Status:

Limited Prototype already working on Macintosh. (All pictures of tools comes from this prototype).

Prototype can obviously draw tools, it can move them around on the screen. The palette can be moved. The basic underlying task to execute tools has been written (trivial).

Major problem is that since Macintosh not a true multi-tasking machine, communications system was not implemented.

Physics Analysis Tools

Only major coding will be the communication system!

Expectation is that a working Version 0.1 system should be running by end of year. Version 0.1 probably will not have all of the support necessary for grouping arbitrary tools together into user defined palettes. Also, limited language support.

Computer Science department interest in working on Distributed processing problem.

Physics Analysis Tools

Conclusion:

A physics analysis environment based on iconic tools is being written for RISC workstations. This environment will allow:

1) Wiring of analysis primitives together to form a program.

2) Ability to group primitives together to form a new tool.

3) External code in any language compiled by machine although calling sequence will initially just be FORTRAN.

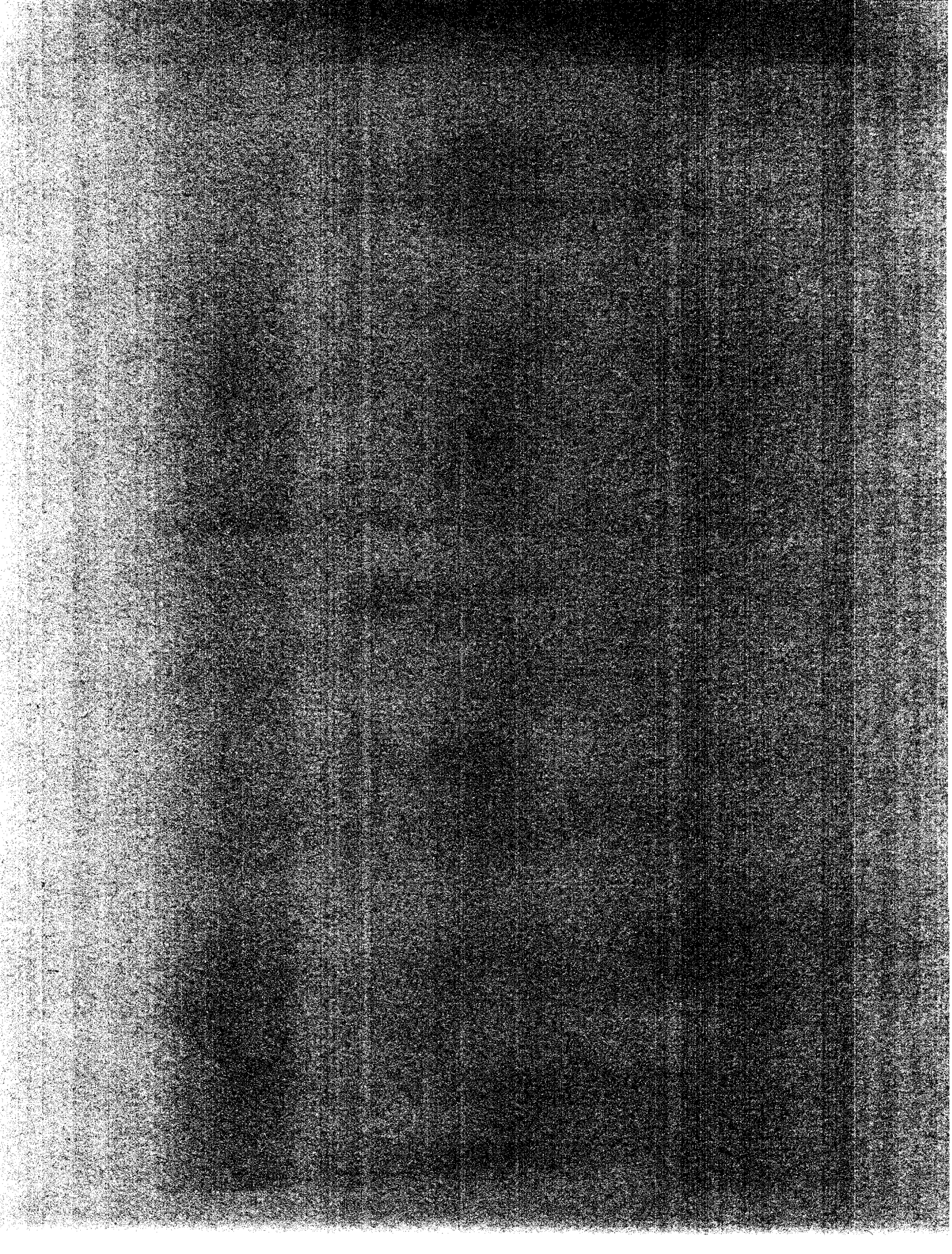
4) Distributed Processing

5) Self Documenting along with many automatic help facilities

365

"Fortran 90 Evaluation and Experience/Status Report"

M. Marquina
M. Metcalf



M. Metcalf

A First Encounter with Fortran 90

Recent events:

22 March: WG5 charges X3J3 to finalise standard

1 April: X3J3 finalises standard

early June: NAG asks CERN to help β-test its compiler

10 June: NAG announces world's first full f90 compiler

early August: ISO publishes standard, replacing FORTRAN 77

10 September: NAG ships Release 1 to customers

CERN's objectives in β-test:

- familiarization with the compiler and the language
- evaluation of the compiler
- assesment of conversion effort if f90 route adopted in the future
- stress testing

NAG Compiler

- A full implementation of the standard,
  using C as an intermediate language.
- Written in portable C, runs on unix platforms
  (VMS coming soon).
- Simple to use:

```
$ f90 name          new source form
or
$ f90 name.f        old source form
$ a.out
```

- Six passes – the C passes can be replaced by vendors with native code generation.
- Compilation speeds with -O 5%-33% slower than ftn.
- Execution speeds not yet measured, but rumoured to be only somewhat slower for numerical code.
- Excellent diagnostics.

Familiarization

Use a 1500-line source form conversion program:

f77 version — now works

f90 version (using MODULE, CASE, INTERFACE, DO...END DO, EXIT, CYCLE, array constructor, etc — now works

Evaluation

Use KERNUM, KERNSEN, GENLIB, ZEBRA, FATHEN, GROPAL:

stage 1: syntax checking — now works

stage 2: code generation - now works (one outstanding OS error in FATHEN)

stage 3: execution testing and timing — now underway starting with KERNUM

Code conversion

1) INTEGER*2 → INTEGER(2) , etc.
(done automatically by CONVERT)

2) Inconsistent use of, for example, UCOPY:

```
CALL UCOPY(I, J, N1)      →    I(:N1) = J(:N1)
CALL UCOPY(A, B, N2)           A(:N2) = B(:N2)
```

or can write a generic subroutine and add, e.g.,
USE GENERIC_UCOPY
to code.

3) No Hollerith in DATA statements, so use CHARACTER,

```
DATA I/4HABCD/          INTEGER:: I=TRANSFER('ABCD',
:                   →   :
WRITE(*,'(A4)') I       WRITE(*,'(A4)') TRANSFER(I, '1234
```

4) EXTERNAL fname → REAL, EXTERNAL :: fname
└ or appropriate type
(not required by standard, but by NAS's use of C).

5) Hexadecimal constants; e.g. Zabcdefff → Z'abcdefff'

... and that's all!

Stress testing

Test the compiler to its limits by giving it large amounts of code in large units (especially as it makes global checks).

KERNNUM

- 1 change required (to EXTERNAL statement).
- 12 warnings of use of obsolescent ASSIGN and arithmetic
- compile time on Apollo 9000 : f90 -O 337 s
  (11200 lines) ftn 251 s

Execution tests proceeding.

GENLIB

- some use of Hollerith
- ~5 trivial changes required to poor code

KERNGEN

- contains obsolete entries which would require effort t
  convert:
  many instances of • Hollerith (suppressed)
  • hex (converted)
- lots of warnings about unused symbols
- some CHARACTER/INTEGER equivalences
- lots of obsolescent arithmetic IFs

⟹ recommend some clean-up before

To be continued ...

They say the first Fortran 90 Compiler won't be available until at least 1992.



We say they're wrong.

NAG is pleased to announce the release of the worlds first Fortran 90 Compiler — one of several new exciting products to be released in the NAGWare Service.

Fortran is still one of the most widely used scientific and engineering programming languages and the f90 Compiler will ensure it stays so.

The NAGWare f90 Compiler is a full implementation of the ISO Fortran 90 Standard and is ready to meet the most demanding test of all — you.

The future is in the present.

NAGWARE™

For further information about NAGWare and the other products and services available from NAG, please contact one of the following:

NAG Ltd, Wilkinson House, Jordan Hill Road, OXFORD, OX2 8DR, UK. Tel: +44 865 511245 Fax: +44 865 310139 Telex: 83354 NAG UK G

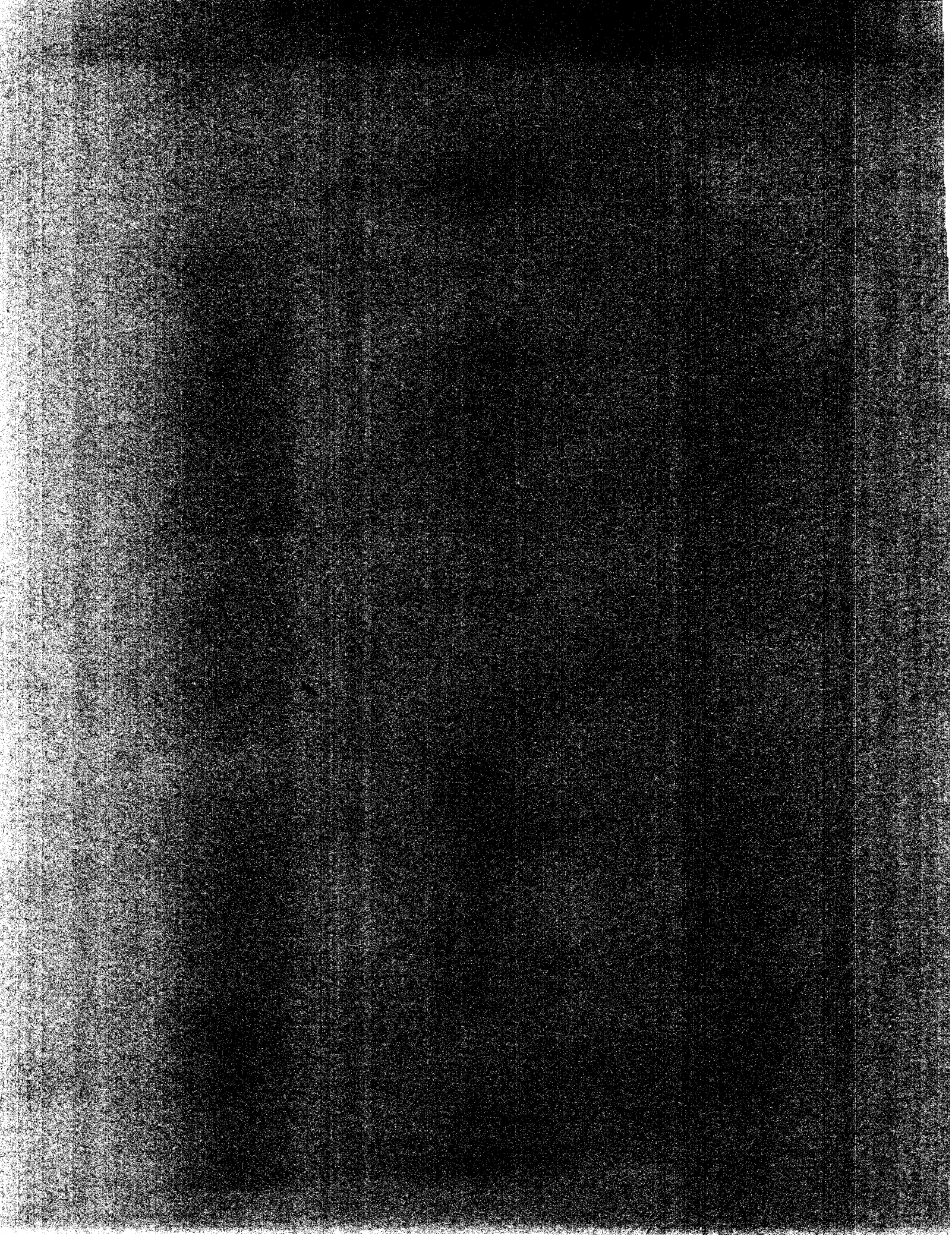
NAG Inc, 1400 Opus Place, Suite 200, Downers Grove, IL 60515—5702, USA. Tel: +1 708 971 2337 Fax: +1 708 971 2706

NAG GmbH, Schleißheimerstr. 5, D—8046 Garching bei München, Deutschland. Tel: +49 89 3207395 Fax: +49 89 3207396

Circle 83 on reply card

"D0 Software Library Organization and Maintenance"

S. Protopopescu



Serban Protopopescu
HEPLIB '91
Sept. 20, 1991

THE DO SOFTWARE LIBRARY

I. Goals and Requirements

II. Organization

III. DOCMS

IV. DOLIBRARY

V. Program Builder

I. Goals and Requirements

50 physicists distributed across 12 institutions contribute to the D0 software library.

New code must be made available to any member of D0. Changes have to propagate quickly.

It is crucial that programmers developing code be working with the same versions of the already developed code.

A central facility and systematic procedures have been set up to coordinate this effort and make the code available to the whole collaboration.

II. Organization

All code development for DO is done on DEC VAX computers.

VAX's provide:

1) good environment for orderly file manipulation

2) good debugging facilities

3) powerful systems for managing large amounts of code:
   a) CMS (Code Management System)
   b) MMS (Module Management System)

4) good communication links between VAX's via DECNET (sometimes)

DECNET makes it feasible to update sources at one central location and then distribute them to all other VAX's within hours.

The D0 software library consists of two major directories:

D0CMS is the repository of CMS sources with a history of all the changes. There is only one copy of this directory which resides at the FNALD0 VAX cluster.
It contains close to 20,000 modules.

D0LIBRARY contains all the fortran sources,object libraries, documentation, etc. Identical copies of this directory are kept on all VAX's in the collaboration.

Each of the main directories are logically subdivided into subdirectories which groups modules according to their function.

The files in D0LIBRARY are distributed from files in D0CMS using release procedures that rely on MMS.

Logical symbols are defined to make user environment identical on any VAX.

III. DOCMS

Subdirectories for each major utility library
and applications program.

A CMS subdirectory contains elements bundled in groups.
One element = one module = one subroutine (by decree)
Any element can be in any number of groups.

Element name is unique.
Module name is the same as the subroutine name.
If more than one version of a subroutine is
needed a prefix is added to the module name.

Documentation is also maintained with CMS.
Applications programs and some utility libraries
have manuals in their respective subdirectory.

We refer to each subdirectory in DOCMS as a library.
Each library has a czar responsible for its state.
At present there exist >40 libraries.
Some are very small (a few tens of modules)
some have thousands of modules.

Procedures for upgrading DOCMS

Basically an authorized programmer has to follow following steps:

1) Log in at FNALDO under an account in his name via DECNET.

2) Use CMS to reserve the file he wishes to modify out of [DOCMS.XXX]

3) Copy the file to his home VAX

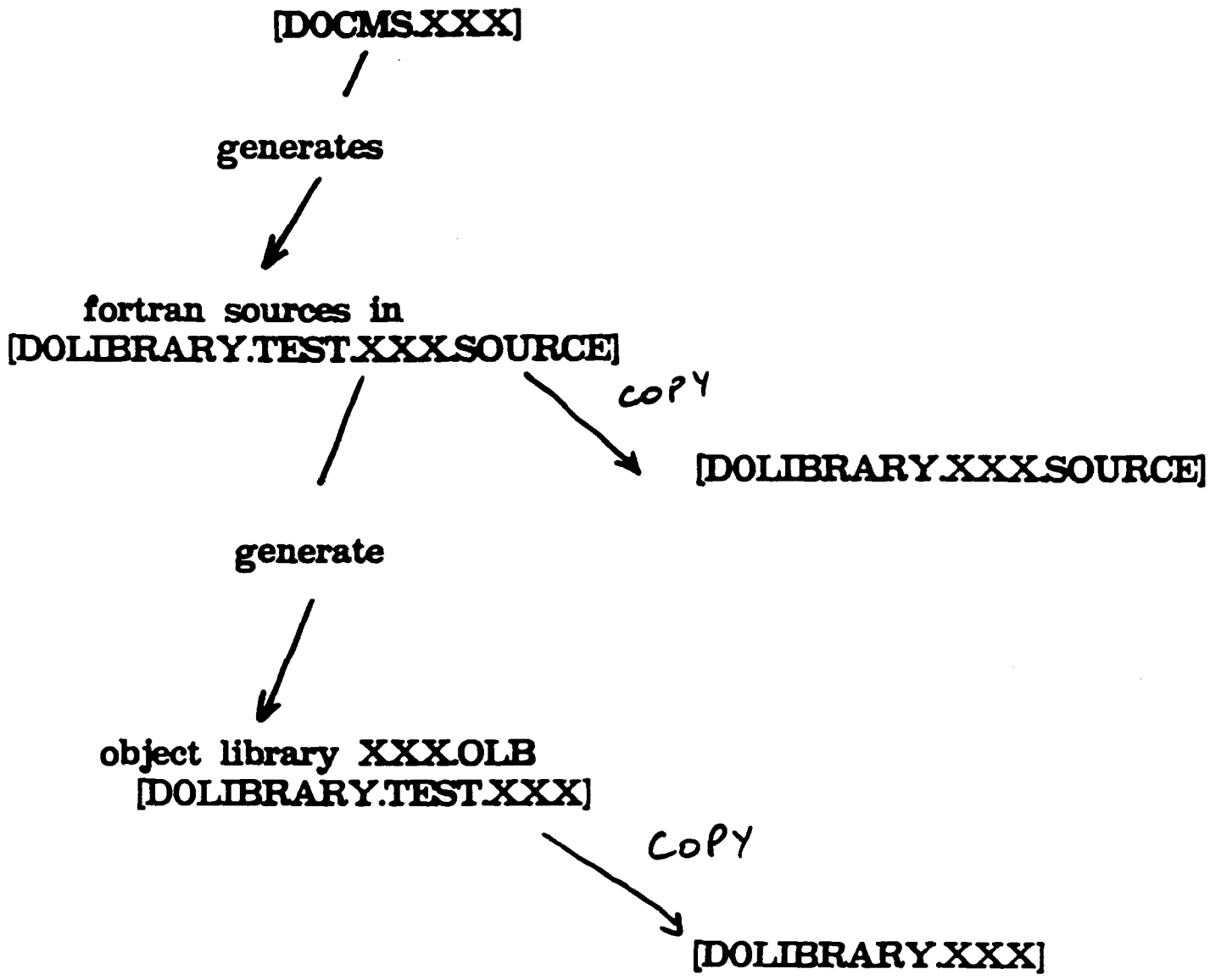
4) Modify and test his modifications

5) Copy modified version to his area at FNALDO

6) Use CMS to replace old version

CMS automatically keeps a record of all changes. Older versions of a modified file can be recreated if needed.

For a simple utility or self contained library subdirectories are connected via MMS procedures in the following way:



A release only ships changed modules, not the full library. Every library XXX has a czar responsible for its state.

The complete cycle in from original code development to full release is more complicated if more than one person is working on code for a program that involves modifying modules accross a number of libraries:

A project local CMS library is created on a designated VAX cluster at FNAL.

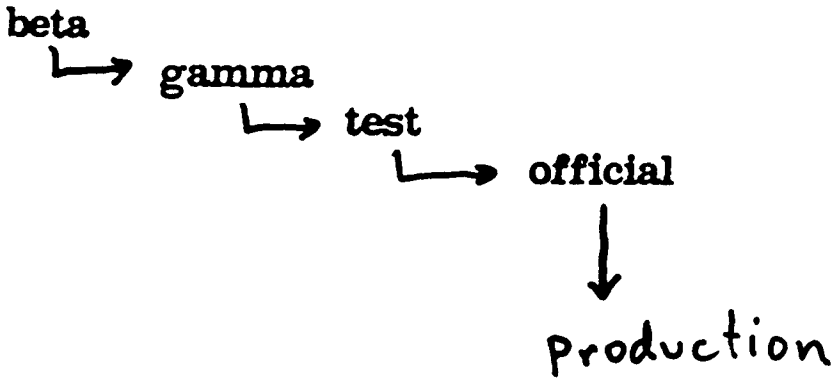
Modules that need modifying are reserved from DOCMS and inserted into the local CMS library.

Programmers update the local CMS library.

Periodically beta releases are made from the local CMS library to the VAX development cluster.

When ready the entire beta release is shipped to DOCMS and the local CMS library is emptied.

Before test releases are requested the czars perform gamma releases to their own node to test that the libraries were properly updated.



TOOLS FOR CODE DEVELOPMENT

EVEDT

D0 editor, provides templates for headers, Zebra bank documentation,etc. Finds entry points in library, has language sensitive features.

D0FLAVOR

Checks that offline standards are satisfied, handles machine dependent features.

D0ENTRY

Find modules in library.

DBANK

Combines Zebra bank documentation with debugger.

ZBANK

Generates templates for all subroutines that must be created for a ZEBRA bank.

CMSINTERFACE

Checks for code standards and module name uniqueness before insertion into CMS library.

IV. DOLIBRARY

Any VAX in the DO collaboration can have copies of subdirectories of the DOLIBRARY by request.

Every subdirectory has a subscription list.

VAX's on subscription list have their libraries automatically updated from FNALDO

Procedures can be guaranteed only if directories are NOT modified locally. Only changes are shipped for new releases.

Command procedure DOLOCAL.COM (in all DO VAX's) assigns logical names to local disks and logical names to directories and procedures. Creates identical working environment for all users.

Examples:

local$disk:[DOLIBRARY.CALOR_UTIL.SOURCE]



defined as D0$CALOR_UTIL$SOURCE

local$disk:[DOLIBRARY.CERNLIB]



defined as D0$CERNLIB

LIBTEST redefines logicals, i.e:

$ LIBTEST CERNLIB

changes D0$CERNLIB so it points to
local$disk:[DOLIBRARY.TEST.CERNLIB]

$ NOLIBTEST CERNLIB

sets D0$CERNLIB back to original value.

Library upgrades use an
MMS (module management system) procedure

For each library or program there is one and only one person responsible for coordinating upgrades. An upgrade will occur only at his request.

The procedure:

1) checks which modules have changed

2) fetches them out of the CMS library

3) copies them to the [DOLIBRARY.XXX.SOURCE] subdirectory automatically via DECNET in all the VAX's on a subscription list.

4) Checks for any dependencies

5) deletes old versions, compiles new versions and inserts them in the object libraries in [DOLIBRARY.XXX]

6) Runs XXX.RELEASE, set of instructions specific to library XXX. This may request some special activity such as generating an EXE or constructing some data file by running special programs, etc.

V. PROGRAM BUILDER

A tool for assembling programs
and enforcing discipline.

Programs consist of a frame, utilities
and applications. Frame and applications
may call utilities but not the reverse.

An application is inserted in a frame via
a set of interface subroutines (package).

Applications communicate with other applications
only through Zebra banks or a very limited set
of subroutine calls.

Requirements for use:

1) A framework program

2) Library and/or user packages

3) A Framework description

4) Package descriptions

5) a list of library packages

6) a link file template (.LNK)

7) a setup command file for the framework

FRAMEWORK PROGRAM:

A program calling a set of hooks at well defined stages. The hooks are logical functions with no arguments. The framework may use a number of utilities to handle I/O and control program flow.

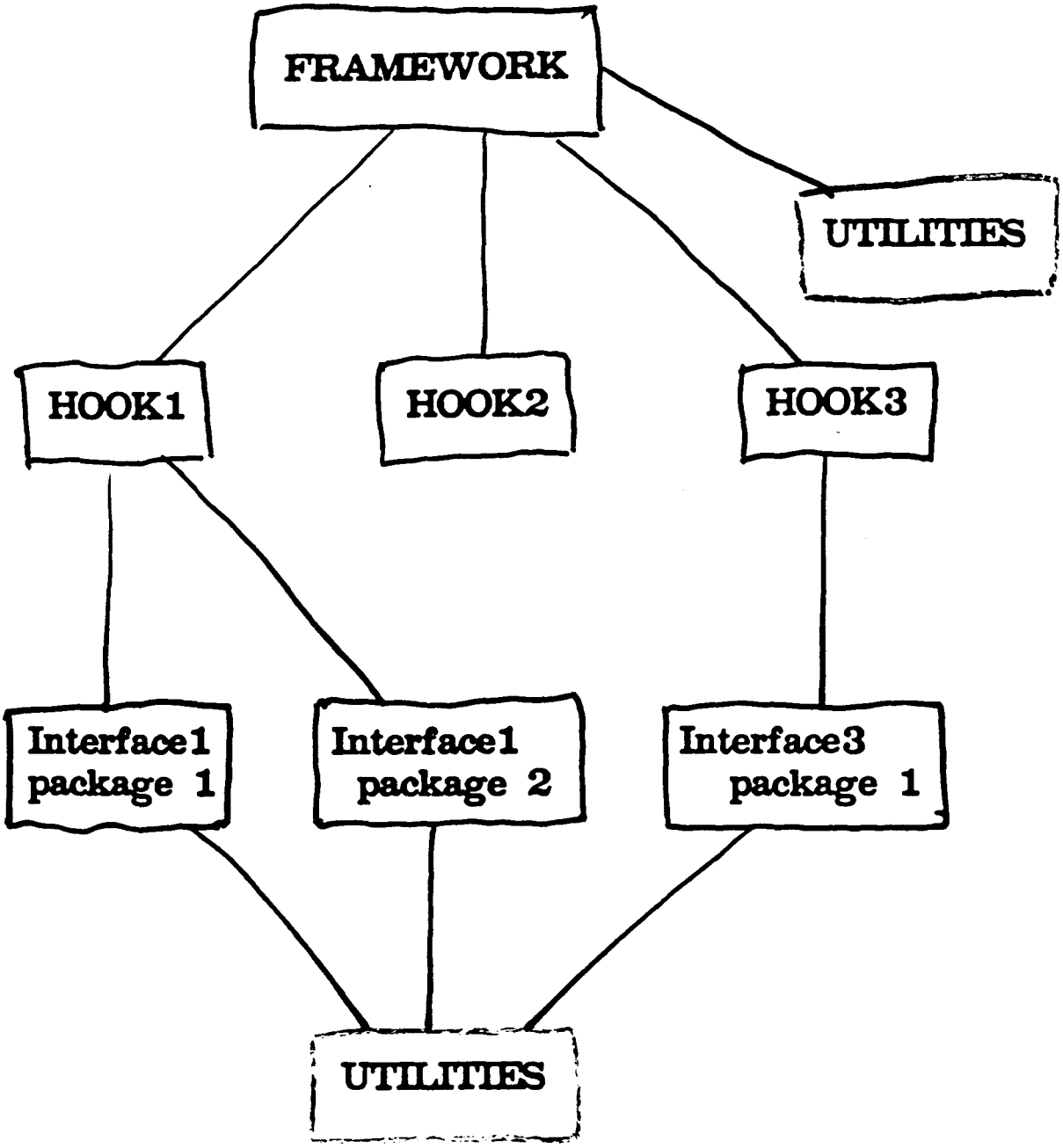
PACKAGES:

A set of subroutines and control data files performing a well defined task. The top level subroutines (interfaces) must be logical functions with no arguments. The control data files must use the SRCP (static run control parameters) facility.

To fit into a framework the package interfaces must match the framework hooks but it is not necessary to supply interfaces for every hook.

Any number of packages can be combined in one frame. Packages are easily transportable from one framework to another.

Packages communicate with other packages only via Zebra banks or utility subroutines that provide access to Zebra banks.


FRAMEWORK
UTILITIES
HOOK1
HOOK2
HOOK3
Interface1 package 1
Interface1 package 2
Interface3 package 1
UTILITIES

PRODUCTION RELEASE

A production release is a release of a program to be used in large scale production, i.e.:

Montecarlo event generation

Offline event reconstruction

Level 2 filtering

Production programs are build by the program builder and special production release procedures.

All needed files are released to production directories. This include:

EXE, object libraries, sources and data files (not event data)

Production releases are still being developed, in particular releases to UNIX platforms.

EXPERIENCES WITH DO SOFTWARE DISTRIBUTION

Overall library distribution works reasonably well.

It requires 2 full time people:
One to handle distribution requests and check status of libraries at all nodes,
another to improve and maintain the procedures.

Biggest problem is unreliability of the network, it is rare that any time all clusters in the collaboration are accessible.

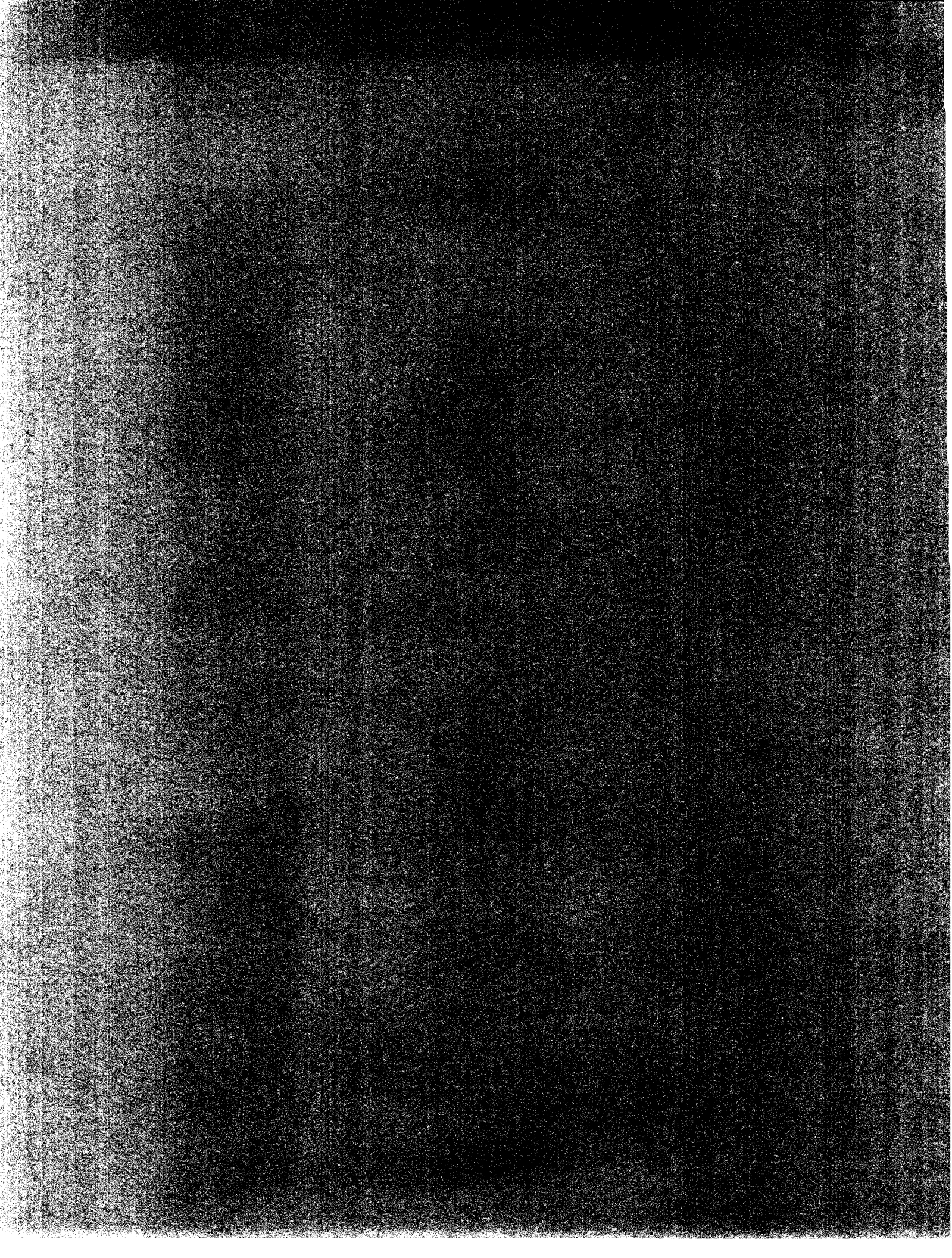
There are automatic recovery procedures, however test releases can only be made when all nodes have same official versions.

If for a given release there are many dependencies between separate libraries the status of the library at a given node may be unstable until all releases are finished.

Release procedures are more complex than anticipated, the complexity comes from attempts to minimize the periods of instability.

"CPS Plans and Evolution"
"UDP/UPS at FNAL"

P. LeBrun
R. Pordes



WHO AM I?

~~Margaret Votava~~ — RUTH PORDES
Online Support Department
Fermi National Accelerator Laboratory

WHY AM I HERE?

UPS

Methodology for

- packaging
- maintenance
- distribution

software products in a multi-platform Unix environment

WHO CARES?

Lab wide interest - many departments have input at various levels:

- Access Liaison
- CDF
- Distributed Computing
- Online Support
- Physics Analysis Tools

Even more are using it!

- E687, E791, E706, E800, D0, E773 and their colloborators
- Offline Farm Machines
- Division Office

Facets of Unix Product Support Methodology:

- Central Computing Facility => Easy access to many different software packages, *by many inexperienced users. Allow for "new" , "test" etc, simple interface*

  => Suitable for packaging of vendor software, *such that software can be installed as per vendors instructions, (so they give support) , but on-site supporters and distributors can track distribution, use and updates.*

- Experiment Online Systems => Easy switching between versions, *so new version can be installed and removed without impacting stability of data taking software.*

  => Node specific installation in a cluster environment, *generally not handled by vendor installation procedures.*

  => methodology for defining and executing boot and installation time procedures, *for device driver code etc.*

  => support for collection of packages *into data acquisition and online systems, allow for dependencies between products, invoking one product from another etc.*

- Experiment Working Groups => Collections of software packages *on a single computer used by defined, independent sets of people.*

  => Support for remote sites , *central distribution, management and expertise*

  => No cost, *needs to be affordable by all collaborating institutions*

- Product Maintainers => Access to lists of available, and installed software

  => Common archival and retrieval procedures

  => common methods for installation of updates

- Software Developers => Definition of source area - *independent of all versions of software.*

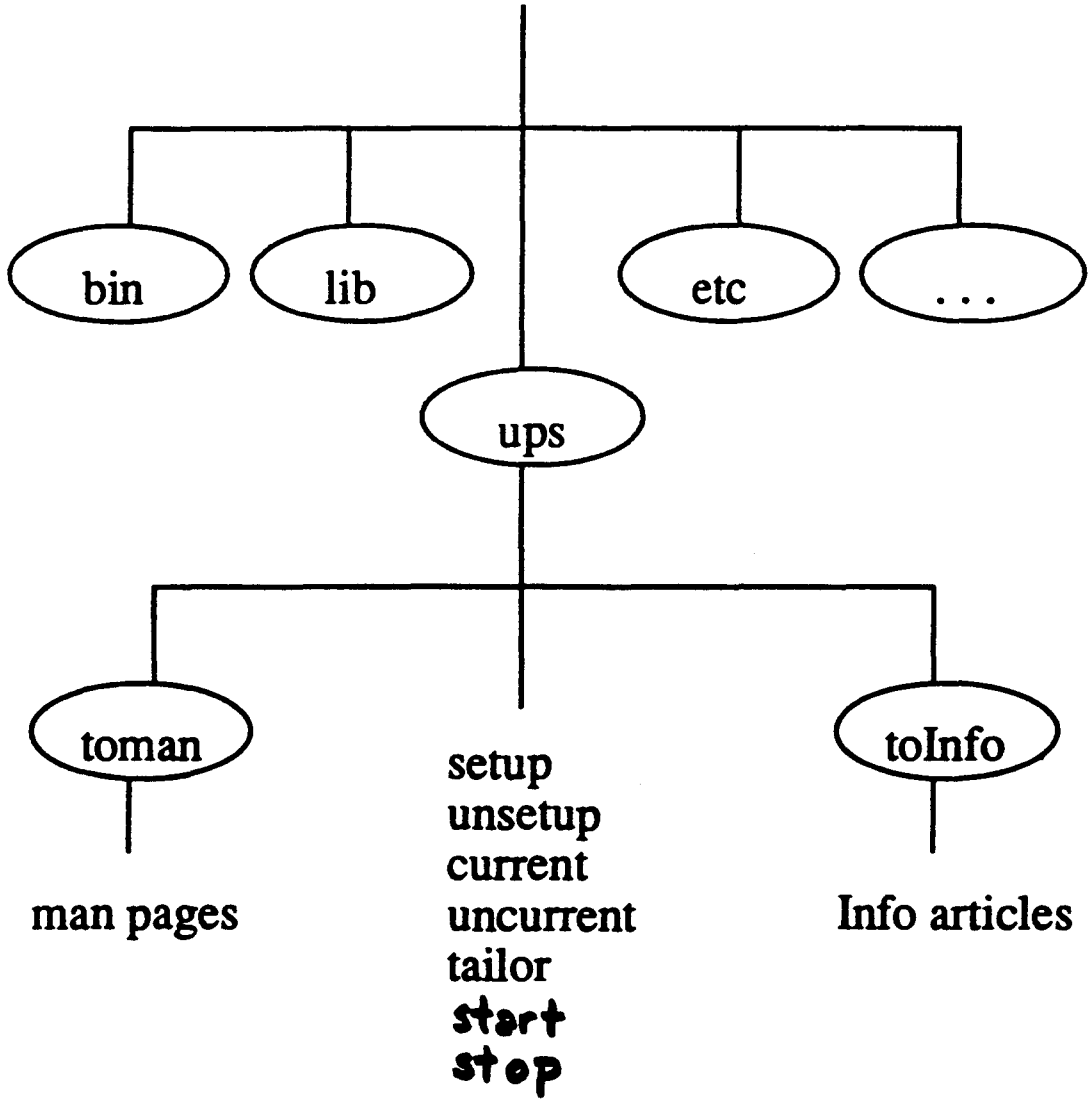
  => Build procedures - *scripts to build platform specific distribution version from sources*

- User => Access to software in a standard way. *"All" software "looks the same" to access - whatever the source. User learns where to look, what to look for*

- Local System Manager => Procedures to help manage system, *be informed of and have available new product releases, track installation of products*

Product Structure

CERN_DIR ⟶ home directory (e.g. /usr/products/fitbb/v1.0)



UPS Implementation

I. DATABASE

Simply ASCII text files

- free
- portable
- easy to read

II. MANIPULATION ROUTINES

C programs

- speed
- easier to maintain

/bin/csh

- official computing division shell
- designed for multiple shell support

Compiled under

- IRIX
- SunOS
- AIX
- ULTRIX

DATABASE FORMAT

database is a directory containing 1 ASCII file per product $PRODUCTS

instance fields include:

- version
- flavor
- home directory
- declarer's login id
- date declared

optional fields:

- list of authorized nodes
- list of use requirements
- list of build requirements
- master source directory
- tar file name

"pointers" to specific instances:

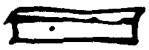
- current
- development
- test
- old
- new

SETUP CERN

CERN_DIR

AVAILABLE COMMANDS

USER:

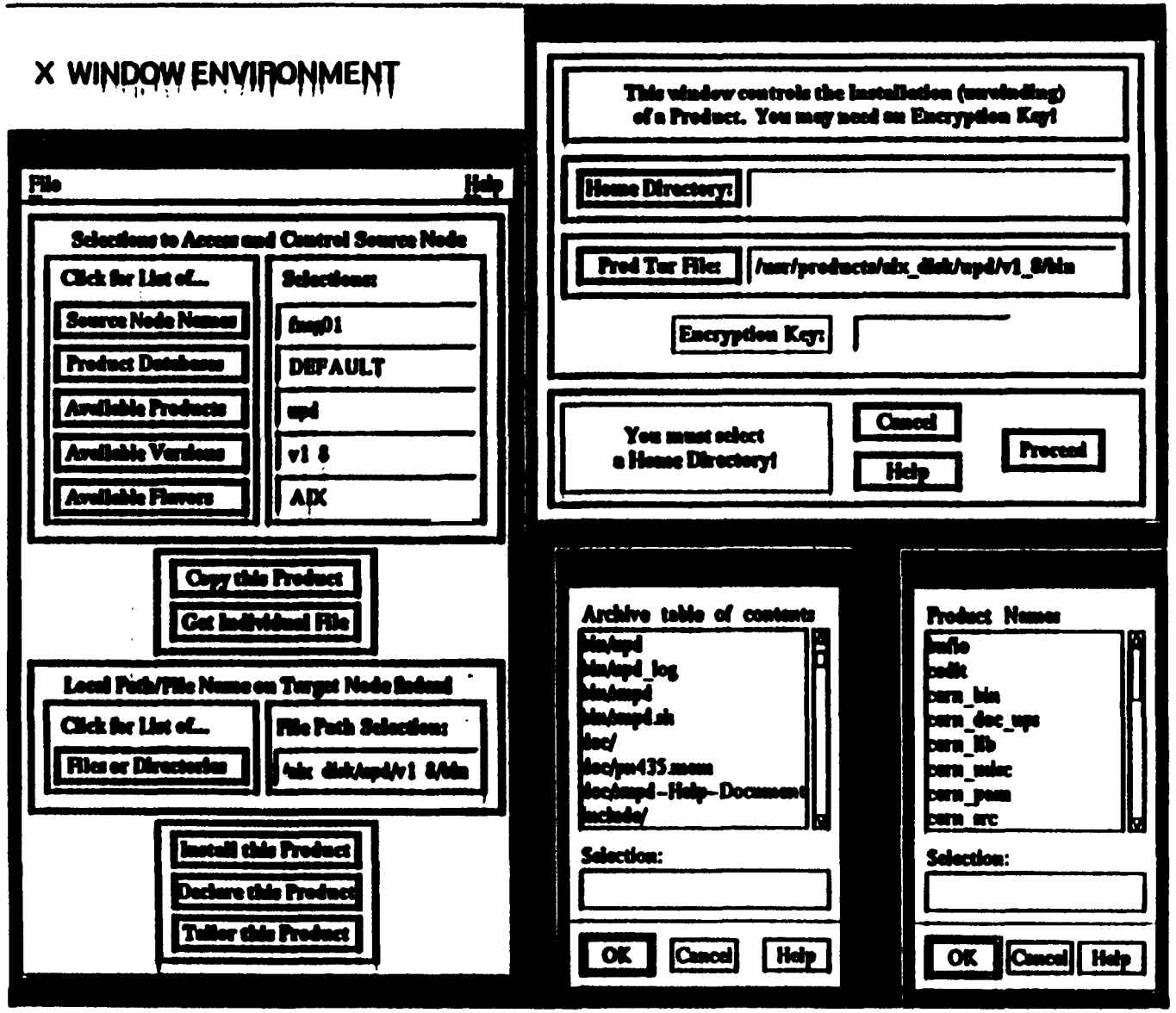
- list available products in various levels of detail ups list -f AIX -a -z 
- make a product available for use ups current
- remove a product from his environment unsetup.

PRODUCT MAINTAINER:

- add products/entries to the database ups declare
- remove products/entries from the database ups remove
- modify existing entries ups modify

PRODUCT INSTALLER:

- tailor a product for his system
- start products up
- shut products down
- make version "current"


X WINDOW ENVIRONMENT
File
Help
Selections to Access and Control Source Node
Click for List of...
Selections:
Source Node Names
fmp01
Product Databases
DEFAULT
Available Products
upd
Available Versions
v1 8
Available Flavors
AIX
Copy this Product
Get Individual File
Local Path/File Name on Target Node Indeed
Click for List of...
File Path Selection:
Files or Directories
Install this Product
Declare this Product
Tailor this Product
This window controls the installation (unwinding) of a Product. You may need an Encryption Key!
Home Directory:
Prod Tar File:
/usr/products/aix_disk/upd/v1_8/bin
Encryption Key:
You must select a Home Directory!
Cancel
Help
Proceed
Archive table of contents
bin/upd
bin/upd_log
bin/mpd
bin/mpd.sh
doc/
doc/pn435.mem
doc/mpd-Help-Document
include/
Selection:
OK
Cancel
Help
Product Names
mfile
pedit
cern_bin
cern_doc_ups
cern_lib
cern_misc
cern_pam
cern_src
Selection:
OK
Cancel
Help

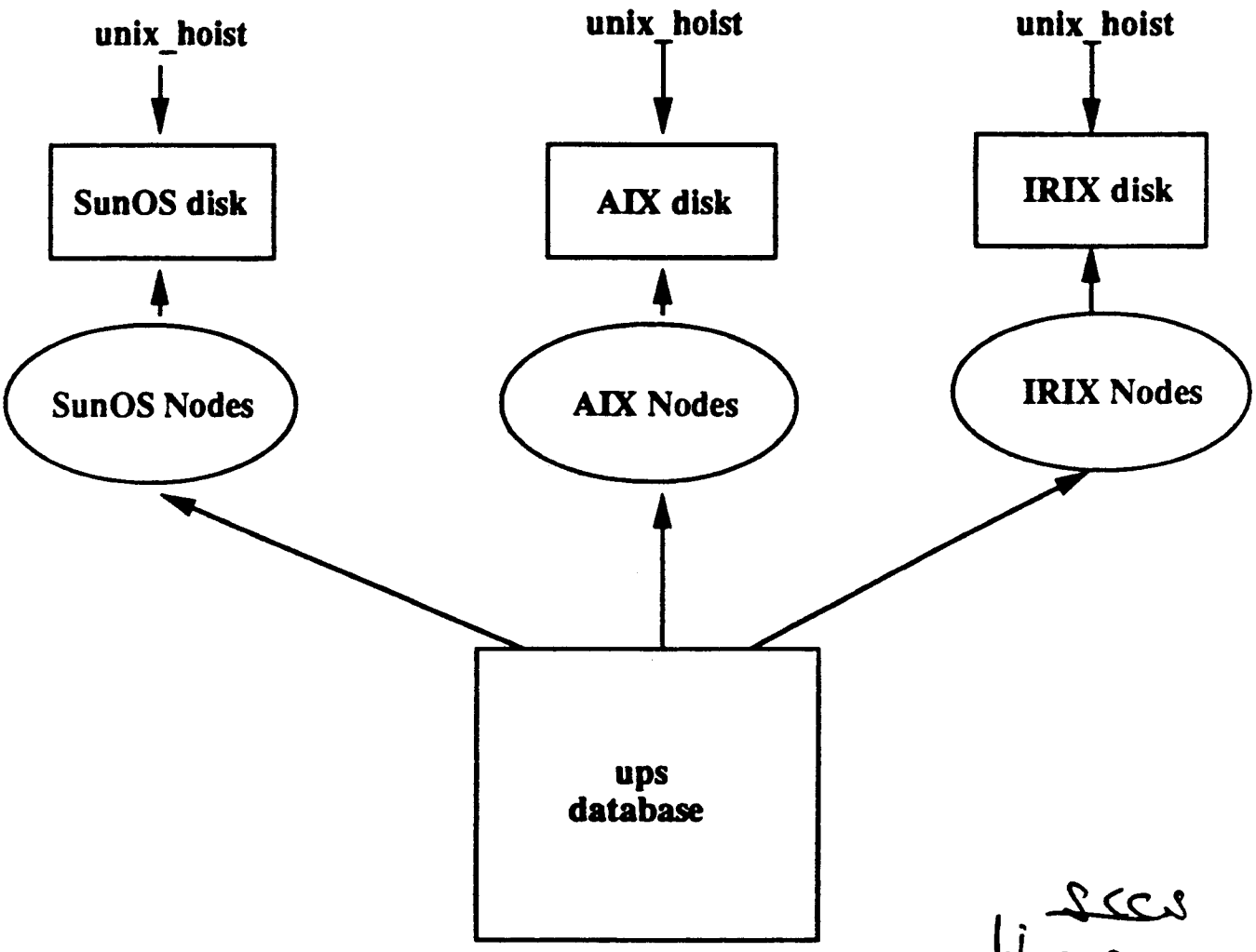
UNIX PRODUCT CENSUS
UPC

- query remote databases with supplied information
- generates summaries of requested information
- mails summaries to interested parties
- can run on a scheduled interval

Helps Software Maintainers:

- see types of products a remote system has
- know if specific sites need upgrades
- tally the number of sites that use a specific product

Online Support Cluster



SCCS
library

Three entries for unix_hoist in the database, one for each o/s type.
Ups "knows" o/s type, and will execute appropriate binaries

FUTURE ENHANCEMENTS

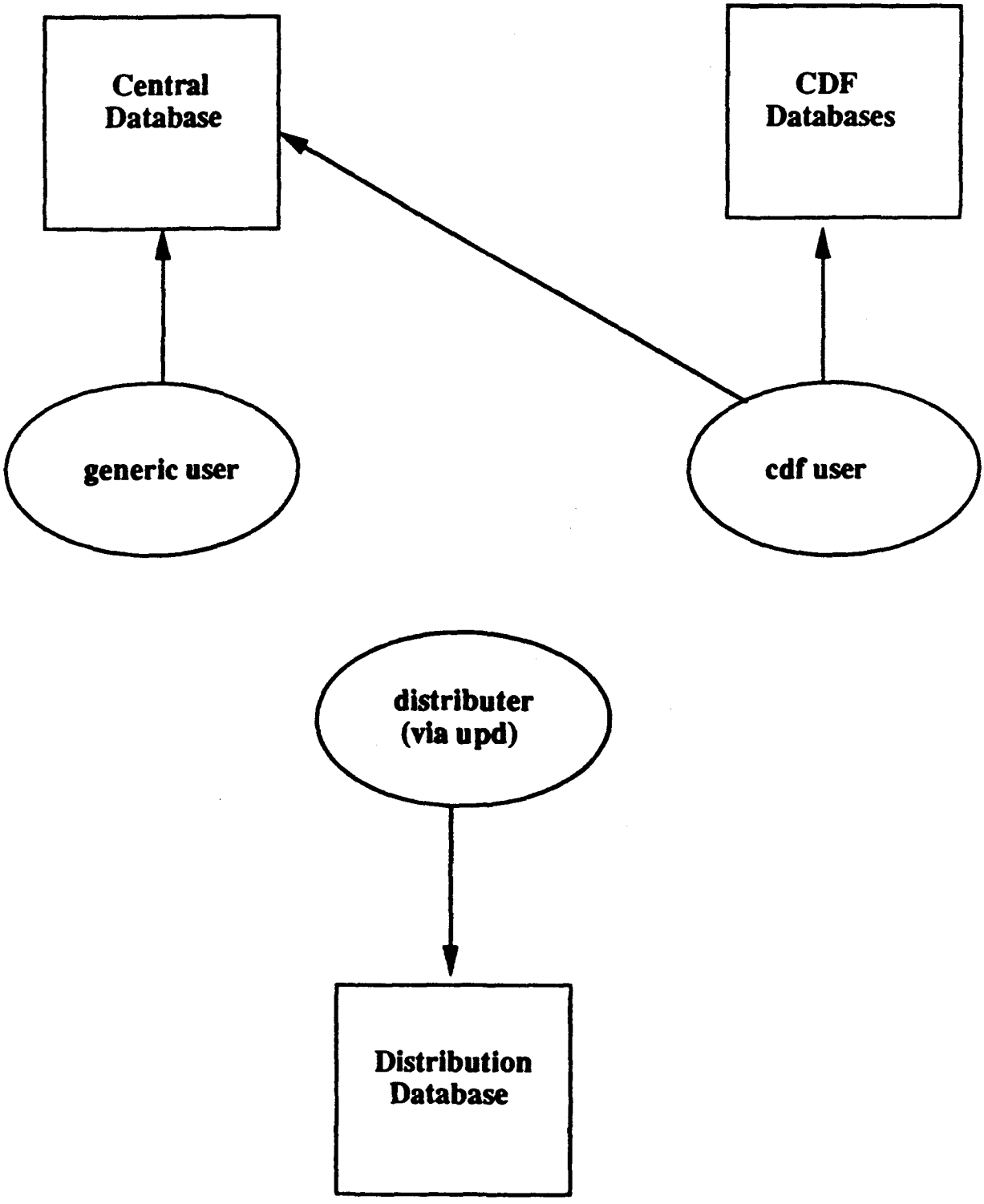
UNIX PRODUCT ARCHIVING
UPA

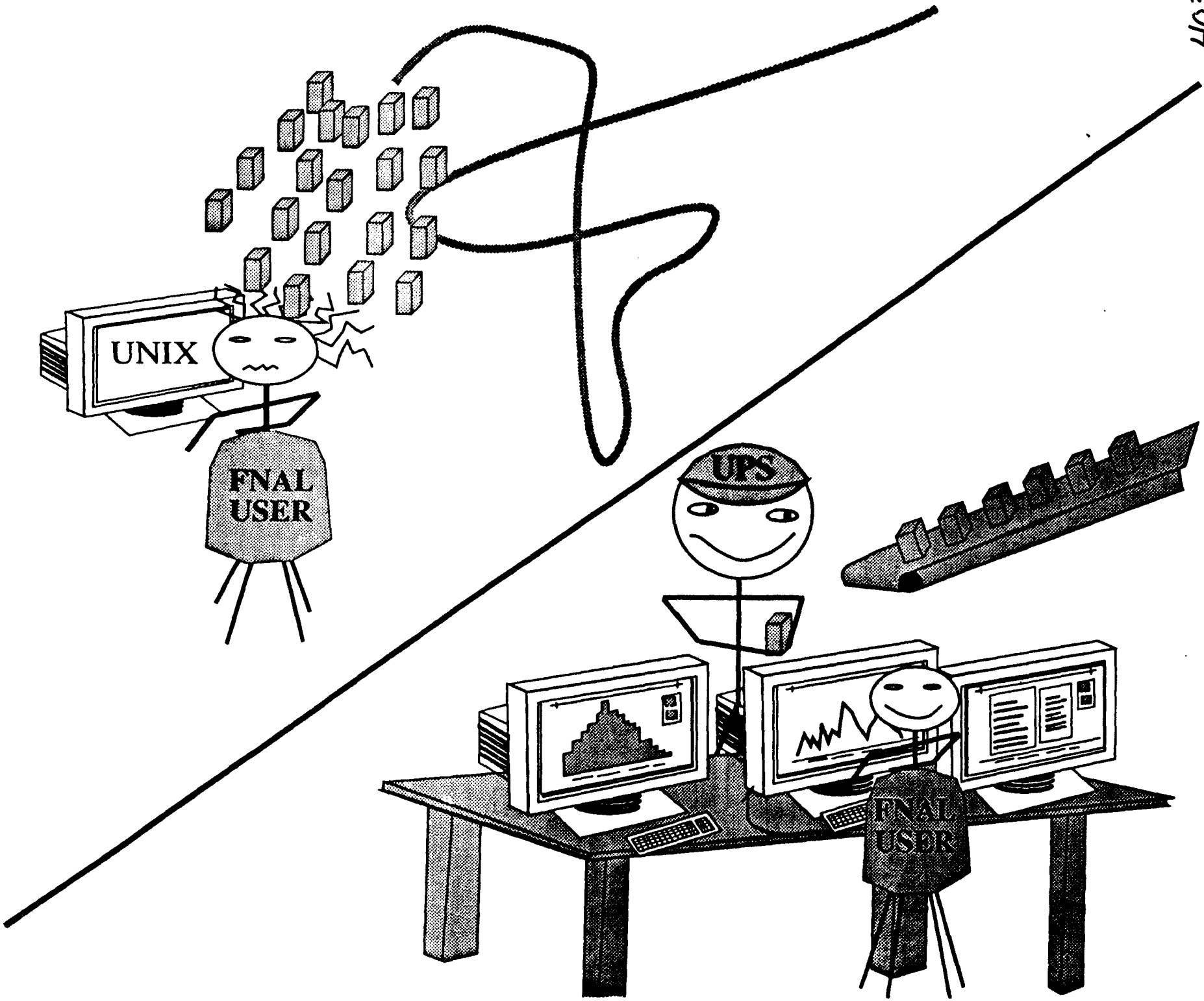
- provide a ups interface with an archiver product

UNIX PRODUCT BUILD
UPB

- make transition from master source area to a released version
- will build a released version for all necessary platforms
- support building of a product that requires multiple platforms

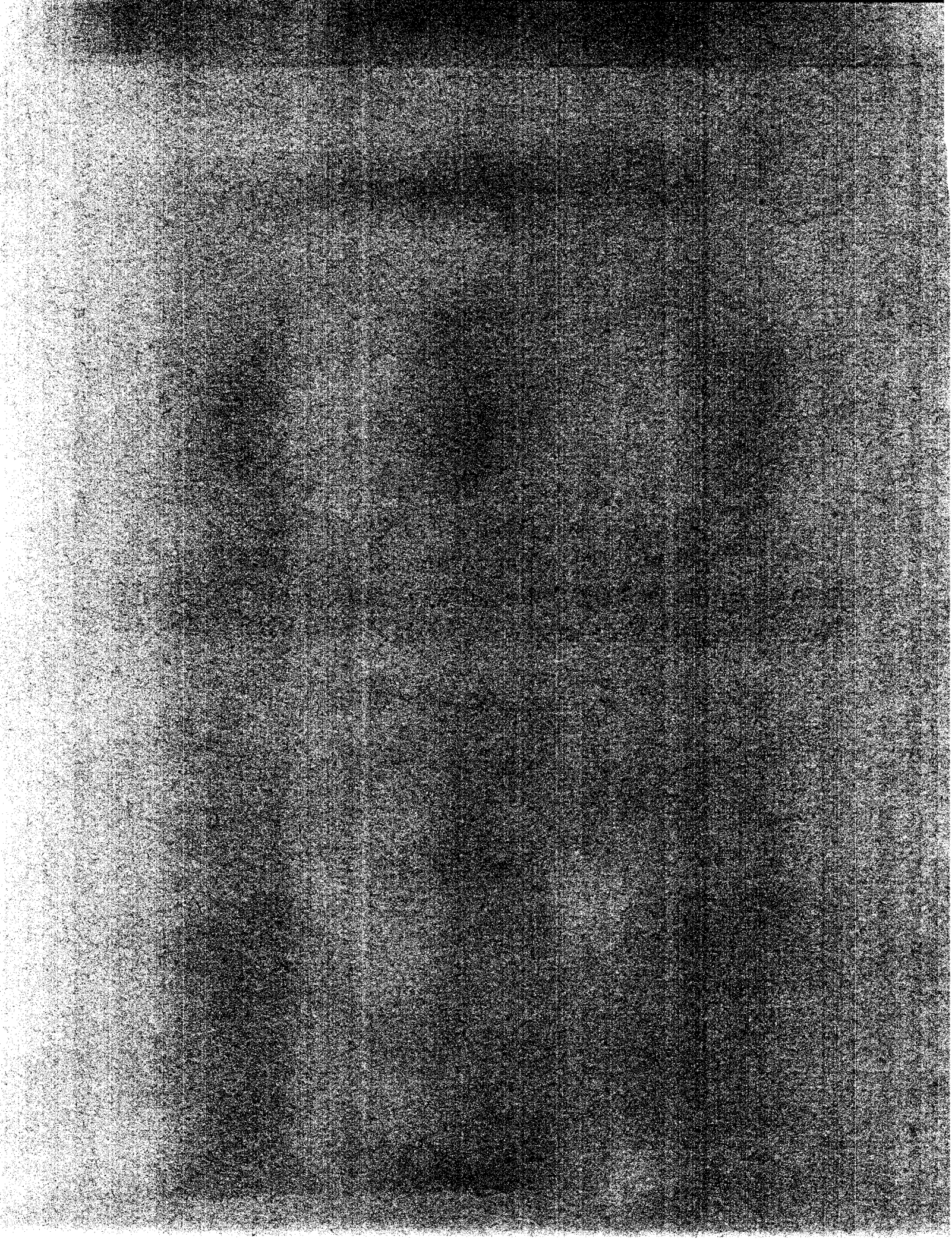
Central Facilities Node



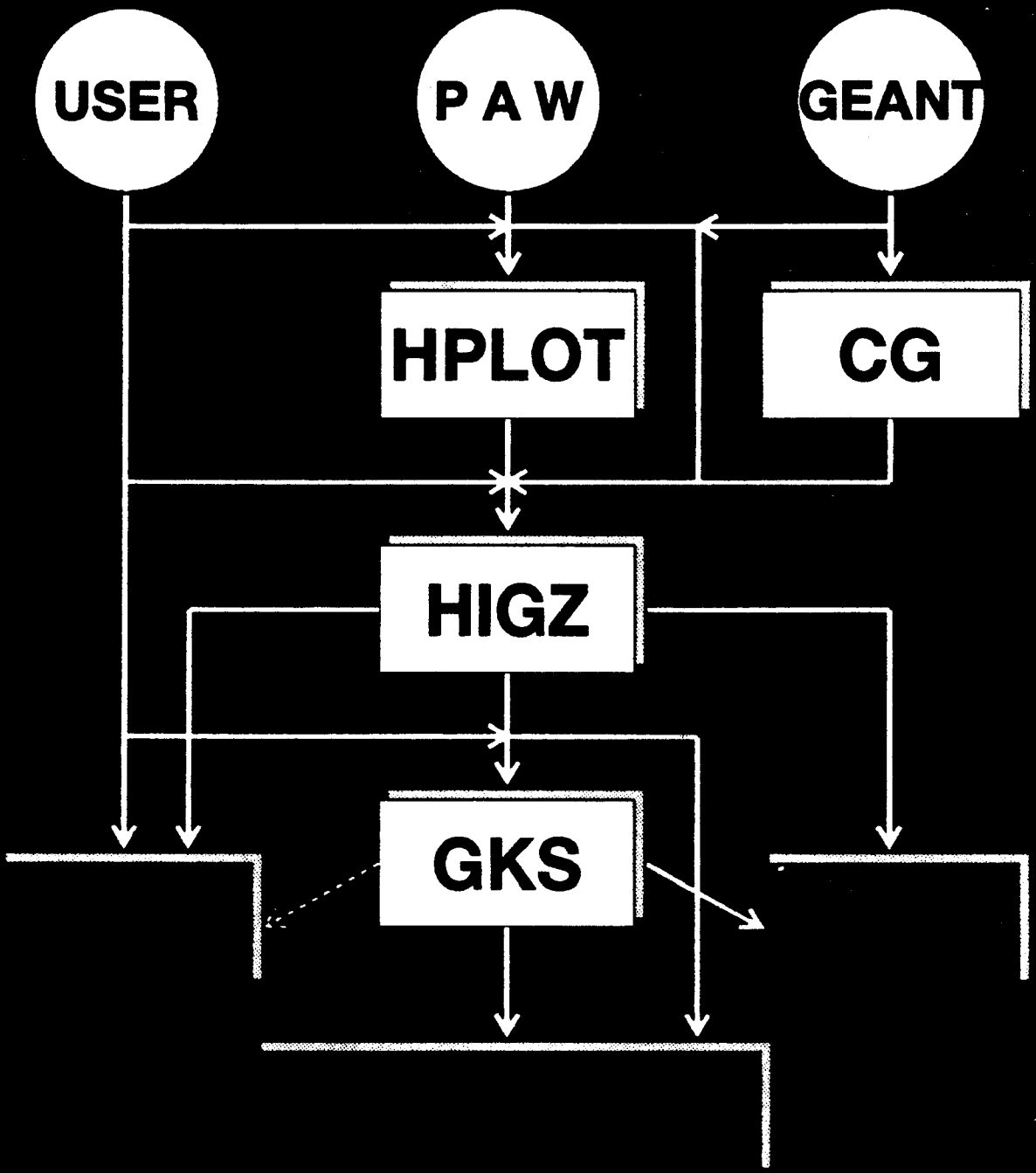

UNIX
FNAL
USER
UPS
FNAL
USER

"Status of Graphics at CERN"
"GKS, X11, HIGZ, PHIGS, PEX"
"The PHIGS Package Selected by CERN"
"Development of User Interface Systems (KUIP/MOTIF)"
"Real-time 3-D Viewers (with MOTIF and PHIGS/PEX)"

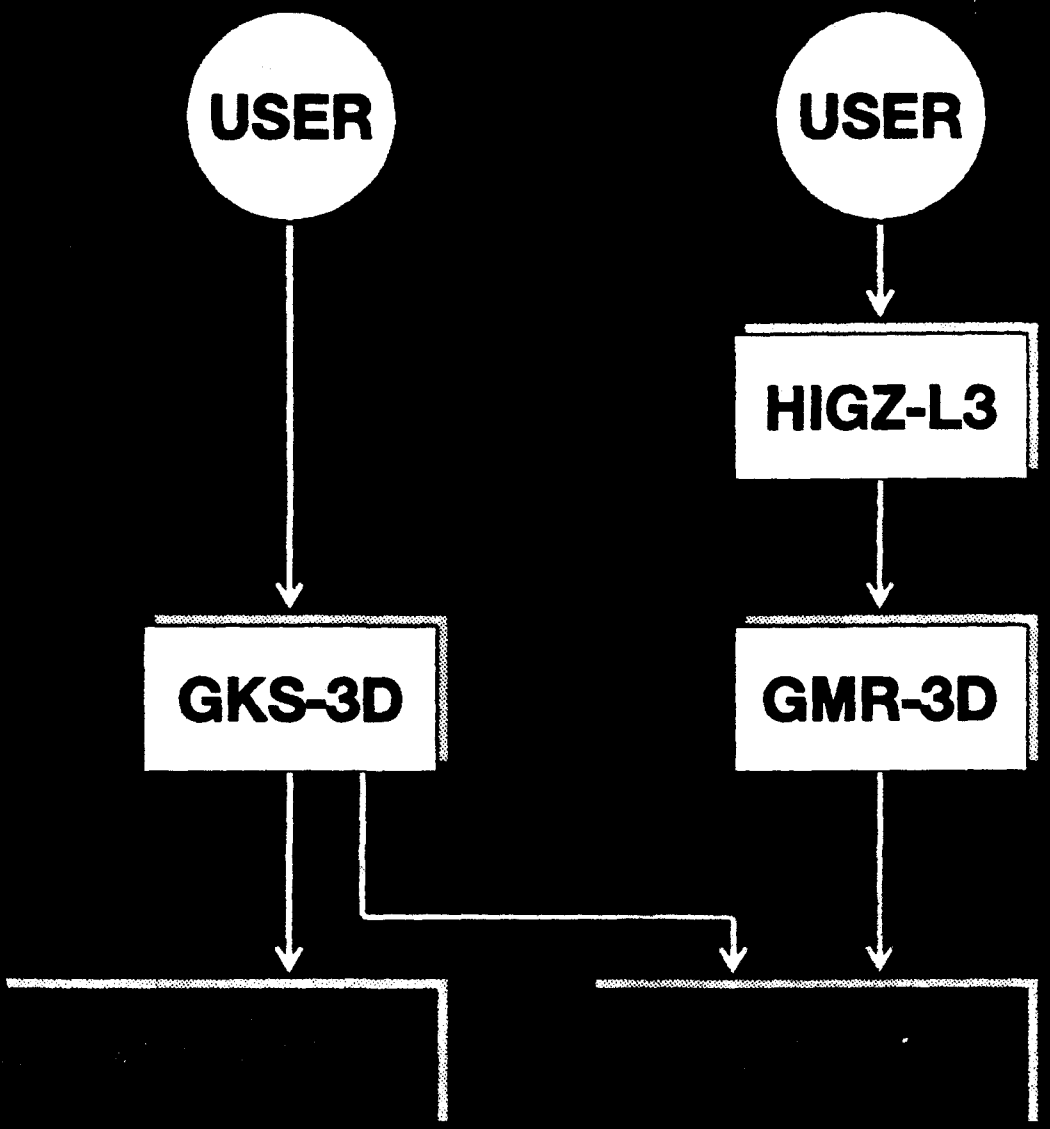
R. Brun, C. Vandoni



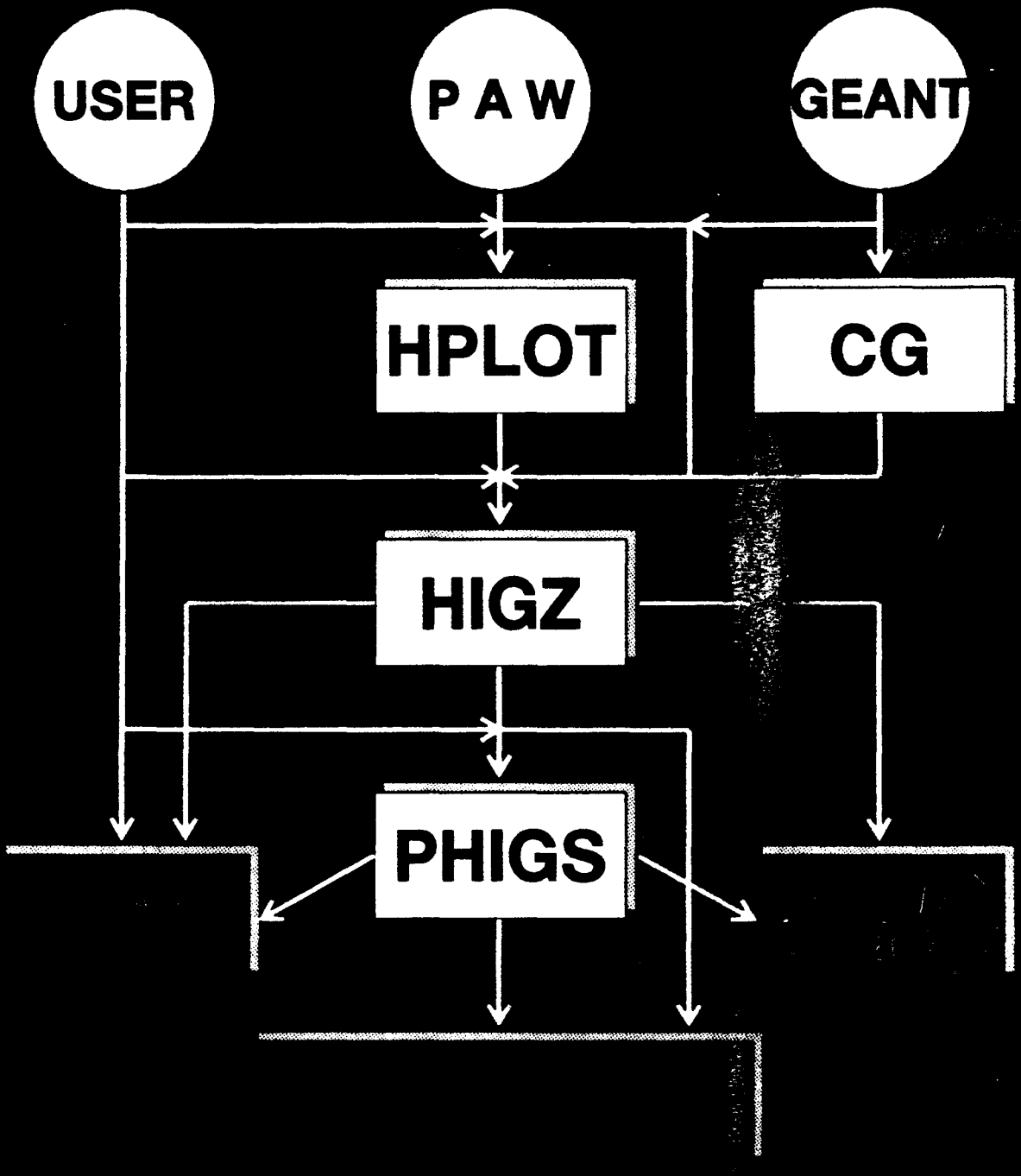
2-D Graphics



3-D Graphics Today



2-D/3-D Graphics



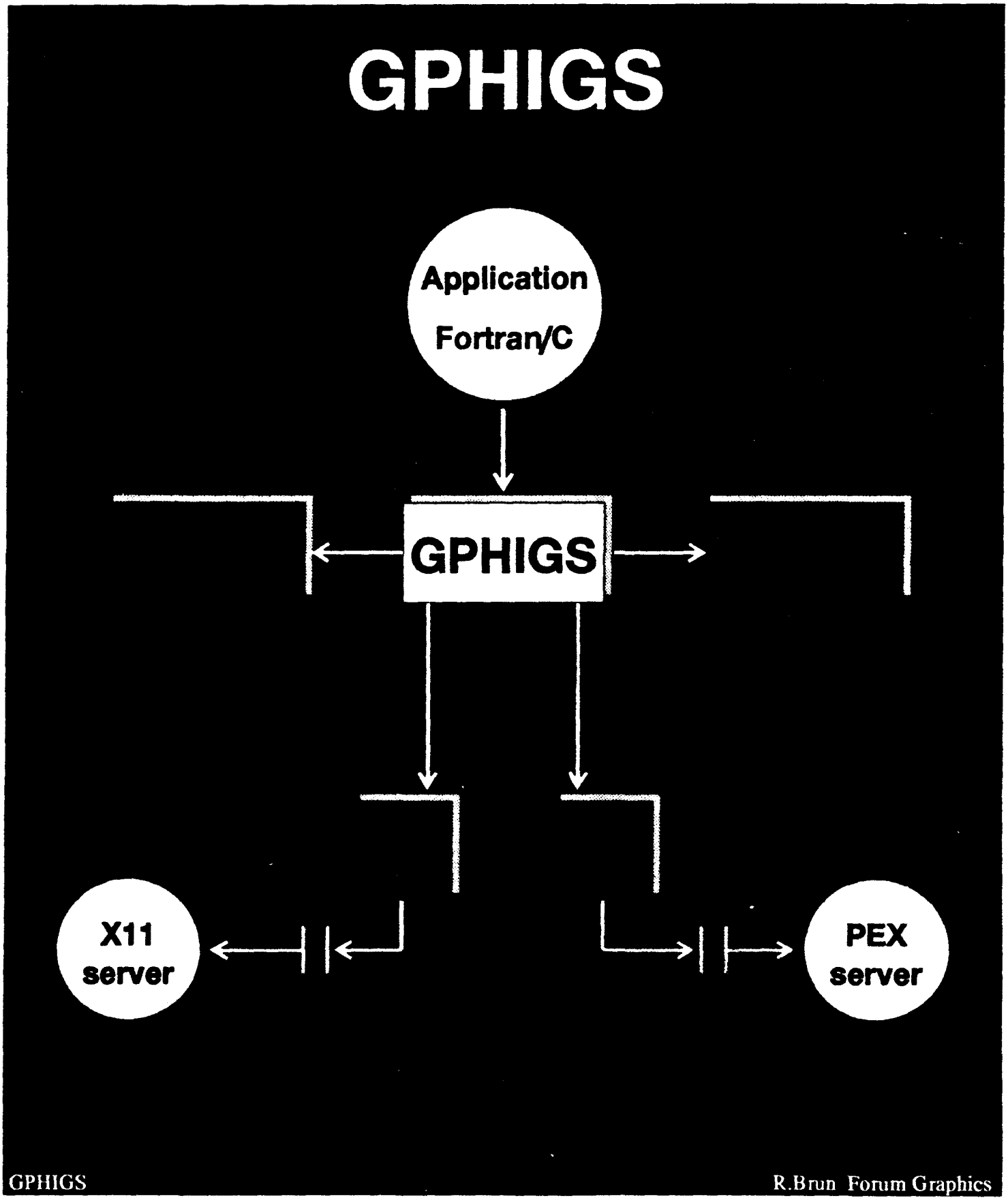
GKS

GKS service

- *Platforms:*
  - *VM/CMS, VMS, CRAY*
  - *UIS, VMS DecWindows, Ultrix, Apollo*

Summary:

- *GKS is frozen at present release level for general usage.*
- *Release 3.37 will not be released for general usage until it proves to be stable and fully compatible with the existing drivers.*
- *Install the X-driver only for some specific applications.*
- *Existing applications requiring X-driver will be moved to HIGZ.*
- *Applications based on GKS needing an X-driver which cannot be moved to HIGZ will have access to GKS 3.37 and its X-driver*


GPHIGS
Application
Fortran/C
GPHIGS
X11
server
PEX
server
GPHIGS
R.Brun Forum Graphics

PHIGS

- *Acquisition of a global licence for usage at CERN and for free distribution to the European collaborating institutes (normally executable, exceptionally source).*
- *Platforms: IBM VM/CMS, VAX VMS, CRAY, Apollo-HP, HP(HP/UX),Silicon Graphics, IBM RISC 6000, SUN, DEC Station(Ultrix), VAX Station(VMS-UIS,VMS-DecWindows)*
- *FORTRAN and C bindings and implementations required.*
- *Driver interfaces required: X-Windows, DEC-Windows, GL, Starbase, XGL, HP GL, CGM, and PostScript*
- *OSF/MOTIF integration and a PEX interface are also required.*
- *Very few suppliers provide a portable version.*
- *Companies contacted:*
  GTS/GRAL (Geramny)
  G5G (France)
  Template Graphics Software (UK and USA)
  Tektronix (USA)
  ITHACA Software (USA)
- *Deadline for replies: 15 July 1991*
- *Evaluation and benchmark tests: June-July 1991*

PHIGS

- *Acquisition of a global licence for usage at CERN and for free distribution to the European Affiliated Institutes (normally binary, exceptionally source).*
- *Platforms: IBM VM/CMS, VAX VMS, CRAY, Apollo-HP, HP(HP/UX),Silicon Graphics, IBM RISC 6000, SUN, DEC Station(Ultrix), VAX Station(VMS-UIS,VMS-DecWindows)*
- *FORTRAN and C bindings and implementations.*
- *Driver interfaces: X-Windows, DEC-Windows, GL, Starbase, XGL, HP GL, CGM, and PostScript.*
- *OSF/MOTIF integration and a PEX interface.*
- *Very few suppliers provide a portable version. Evaluation completed, supplier has been selected.*
- *There will be special conditions for:*
  - *Affiliated Institutes outside Europe*
  - *Non-HEP departments of Affiliated Institutes*
- *Distribution via CERN library*
- *Availability: 4th quarter 1991*

PHIGS licence conditions

- ***European Collaborating Institutes***

  Binary obtainable from CERN free of charge.

  Documentation payable to CERN.

- ***Non-European Collaborating Institutes***

  Binary obtainable from CERN against payment of 1,500.- Sfr. to the supplier. Documentation payable.

  Details of this agreement are now being negotiated.

- ***Non-HEP Departments of European Collaborating Institutes***

  Binary obtainable from the supplier against payment of a variable fee depending upon the installation (7,000-135,000 French Francs i.e. 1,750-35,000 Sfr.)

  Documentation obtainable from the supplier.

  Maintenance obtainable from the supplier.

User Interfaces

- Major developments in KUIP

  Command mode, Macro mode

  Menu and Panel mode

  Object Oriented mode

  Concept of Action Routine excellent for Unix Threads

  Could be generalized to write an :

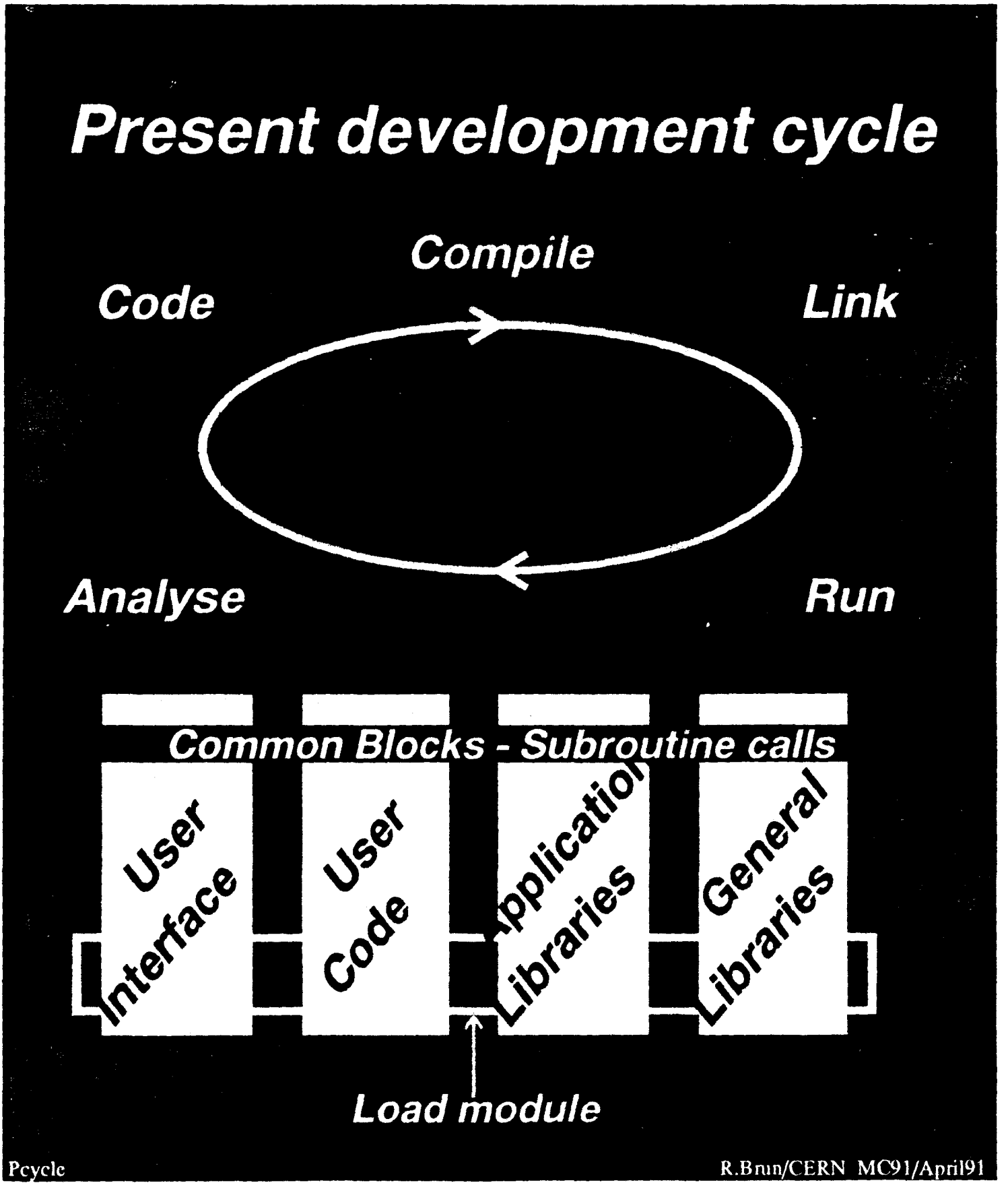
  Automatic Program Builder

- MOTIF Interface (big effort)

- User Interfaces for distributed systems

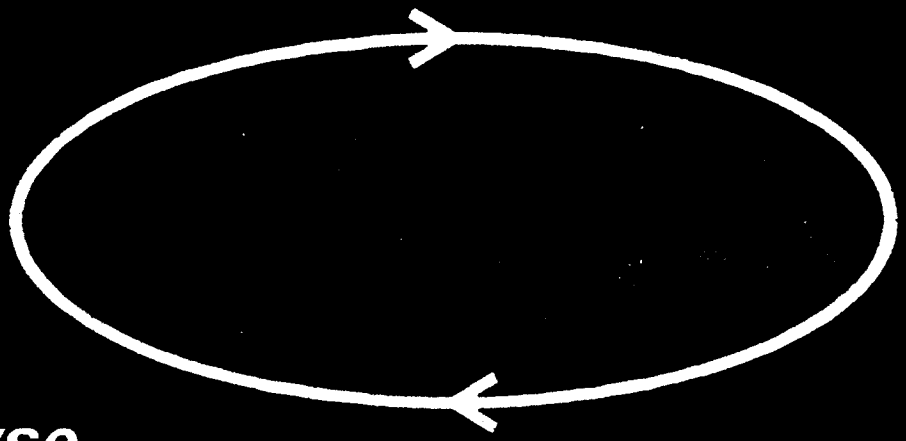
- KUIB: The KUIP Interface Builder




Present development cycle
Compile
Code
Link
Analyse
Run
Common Blocks - Subroutine calls
User Interface
User Code
Application Libraries
General Libraries
Load module
Pcycle
R.Brun/CERN MC91/April91

Future development cycle

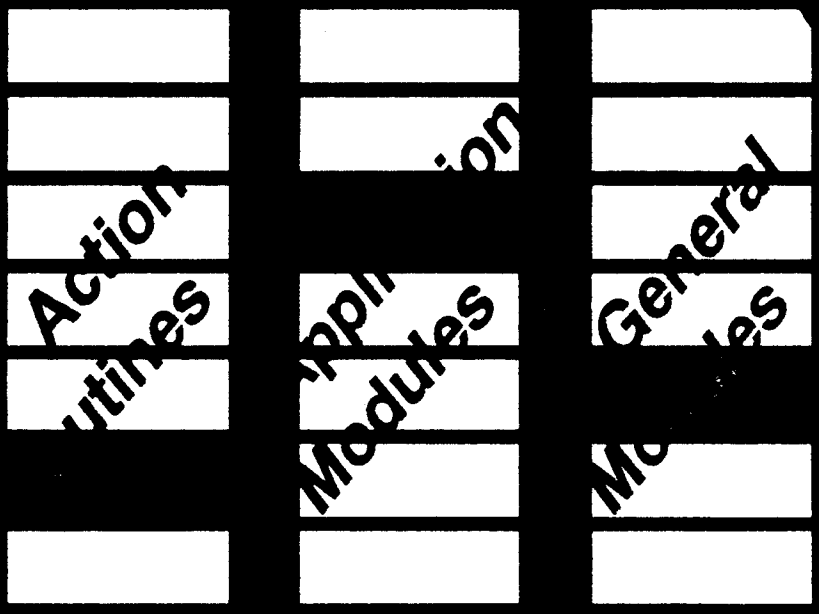
(Code action routine)



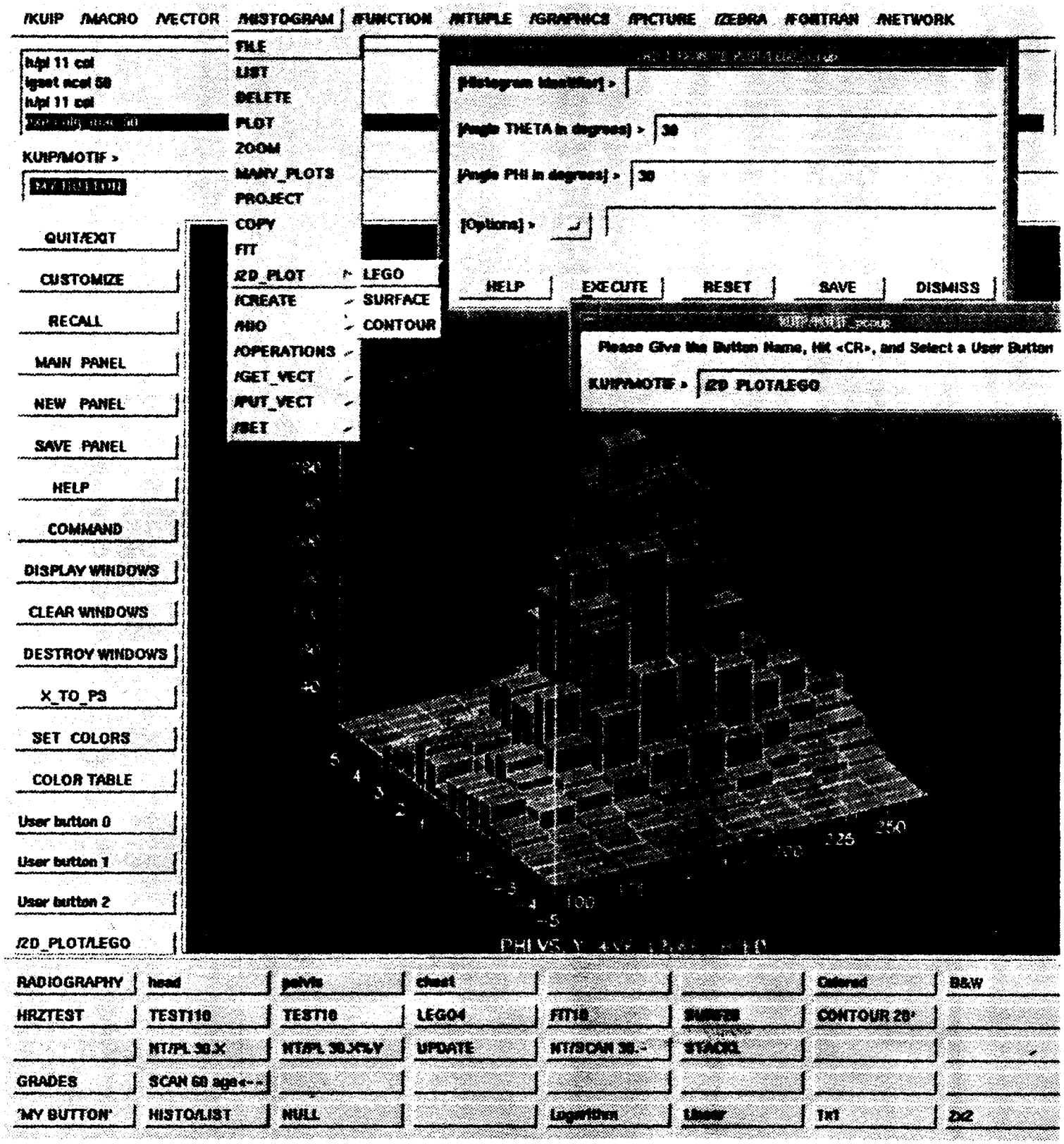
Analyse

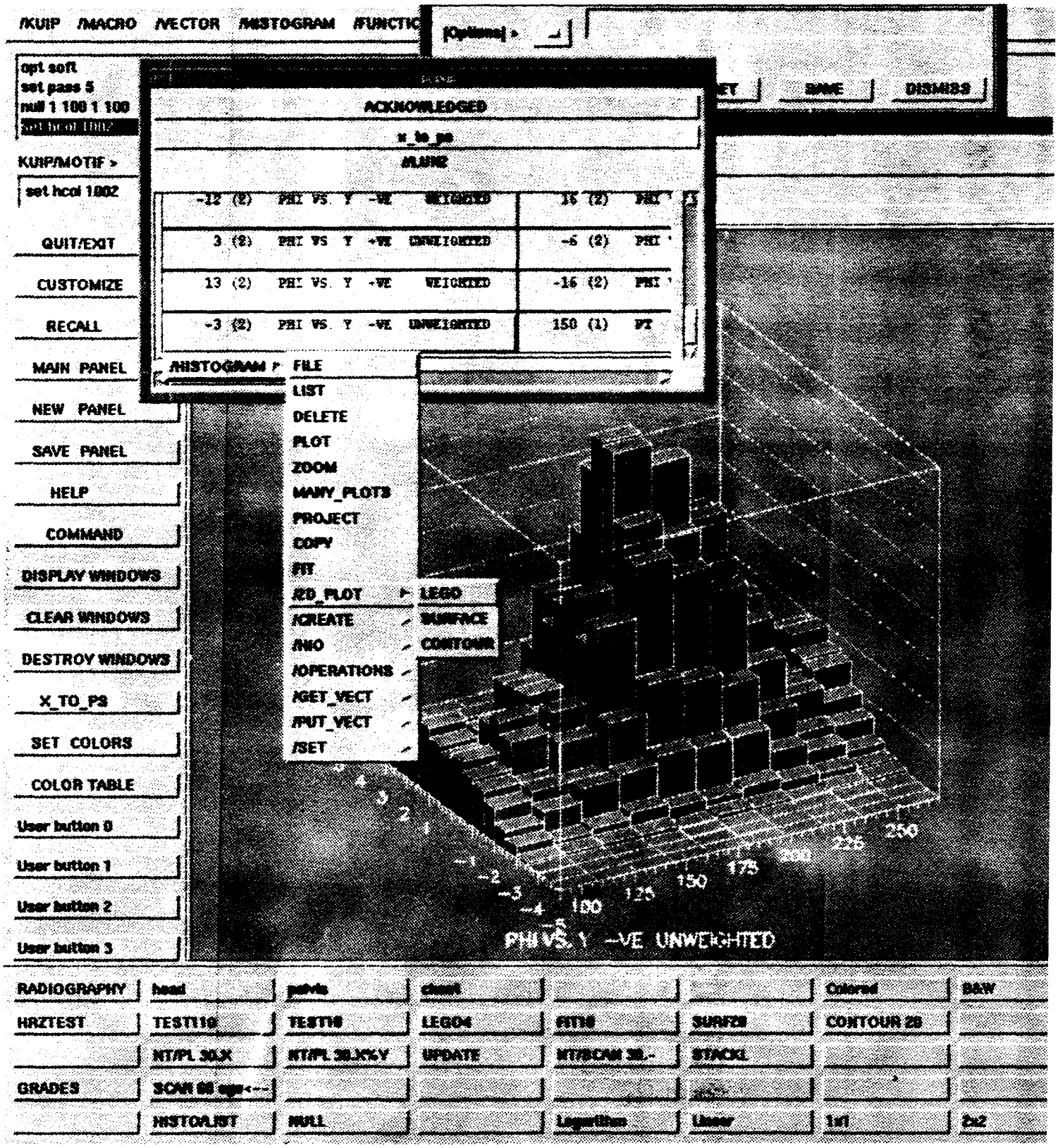
Run

Shared memory, NCS, Threads..

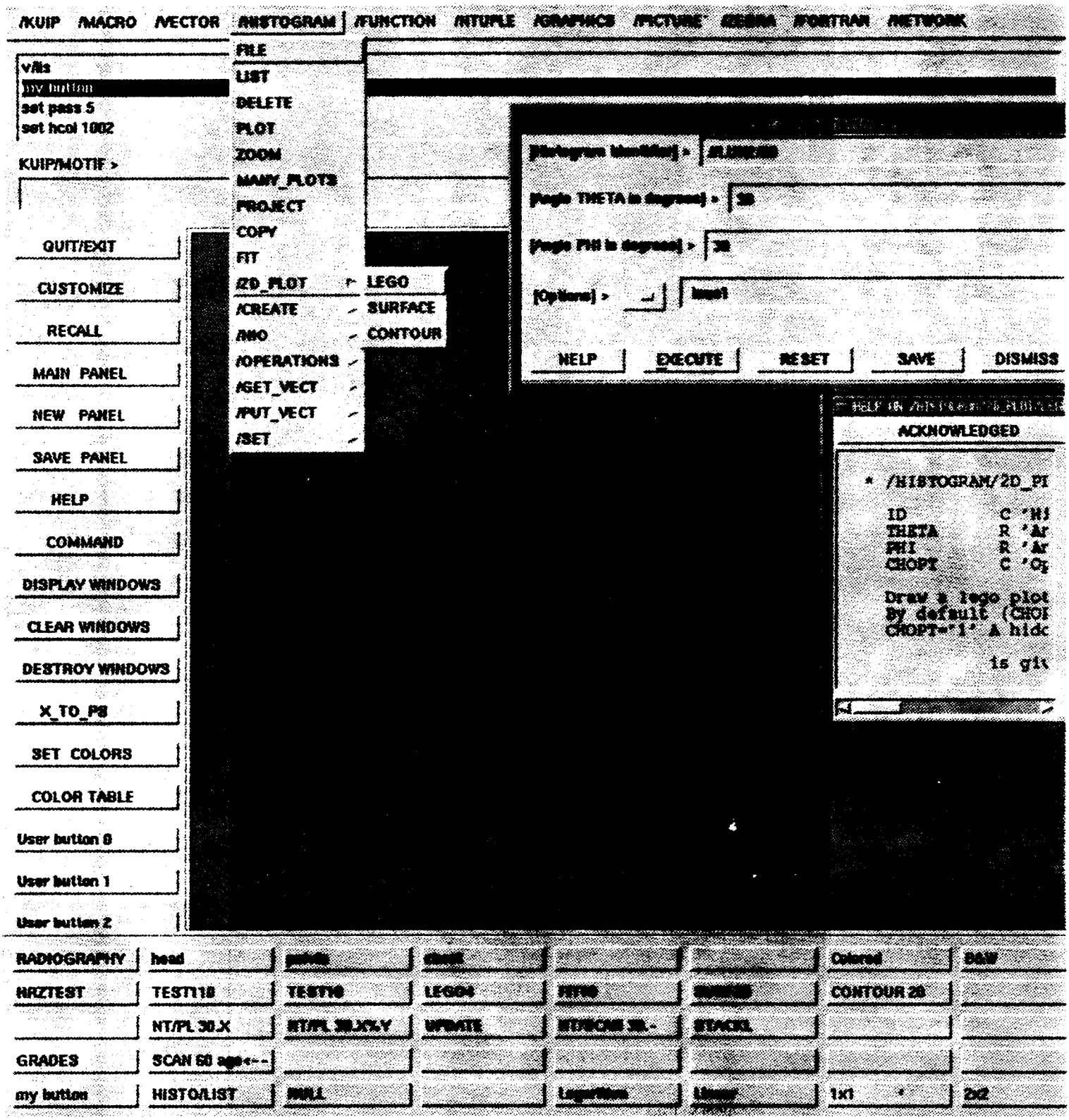


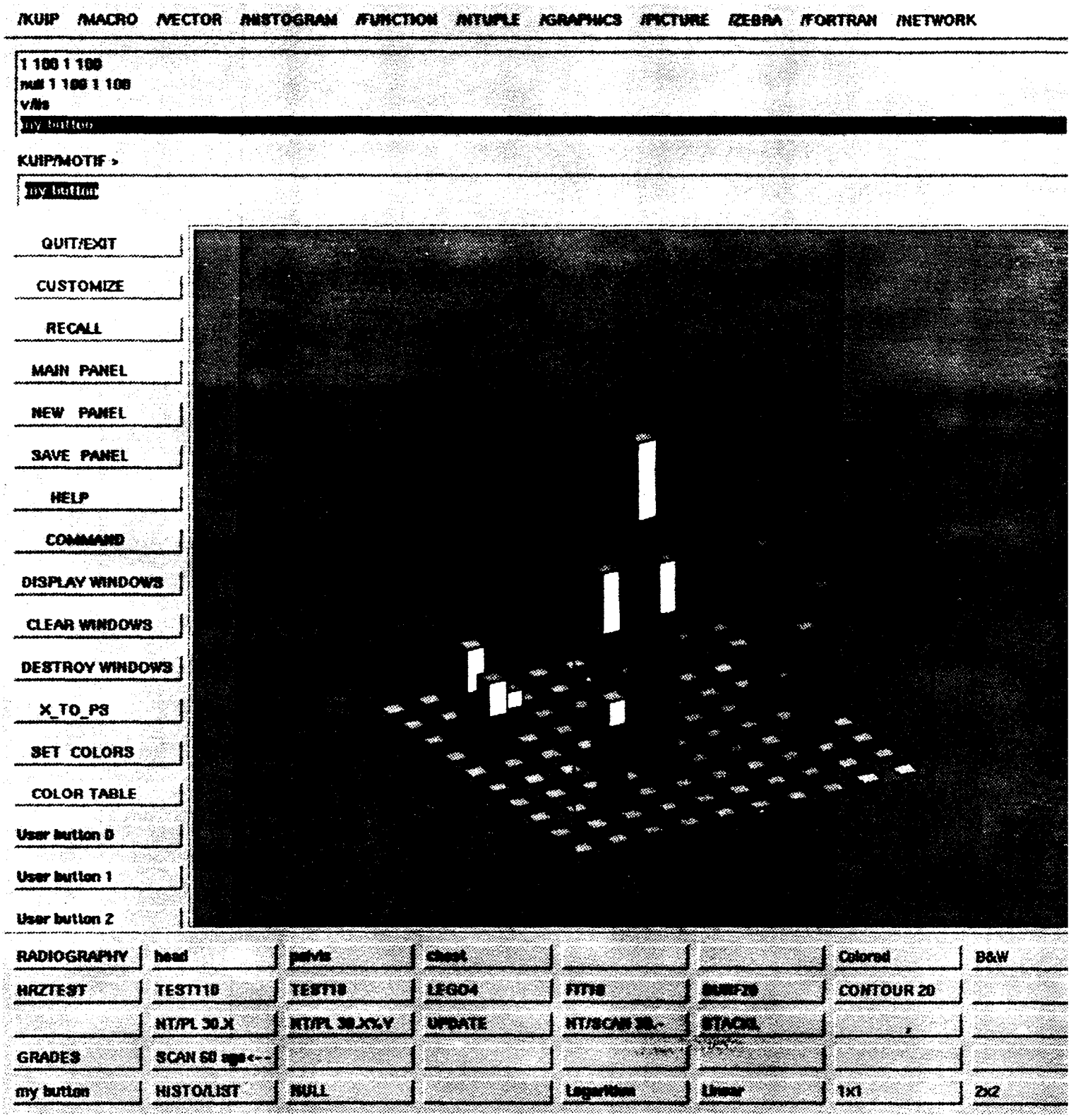

/KUIP /MACRO /VECTOR /HISTOGRAM /FUNCTION /NTUPLE /GRAPHICS /PICTURE /ZEBRA /FORTRAN /NETWORK
h/pl 91
h/pl 2
h/pl 2
h/pl 10
KUIP/MOTIF >
EXIT
PANEL
HELP COMMAND
ENTER COMMAND
RECALL COMMAND
SET COLOR
GET GEOMETRY
WELCOME
TO
P A W

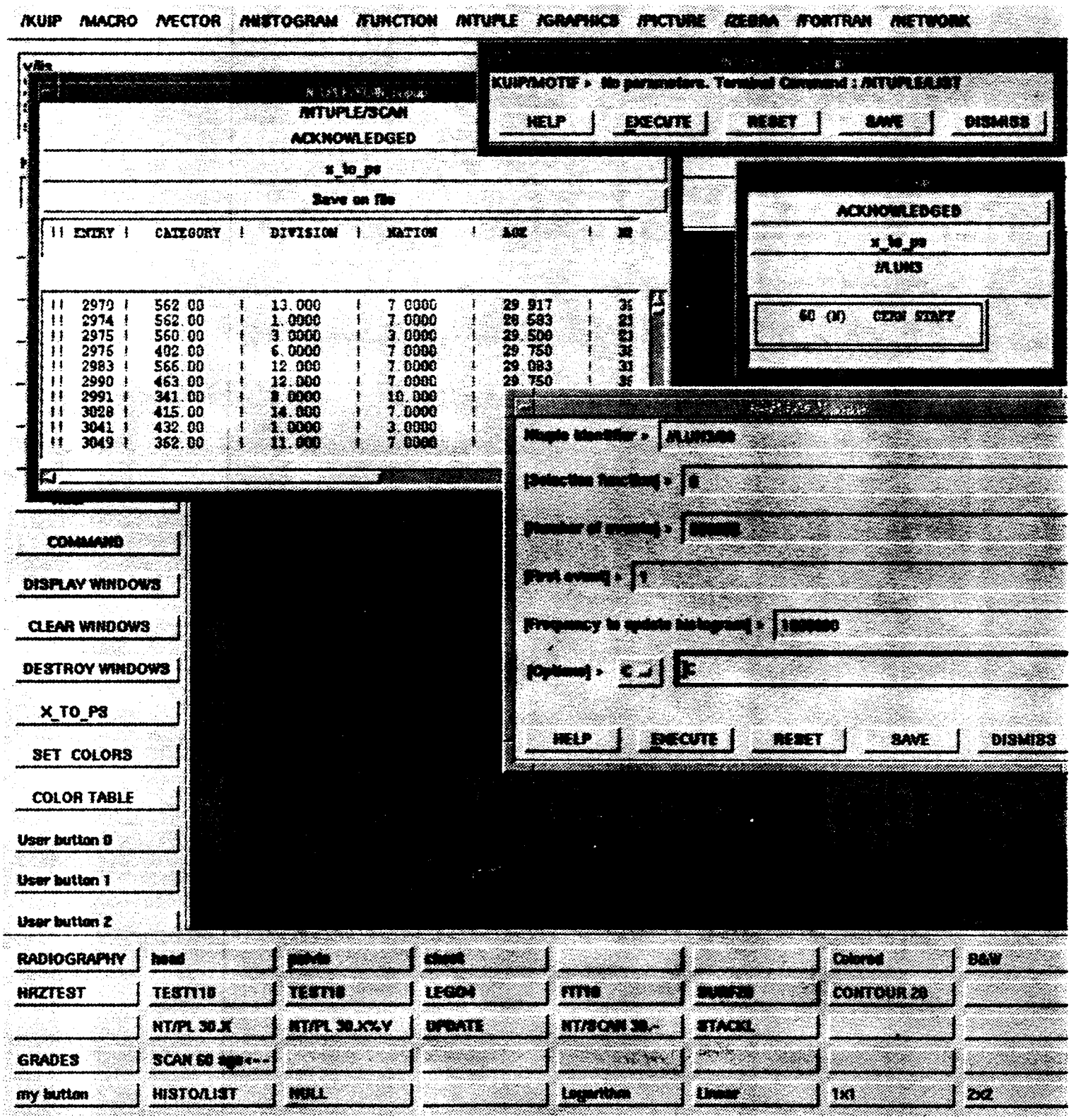

/KUIP /MACRO /VECTOR /HISTOGRAM /FUNCTION /NTUPLE /GRAPHICS /PICTURE /ZEBRA /FORTRAN /NETWORK
FILE
LIST
DELETE
PLOT
ZOOM
MANY_PLOTS
PROJECT
COPY
FIT
/2D_PLOT
/CREATE
/I/O
/OPERATIONS
/GET_VECT
/PUT_VECT
/SET
LEGO
SURFACE
CONTOUR
[Histogram identifier] >
[Angle THETA in degrees] > 30
[Angle PHI in degrees] > 30
[Options] >
HELP
EXECUTE
RESET
SAVE
DISMISS
Please Give the Button Name, Hit <CR>, and Select a User Button
KUIP/MOTIF > /2D_PLOT/LEGO
KUIP/MOTIF >
QUIT/EXIT
CUSTOMIZE
RECALL
MAIN PANEL
NEW PANEL
SAVE PANEL
HELP
COMMAND
DISPLAY WINDOWS
CLEAR WINDOWS
DESTROY WINDOWS
X_TO_P3
SET COLORS
COLOR TABLE
User button 0
User button 1
User button 2
/2D_PLOT/LEGO
RADIOGRAPHY
head
pelvis
chest
Colored
B&W
HRZTEST
TEST110
TEST10
LEGO4
FIT10
CONTOUR 20
NT/PL 30.X
NT/PL 30.X&Y
UPDATE
NT/SCAN 30.-
STACKL
GRADES
SCAN 60 age<--
'MY BUTTON'
HISTO/LIST
NULL
Logarithm
Linear
1x1
2x2

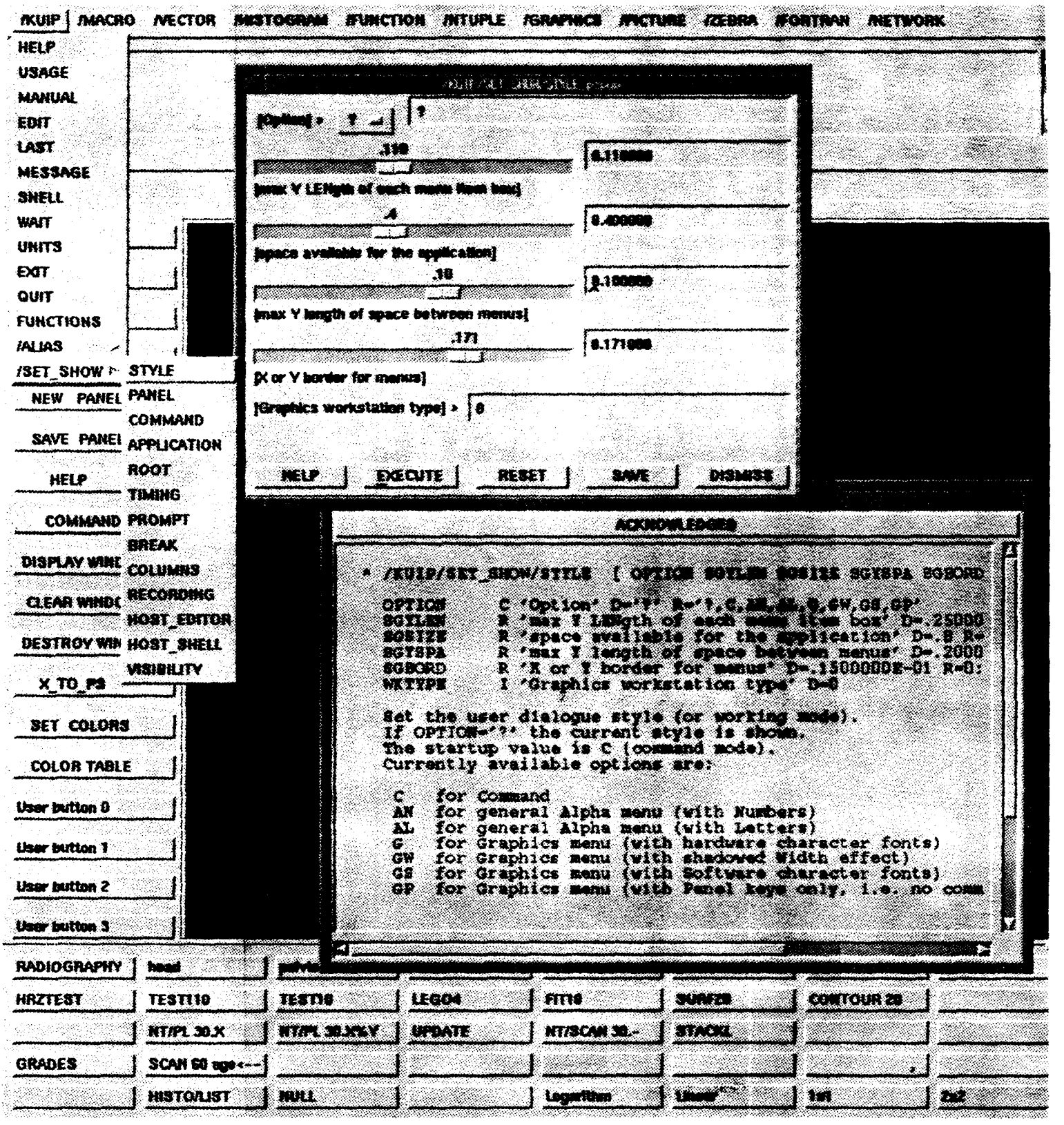

/KUIP /MACRO /VECTOR /HISTOGRAM /FUNCTIO
opt soft
set pass 5
null 1 100 1 100
KUIP/MOTIF >
set hcol 1002
QUIT/EXIT
CUSTOMIZE
RECALL
MAIN PANEL
NEW PANEL
SAVE PANEL
HELP
COMMAND
DISPLAY WINDOWS
CLEAR WINDOWS
DESTROY WINDOWS
X_TO_PS
SET COLORS
COLOR TABLE
User button 0
User button 1
User button 2
User button 3
ACKNOWLEDGED
x_to_ps
/LUN2
-12 (2) PHI VS Y -VE WEIGHTED 16 (2) PHI
3 (2) PHI VS Y +VE UNWEIGHTED -6 (2) PHI
13 (2) PHI VS Y -VE WEIGHTED -16 (2) PHI
-3 (2) PHI VS Y -VE UNWEIGHTED 150 (1) PH
/HISTOGRAM
FILE
LIST
DELETE
PLOT
ZOOM
MANY_PLOTS
PROJECT
COPY
FIT
/2D_PLOT
/CREATE
/NO
/OPERATIONS
/GET_VECT
/PUT_VECT
/SET
LEGO
SURFACE
CONTOUR
DISMISS
PHI VS. Y -VE UNWEIGHTED
RADIOGRAPHY
HRZTEST
GRADES
head
pelvis
chest
Colored
B&W
TEST110
TEST18
LEGO4
FIT18
SURF2B
CONTOUR 2B
NT/PL 30.X
NT/PL 30.X%Y
UPDATE
NT/SCAN 30.-
STACK1
HISTOLIST
NULL
Logarithm
Linear
1x1
2x2

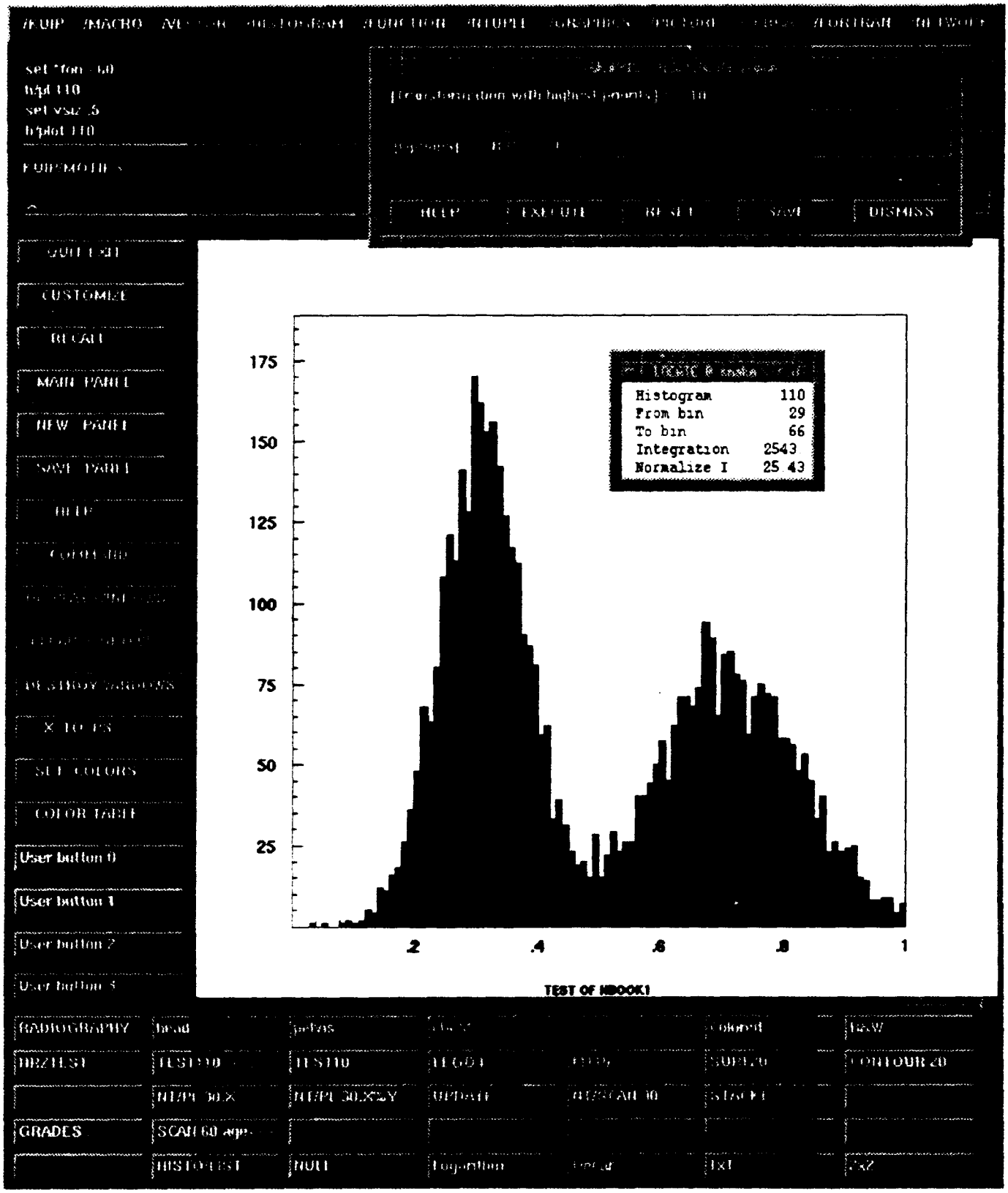
418

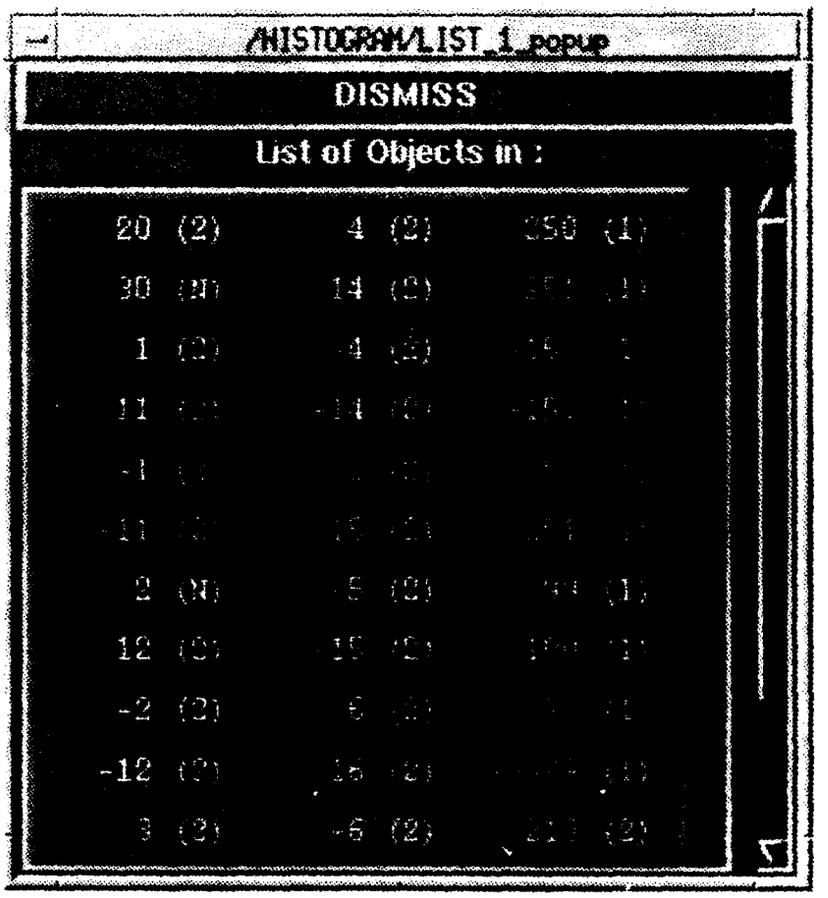

/KUIP /MACRO /VECTOR /HISTOGRAM /FUNCTION /NTUPLE /GRAPHICS /PICTURE /GEANA /FORTRAN /NETWORK
FILE
LIST
DELETE
PLOT
ZOOM
MANY_PLOTS
PROJECT
COPY
FIT
/2D_PLOT
/CREATE
/INFO
/OPERATIONS
/GET_VECT
/PUT_VECT
/SET
LEGO
SURFACE
CONTOUR
v/ls
my button
set pass 5
set hcol 1002
KUIP/MOTIF >
QUIT/EXIT
CUSTOMIZE
RECALL
MAIN PANEL
NEW PANEL
SAVE PANEL
HELP
COMMAND
DISPLAY WINDOWS
CLEAR WINDOWS
DESTROY WINDOWS
X_TO_PS
SET COLORS
COLOR TABLE
User button 0
User button 1
User button 2
HELP
EXECUTE
RESET
SAVE
DISMISS
ACKNOWLEDGED
* /HISTOGRAM/2D_PI
ID C 'Hi
THETA R 'Ar
PHI R 'Ar
CHOPT C 'Cl
Draw a lego plot
By default (CHOI
CHOPT='1' A hidc
is gi
RADIOGRAPHY
head
Colored
B&W
HRZTEST
TEST118
TEST10
LEGO4
CONTOUR 20
NT/PL 30.X
GRADES
SCAN 50 ago<--
my button
HISTO/LIST
1x1
2x2

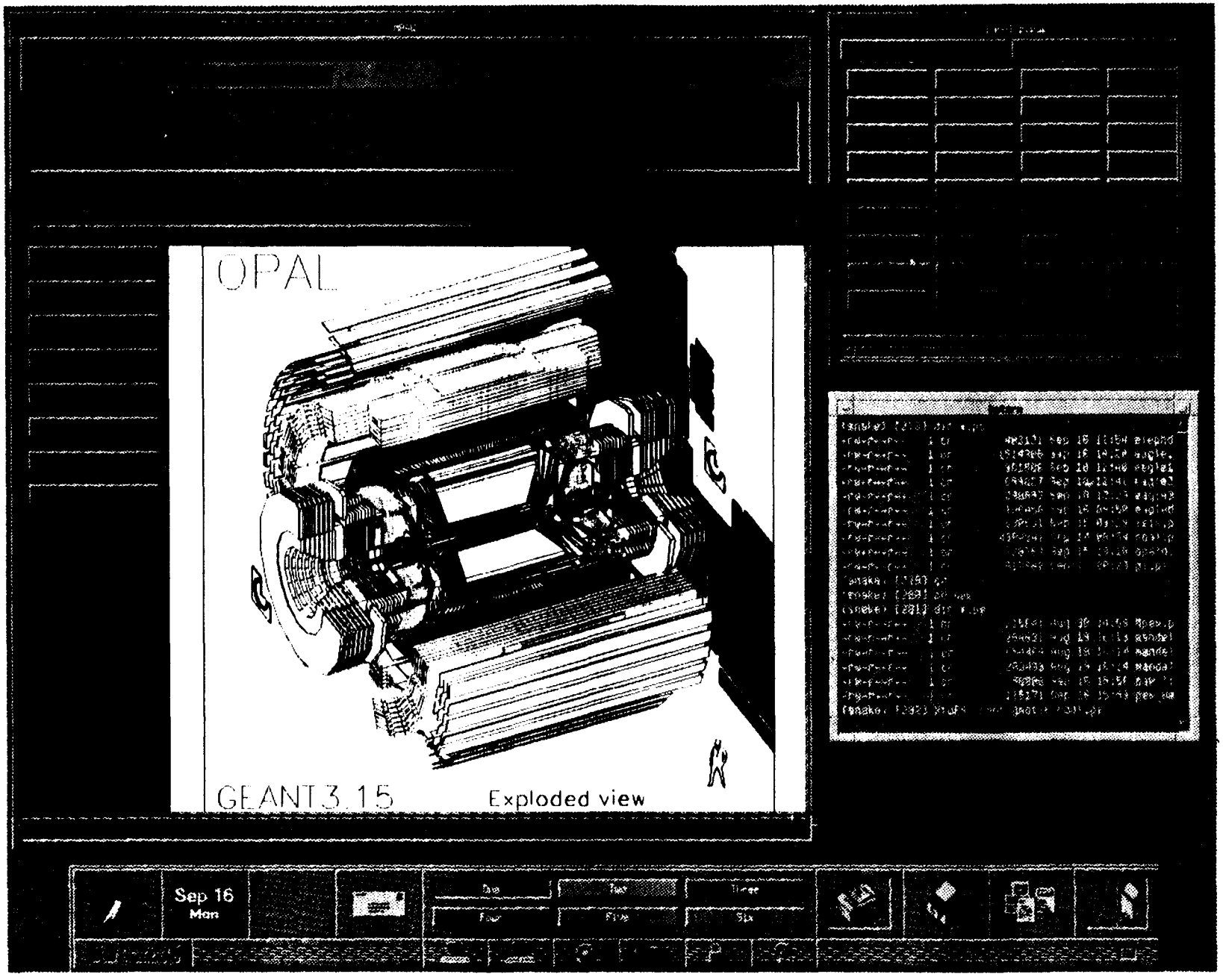

/KUIP /MACRO /VECTOR /HISTOGRAM /FUNCTION /NTUPLE /GRAPHICS /PICTURE /ZEBRA /FORTRAN /NETWORK
1 100 1 100
null 1 100 1 100
v/ls
my button
KUIP/MOTIF >
my button
QUIT/EXIT
CUSTOMIZE
RECALL
MAIN PANEL
NEW PANEL
SAVE PANEL
HELP
COMMAND
DISPLAY WINDOWS
CLEAR WINDOWS
DESTROY WINDOWS
X_TO_PS
SET COLORS
COLOR TABLE
User button 0
User button 1
User button 2
RADIOGRAPHY
head
pelvis
chest
Colored
B&W
HRZTEST
TEST110
TEST10
LEGO4
FIT10
CONTOUR 20
NT/PL 30,X
NT/PL 30,X:Y
UPDATE
STACK
GRADES
SCAN 60 age<--
my button
HISTO/LIST
NULL
Linear
1x1
2x2


/KUIP /MACRO /VECTOR /HISTOGRAM /FUNCTION /NTUPLE /GRAPHICS /PICTURE /ZEBRA /FORTRAN /NETWORK
/NTUPLE/SCAN
ACKNOWLEDGED
x_to_ps
Save on file
ENTRY | CATEGORY | DIVISION | NATION | AGE
2970 | 562.00 | 13.000 | 7.0000 | 29.917
2974 | 562.00 | 1.0000 | 7.0000 | 28.583
2975 | 560.00 | 3.0000 | 3.0000 | 29.500
2976 | 402.00 | 6.0000 | 7.0000 | 29.750
2983 | 565.00 | 12.000 | 7.0000 | 29.083
2990 | 463.00 | 12.000 | 7.0000 | 29.750
2991 | 341.00 | 8.0000 | 10.000
3028 | 415.00 | 14.000 | 7.0000
3041 | 432.00 | 1.0000 | 3.0000
3049 | 352.00 | 11.000 | 7.0000
KUIP/MOTIF > file parameters. Terminal Command : /NTUPLE/LIST
HELP
EXECUTE
RESET
SAVE
DISMISS
ACKNOWLEDGED
x_to_ps
/PLUMS
COMMAND
DISPLAY WINDOWS
CLEAR WINDOWS
DESTROY WINDOWS
X_TO_PS
SET COLORS
COLOR TABLE
User button 0
User button 1
User button 2
HELP
EXECUTE
RESET
SAVE
DISMISS
RADIOGRAPHY
HRZTEST
GRADES
my button
HISTO/LIST
Colored
B&W
CONTOUR 20
Logarithm
Linear
1x1
2x2


/KUIP /MACRO /VECTOR /HISTOGRAM /FUNCTION /NTUPLE /GRAPHICS /PICTURE /ZEBRA /FORTRAN /NETWORK
HELP
USAGE
MANUAL
EDIT
LAST
MESSAGE
SHELL
WAIT
UNITS
EXIT
QUIT
FUNCTIONS
/ALIAS
/SET_SHOW
STYLE
PANEL
COMMAND
APPLICATION
ROOT
TIMING
PROMPT
BREAK
COLUMNS
RECORDING
HOST_EDITOR
HOST_SHELL
VISIBILITY
NEW PANEL
SAVE PANEL
HELP
COMMAND
DISPLAY WIND
CLEAR WINDO
DESTROY WIN
X_TO_PS
SET COLORS
COLOR TABLE
User button 0
User button 1
User button 2
User button 3
[max Y LENgth of each menu item box]
[space available for the application]
[max Y length of space between menus]
[X or Y border for menus]
[Graphics workstation type] > 0
HELP
EXECUTE
RESET
SAVE
DISMISS
ACKNOWLEDGE
* /KUIP/SET_SHOW/STYLE [OPTION SGYLEN SGSIZE SGYSPA SGBORD
OPTION C 'Option' D='Y' R='?,C,AN,AL,G,GW,GS,GP'
SGYLEN R 'max Y LENgth of each menu item box' D=.25000
SGSIZE R 'space available for the application' D=.8 R=
SGYSPA R 'max Y length of space between menus' D=.2000
SGBORD R 'X or Y border for menus' D=.15000000E-01 R=0:
WKTYPE I 'Graphics workstation type' D=0
Set the user dialogue style (or working mode).
If OPTION='?' the current style is shown.
The startup value is C (command mode).
Currently available options are:
C for Command
AN for general Alpha menu (with Numbers)
AL for general Alpha menu (with Letters)
G for Graphics menu (with hardware character fonts)
GW for Graphics menu (with shadowed Width effect)
GS for Graphics menu (with Software character fonts)
GP for Graphics menu (with Panel keys only, i.e. no comm
RADIOGRAPHY
HRZTEST
GRADES
TEST110
NT/PL 30.X
SCAN 60 age<--
HISTO/LIST
TEST10
NT/PL 30.X%Y
NULL
LEGO4
UPDATE
FIT6
NT/SCAN 30...
Logarithm
SURF20
STACK1
CONTOUR 20
1x1
2x2


/HELP /MACRO
set vsiz .6
h/plot 110
h/plot 110
HELP
EXECUTE
RESET
SAVE
DISMISS
QUIT PAW
CUSTOMIZE
RECALL
MAIN PANEL
NEW PANEL
SAVE PANEL
HELP
SET COLORS
COLOR TABLE
User button 0
User button 1
User button 2
User button 3
Histogram 110
From bin 29
To bin 66
Integration 2543
Normalize I 25.43
175
150
125
100
75
50
25
.2
.4
.6
.8
1
TEST OF HBOOK1
RADIOGRAPHY
HBZTEST
GRADES
TEST110
TEST110
HIST110


/HISTOGRAM/LIST_1_popup
DISMISS
List of Objects in :
20 (2)
4 (2)
350 (1)
30 (2)
14 (2)
1 (2)
4 (2)
11 (2)
-14 (2)
-1 (2)
-11 (2)
2 (2)
-5 (2)
12 (2)
-2 (2)
-12 (2)
3 (2)
-6 (2)


OPAL
GEANT3.15
Exploded view
Sep 16
Mon

"Fermilab View of HEPLIB"

I. Gaines

HEPLib: Possibilities and Potential
A FNAL "Management" Point of View

Irwin Gaines
Sept 20, 1991

Areas for HEPLIB Concern
(what software it deals with)

- Topics for Software:
  - Application Code Libraries (Geant/Cernlib/mathlib...)
  - Utility Code Libraries (Graphics, Data/Memory Managers,Int analysis...)
  - Code Development Tools (Code Management, SW Engineering,...)
  - Operating Systems & User Environments (HEPVM, HEPiX next week)

- Origin of Software:
  - Commercial SW?
  - A few major SW centers?
  - Major Experiments?
  - Anyone in the HEP community?

- Quality of Software:
  - Fully tested and certified?
  - Anything that is submitted (let the buyer beware; cf. anonymous ftp in the worldwide UNIX community)

- "Customers" for Software
  - National Labs
  - Universities
  - "Anyone" doing HEP
  - Wider community

- Regulatory bodies
  - National vs International distribution
  - Licensing

- Centralized vs Distributed

- User Group vs Organization

Models for HEPLIB
(how it deals with software)

- User group meeting, lots of interesting talks (CHEP-like)

- Communication center
  Generate, maintain and distribute lists of software and availability

- Clearing house
  distribution center, "one-stop shopping", no real judgement

- Software Center
  testing
  porting
  product evaluation
  unified distributions

- Software Development Center
  one of several centers that write/develop/implement products
  coordinate approaches (insure all needs are being addresses without lots of duplication
  develop needs/requirements/specs

- Collaborative efforts
  promote and support inter-lab efforts on HEP software projects

- Standards manager
  represent HEP on standards bodies, committees
  develop and promote community standards

- Centralized control

Ma(r)ximalist Approaches

- HEPLIB: The world center for HEP Software
- Formal SW projects: From user provided specifications, all software projects are formally initiated, managed and reviewed.
- Center is responsible for all HEP software development, distribution and management.
- Quality assurance: from formality of specification and review process]
- Integration: from centralized operation

No, clearly, this is not the way our community works. But may be worthwhile to understand which of the strong points of this approach can be recovered in a less autocratic environment.

Minimalist Approaches

- Meetings
- Communication Center
- Clearing House

All of these take place to some extent already (CHEP and other computing oriented conferences, listserv, CERN and FNAL software distributions), and could benefit from some formalization. In particular procedures on how software is submitted to the "archives" and what are the criteria for acceptance and re-distribution would be useful, as would cataloging facilities that go beyond a single lab/continent. Distribution can be from a single source, but more usefully could be from a distributed series of sites each responsible for different products or ports of a single product (cf. anonymous ftp on uunet). A further step would be to agree on testing and certification procedures and to distinguish between "tested" and "use-at-your-own-risk" software (there is probably a need for both types!!).

Aggressive (but realistic?) Approach

- An HEP "Software Development Center"
  - An organization, not just a user group
  - Distributed, not centralized (not a geographic "center" but a logical entity)

- Goals and functions of the SW Center
  - All of the cataloging and distribution functions described above;
  - Coordination of efforts: flag duplications
    - insure that requirements exist (and are understood by sw developers
    - find "volunteers" for needed projects
  - Promote collaborative efforts between institutions

- Participation is voluntary, the proponents of the center act as consultants and coordinators

- Could eventually become respnsible for computing "foreign policy"
  - Negotiations with vendors
  - Standards committees

- Start with application SW and broaden spectrum as center catches on

Potential Roadblocks

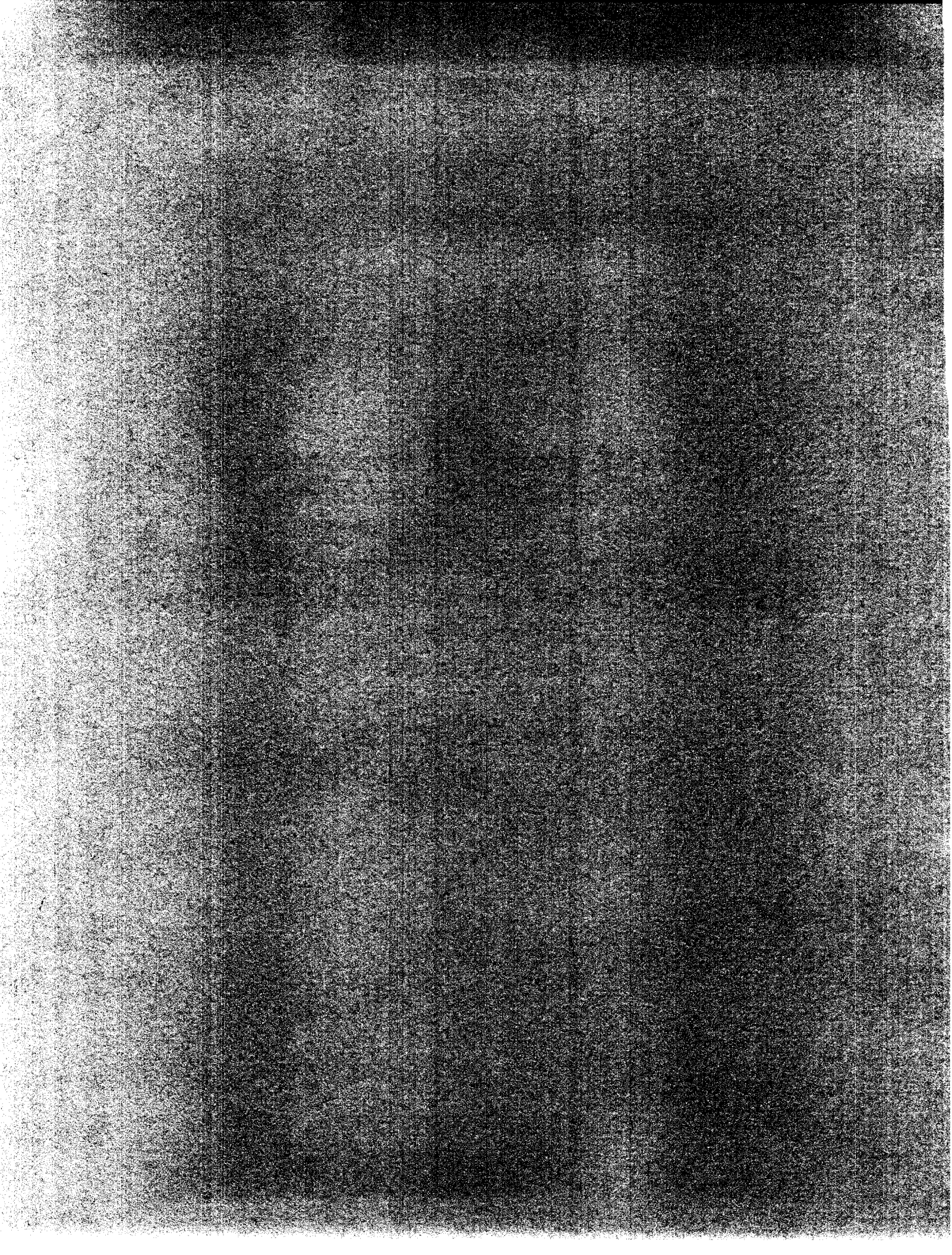
- Funding
  - What manpower is required
  - Do separate institutions want to participate
  - Will they commit real funds/people

- Customer Acceptance
  - What do users want
  - What do users need
  - What are users willing to accept

- Politics
  - International distribution of software?
  - How do DOE labs distribute software? National Software Center?
  - Licenses, site-licenses, collaboration wide licenses,...
  - Use of commercial or semi-commercial products

"GEANT Version 3.15 and Beyond"
"Graphics and CAD Interface"
"The New Geometry Package"
"Standard Interface to Event Generators (LHC/SLUG)"
"The FLUKA/DTUNUC Interfaces and Other Physics"
"Improvements"
"Parametrization"

R. Brun, F. Carminati




HPHW
CERN
European Laboratory for Particle Physics
COMING...
GEANT
VERSION 3.15

Hadronic Physics

- Interface with GHEISHA 8
- Interface HADRIN/NUCRIN
- Interface with FLUKA in preparation (CERN/Helsinki)
- Improvement of the interface GEANT/hadronic showers

FLUKA

σ^{inel}

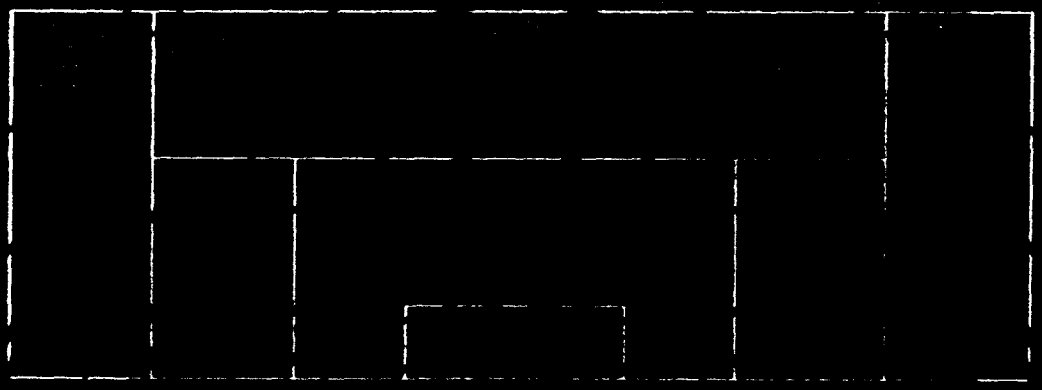
= 0	E_{kin} < 50MeV
data+extrapolation	E_{kin} < 100GeV
parametrization	E_{kin} < 10TeV
log extrapolation	E_{kin} > 10TeV

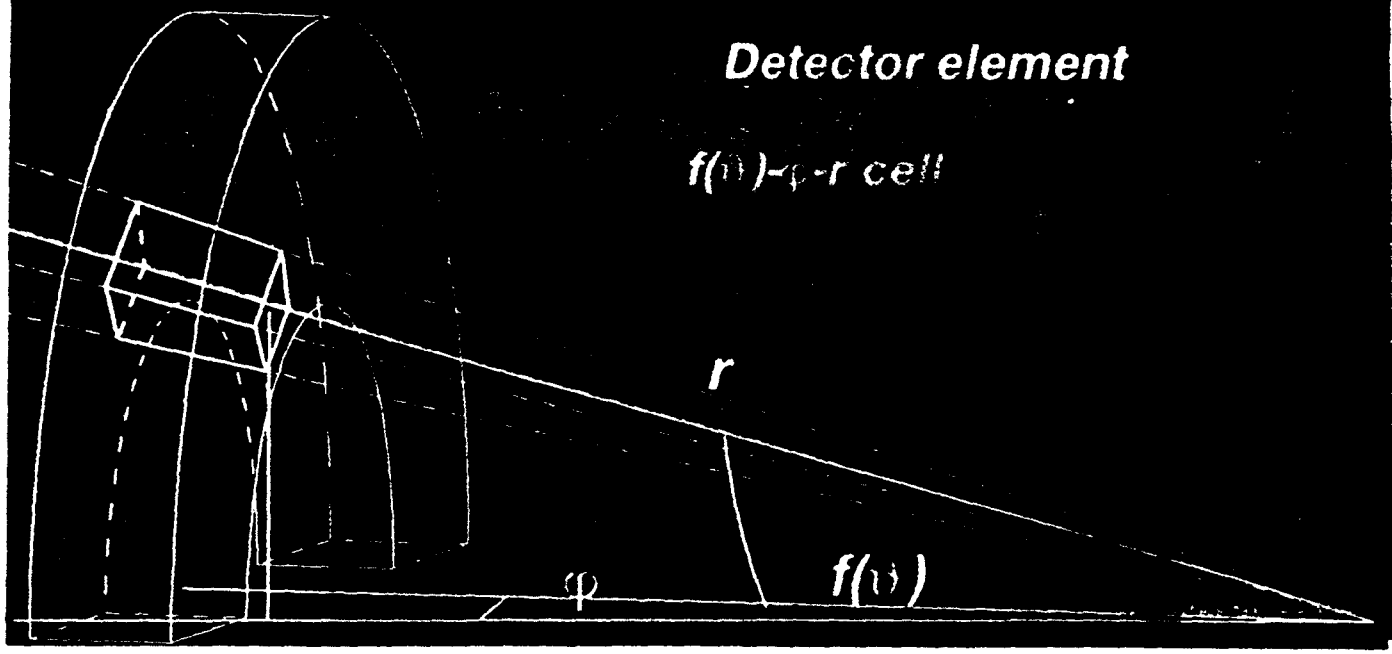
- σ^{el} parametrization introduced in 1988 by Moehring
- p<5GeV: known cross sections (2%) for resonances and decay (1%)
- p>5GeV: fragmentation of q-aq ... multip. collision q-aq sea, hard diff

FLUKA

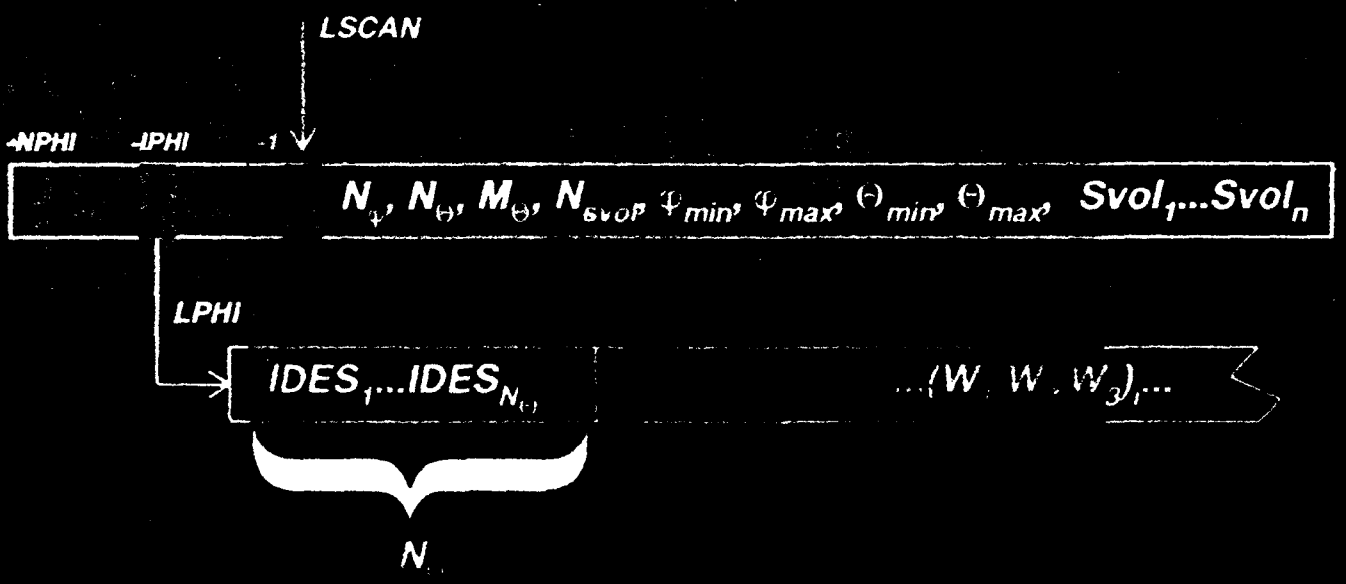
- Fermi momentum taken into account
- Excitation energy computed from a detailed balance taking into account binding energies and so on...
- Nuclear evaporation from tabulated binding energies
- Very actively developed/maintained by a Milano-Helsinki-CERN-Leipzig... collaboration

Scan geometry





Scan Data Structure



M_Θ = Mode of Θ divisions (Θ, cos(Θ), η)

$Svol_i$ = Name of scan volume i

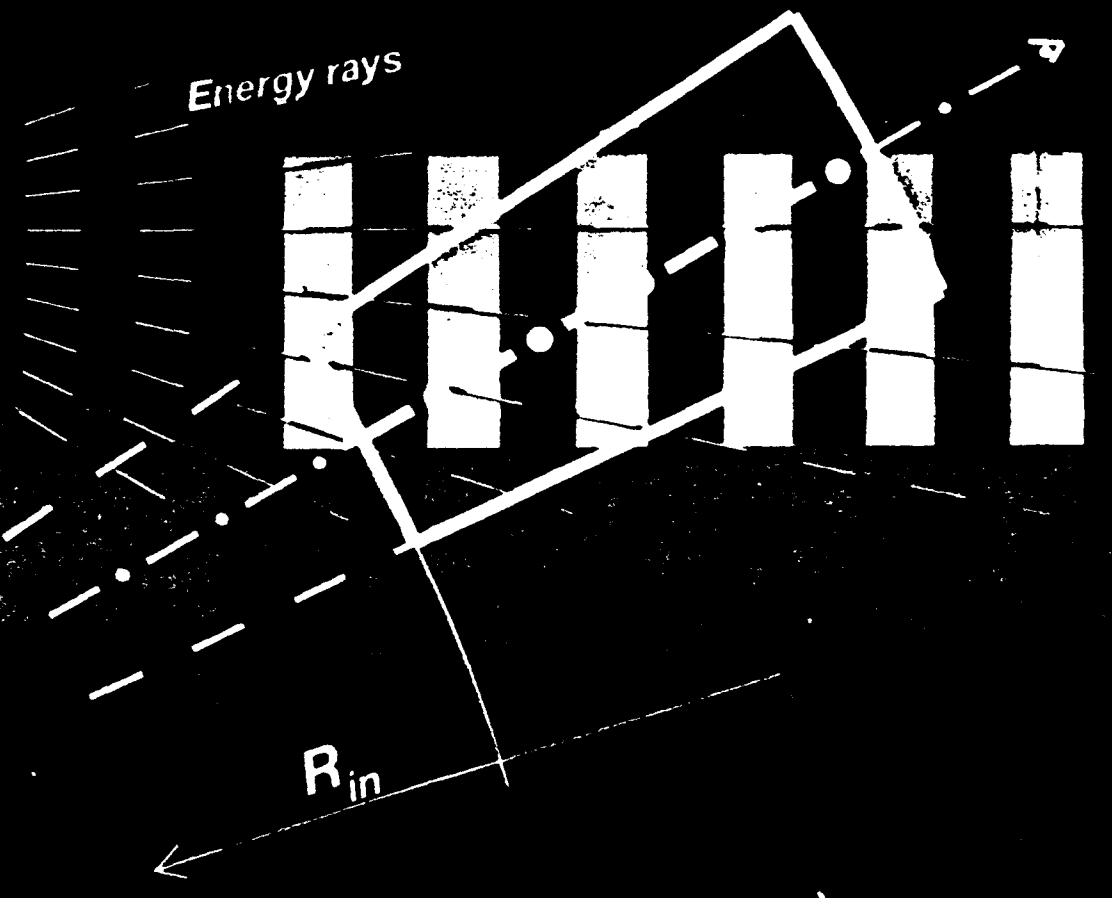
$IDES_i$ = { Pointer to first W_1, W_2, W_3 triplet; Number of triplets (segments in R) }

$(W_1)_i$ = { Volume identifiers packed (NUMBV); Hadronic sampling fraction }

$(W_2)_i$ = { Inner radius of R segment; Detector number; Electromagnetic sampling fraction }

$(W_3)_i$ = { Average absorbtion length; Average radiation length }

Cell parameters



$$\text{Samp. Frac.} = \frac{\sum}{\sum + \sum \int dx}$$

$$HIT = I_{\Theta}, I_{\varphi}, I_{r}, I_{track}, I_{para}, E_{loss}$$

Scan parametrization

- JSCAN automatically generated
- Direct I/O via RZ (part of INIT)
- Integrated with the GEANT HITS/DIGI
- SCAN set and detectors automatically created at initialization time
- Different cuts for γ, e, h, n, μ
- Tracking of primaries in real geometry and energy rays in scan geometry
- Θ-φ energy flow

Parametrization times

$$T_{event} = T_{GEL} + T_{GPI} + T_{PAR}(N_{PRIM} + N_{SEC})$$

T_{GEL} = Total time to track all electromagnetic. particles

T_{GPI} = Total time to track all other particles

T_{PAR} = Time to parametrize one track = $T_{GEOM} + T_{SPOT} + T_{CHAN}$

= 10 + 15 + 10ms (Let us assume 30ms)

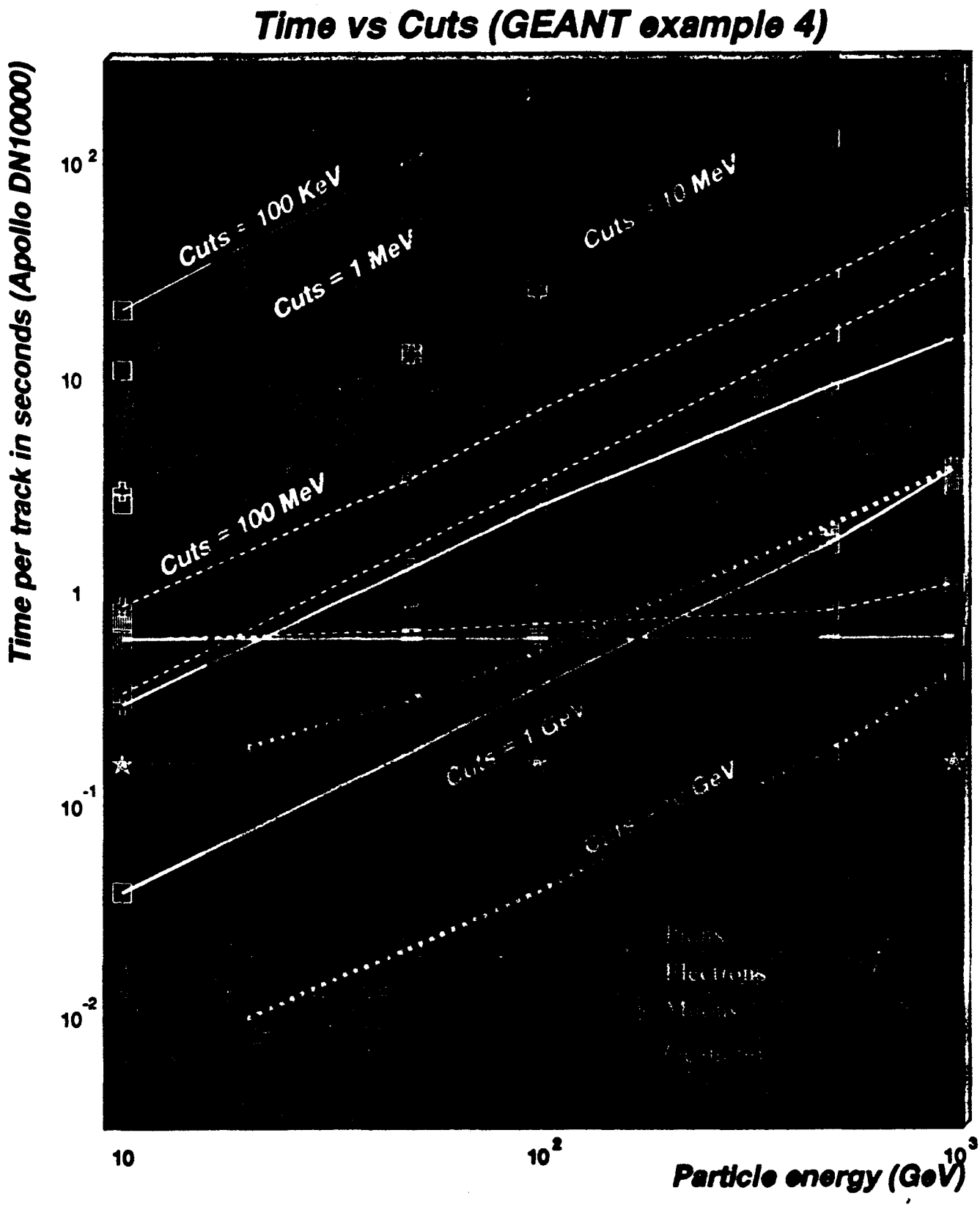
(See GFLASH reference H1)

N_{PRIM} = Number of primary tracks (assume 200 at LHC)

N_{SEC} = Number of secondary particles below ECUT

Time estimate for LHC in seconds (40 MIPS)

ECUT	Full Geometry	SCAN Geometry (Full + SCAN)
1 GeV	$25 + 250 + 3.10^{-2}(200 + 10000)$ = 580	100 (335)
10 GeV	$2 + 35 + 3.10^{-2}(200 + 1000)$ = 75	15 (42)
100 GeV	$0 + 4 + 3.10^{-2}(200 + 100)$ = 14	4 (5)
1 TeV	$0 + 0 + 3.10^{-2}(200 + 10)$ = 6	1 (1)


Time vs Cuts (GEANT example 4)
Time per track in seconds (Apollo DN10000)
10 2
10
1
10-1
10-2
Cuts = 100 KeV
Cuts = 1 MeV
Cuts = 10 MeV
Cuts = 100 MeV
Cuts = 1 GeV
Cuts = 10 GeV
Electrons
10
10 2
10 3
Particle energy (GeV)

New geometry package

- Constructive solid geometry with history of creation
- Basic and generic shapes
- Boolean operations
- Binary tree
- Second order surfaces
- Detector database
- Collaboration with Serpukhof

General idea

- 'Constructive' approach
- Basic blocks similar to CAD/CAM
- Extended Boolean operations allowed
- Binary Decision Tree for tracking built at initialization time
- Surface description for 3D graphics
- Complete reorganization of the JVOLUM data structure
- Complete backward compatibility

Data-base elements

- Detectors are made of volumes
- Volumes are made of shapes
- Shapes are basic or general
- Basic shapes are
  - Box
  - Cylinder with hole
  - Cone with hole
  - Spheroid
  - Paraboloid
  - Trapezoid

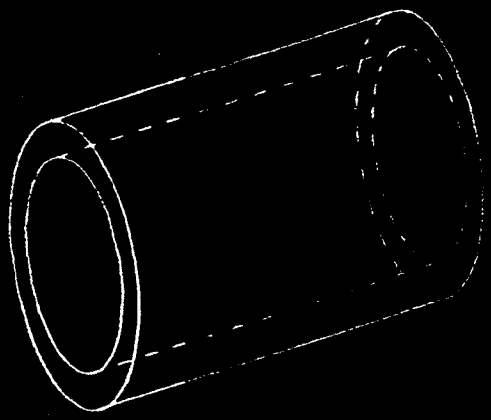
Data-base elements

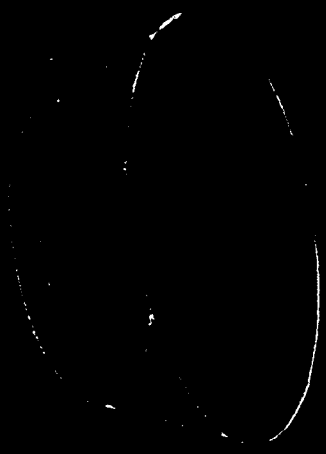
- General shapes are:
  - General prism
  - Body limited by plane faces
  - Convex body
  - Body of revolution
- To these we could add
  - Kinetic body
  - Body defined by a set of slices and faces

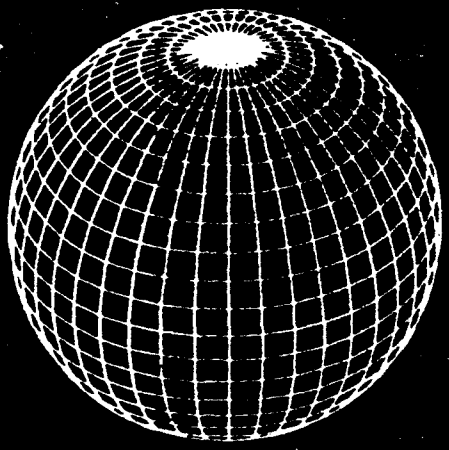
Basic shapes

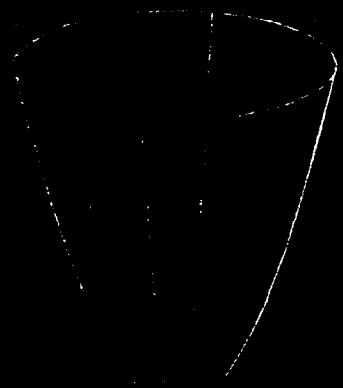




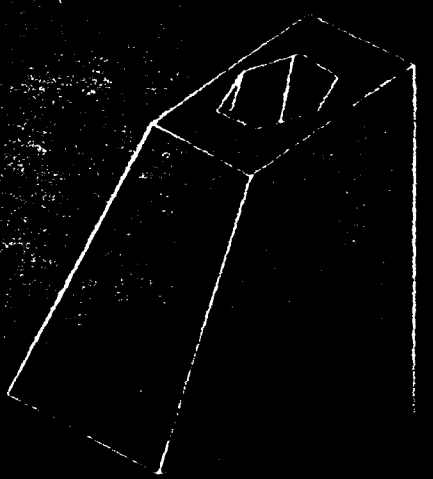








Generic shapes



General prism

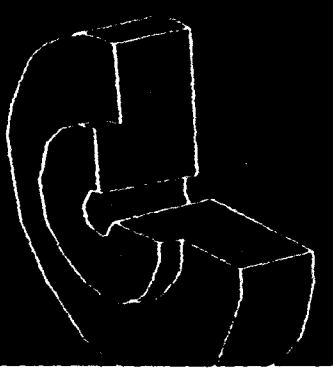


Body limited by faces



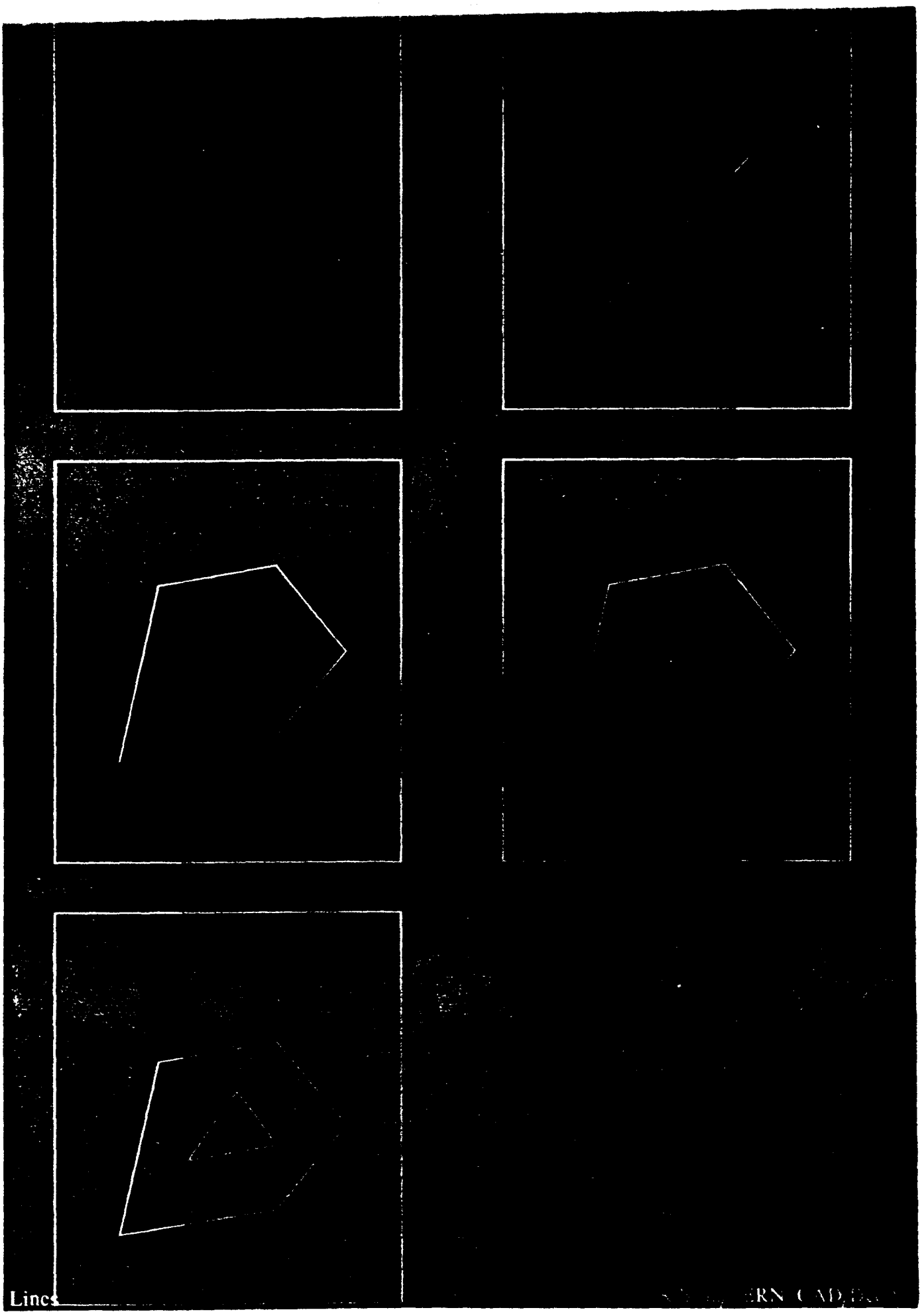
Convex body

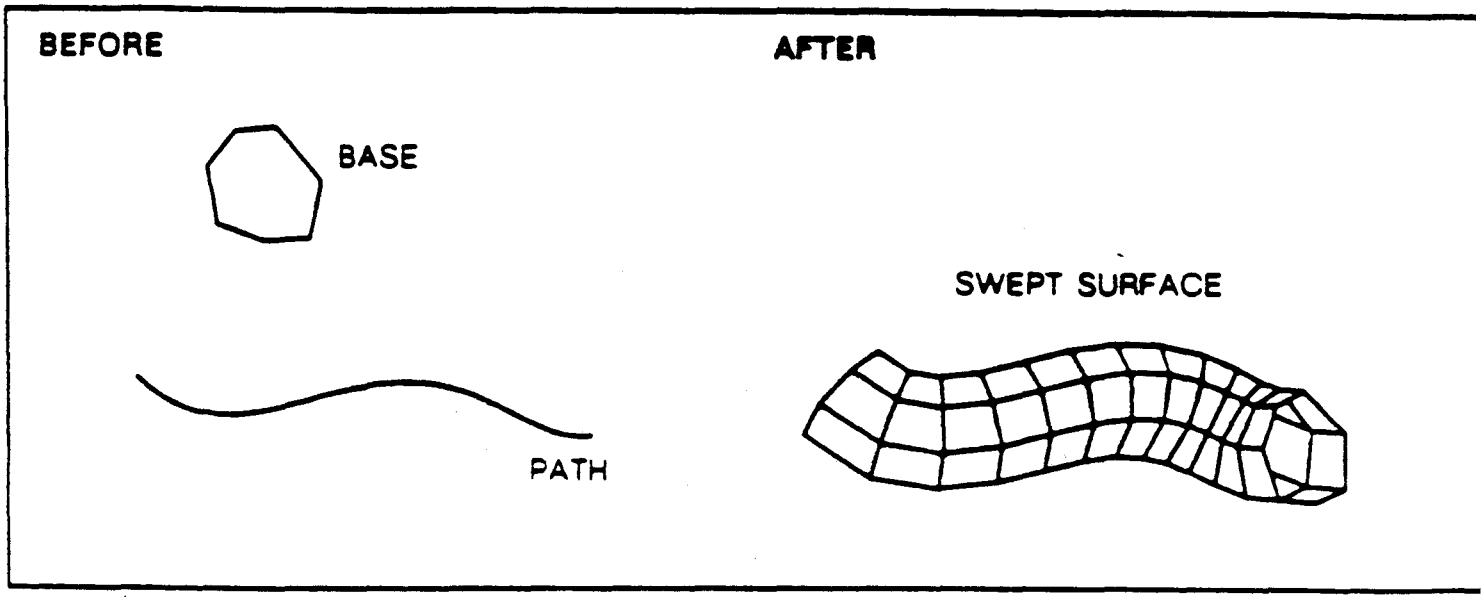
Continuous revolution

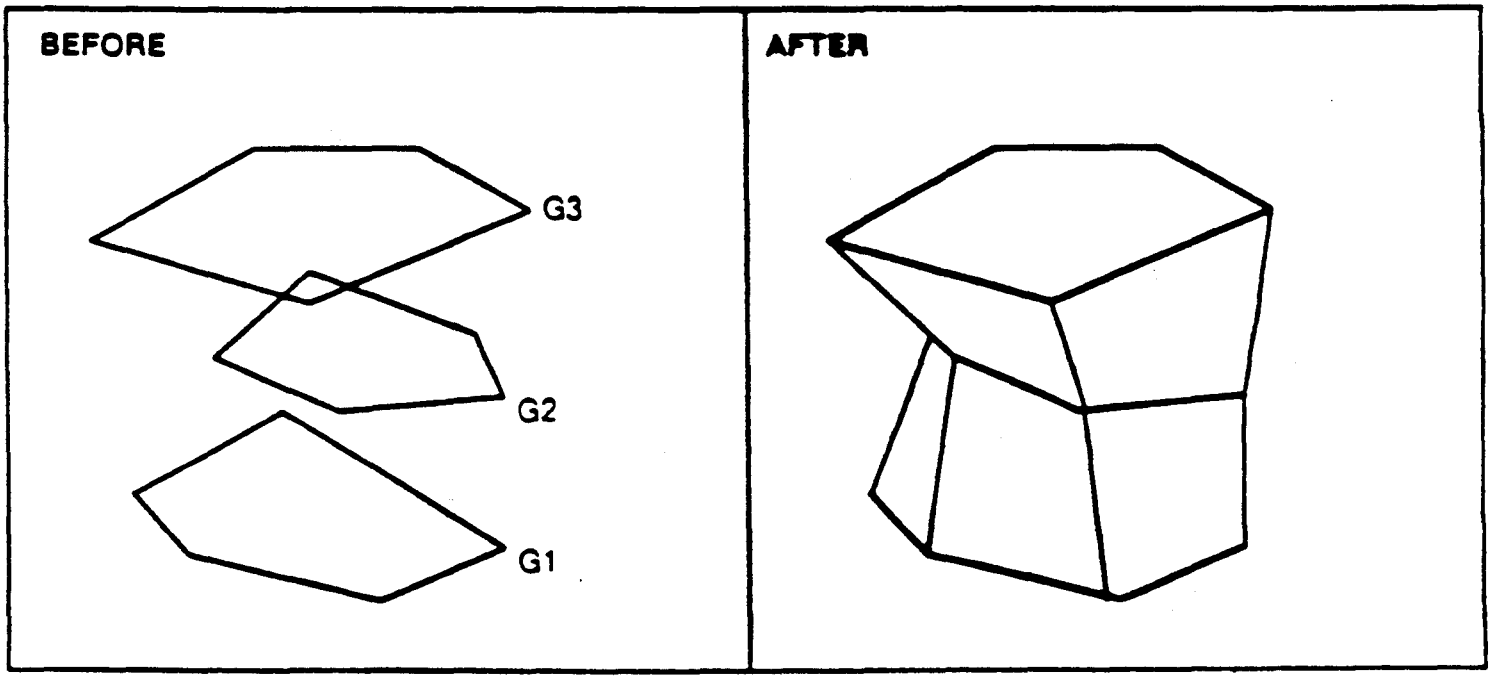


Discrete revolution




Lines


BEFORE
AFTER
BASE
SWEPT SURFACE
PATH


BEFORE
AFTER
G3
G2
G1

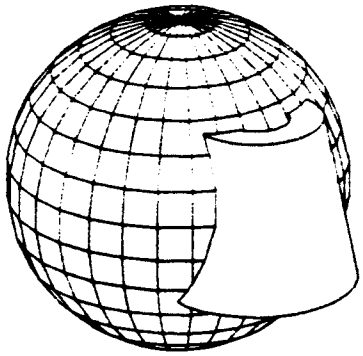
Operations on shapes

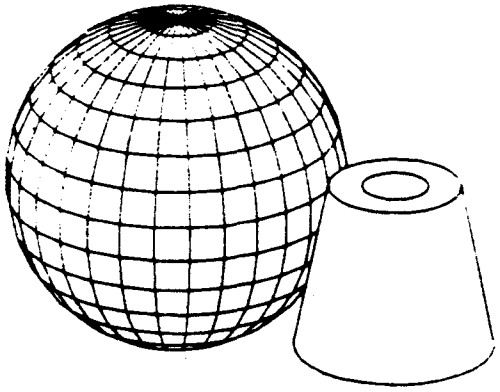
- Boolean operations possible among shapes:
  - OR = sum
  - AND = intersection
  - - = subtraction
  - GLUE = concatenation
- Some shapes can be divided with equal size steps
- 3D-2D symmetric divisions will be provided (honeycomb)

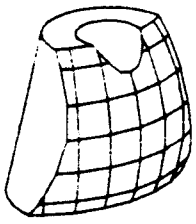
Transformations

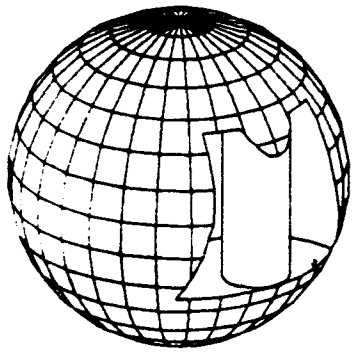
- General shape transformations are:
  - Translation
  - Rotation
  - Reflection
  - Scale transformation
  - Affine transformation
- General transformations can be combined with boolean operations

1:2+2:(4:5)\ 6:1





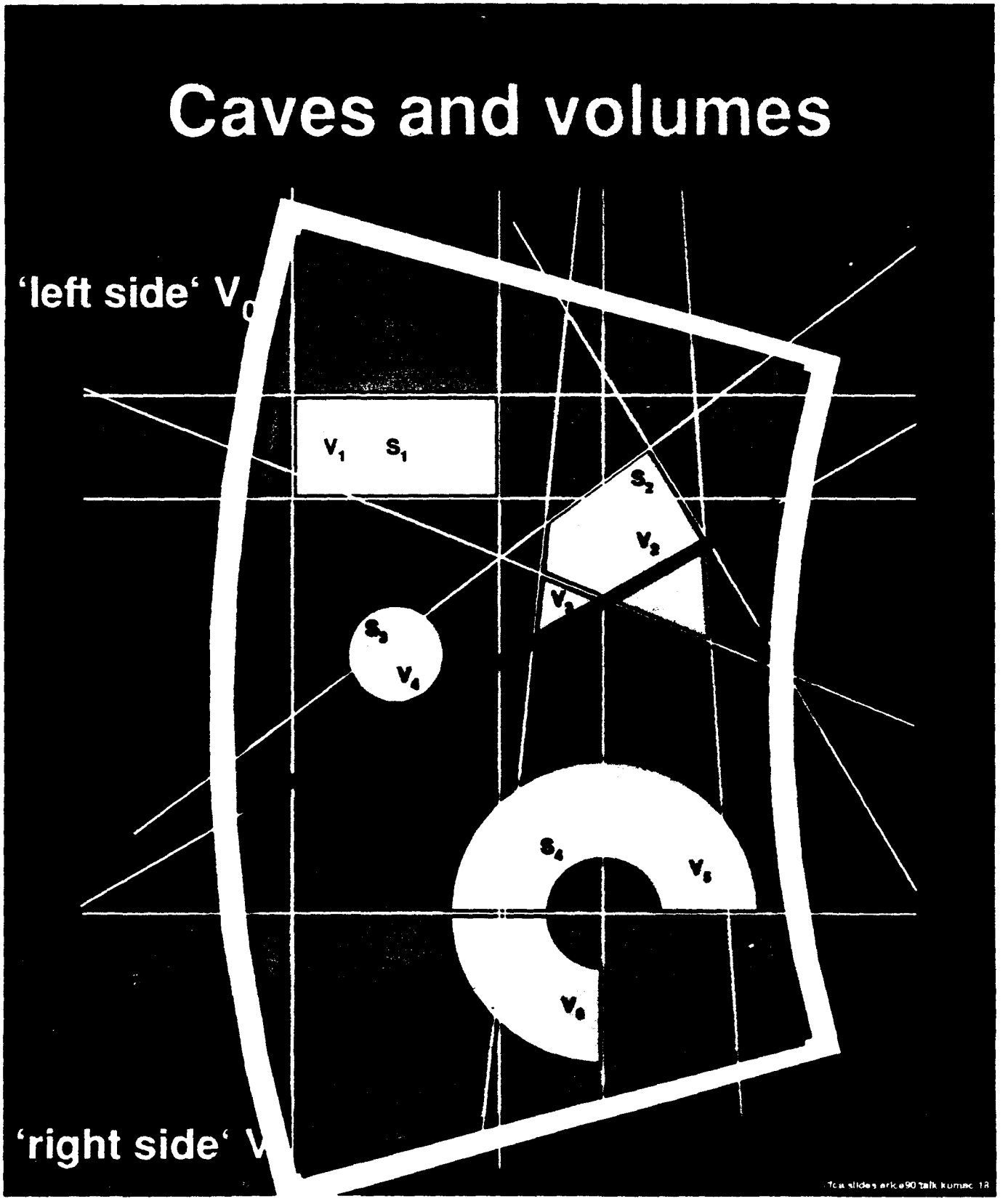




Caves

- Cave V_i limited by 2nd order surf $S_{ij}(P)$
- If for every j, $S_{ij}(P) > 0$, P is in V_i
- Shapes may be split in caves
- S_{ij} divide the space in subspaces branches of the binary tree
- S_{ij} chosen to maximize the volumes rejected at every branch
- At initialization time shapes are split into caves and optimum tree is built

Caves and volumes

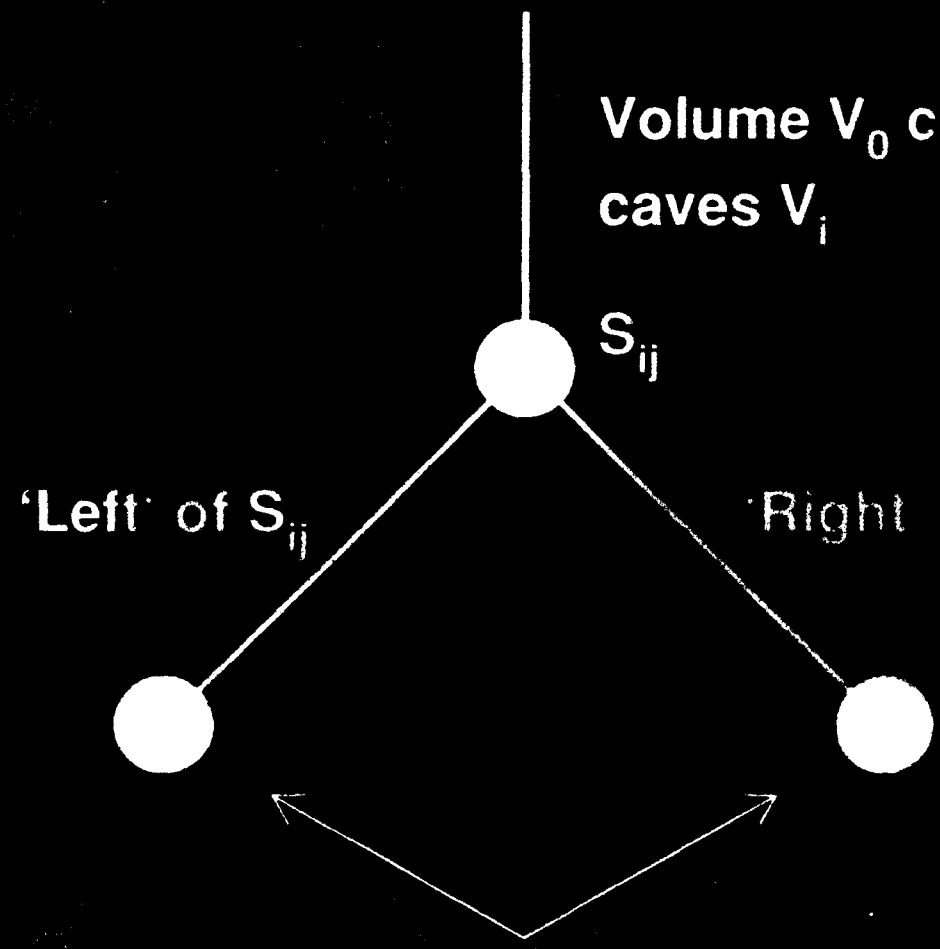


Where am I? - binary tree



- A binary decision is taken at every node
- Time prop to log$_2$ of number of volumes
- Used by GSORD to find the cell
- Best algorithm BUT a good binary tree must be built

Tracking Binary Tree



Sub-volumes containing many caves or a single one or none

Points and surfaces

$$S = A_0 + A_x x + A_y y + A_z z +$$
$$A_{xx} x^2 + A_{yy} y^2 + A_{zz} z^2 +$$
$$A_{xy} xy + A_{yz} yz + A_{zx} zx +$$

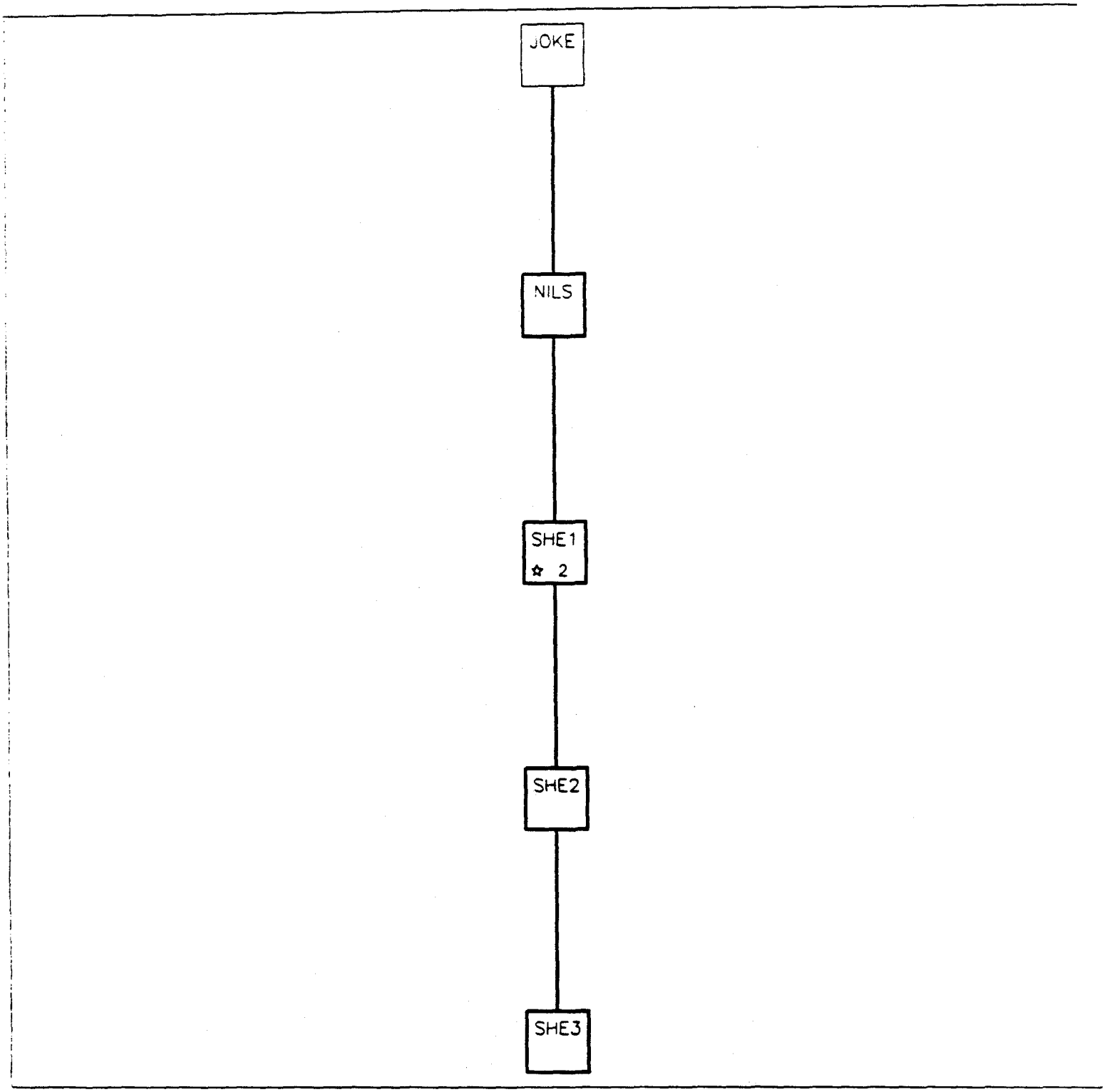
$S(P) > 0$

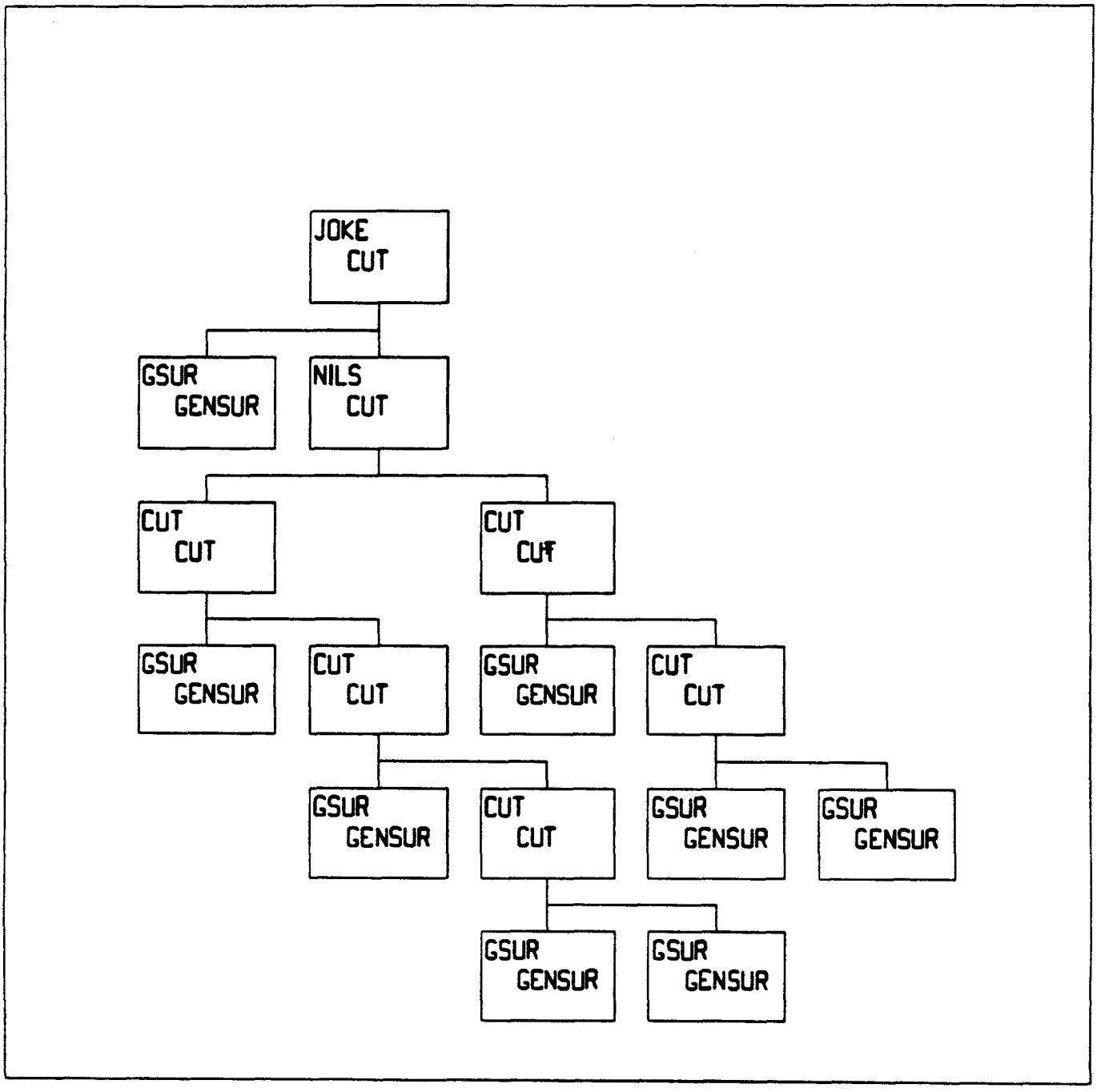
$S(P) = 0$

$S(P) < 0$

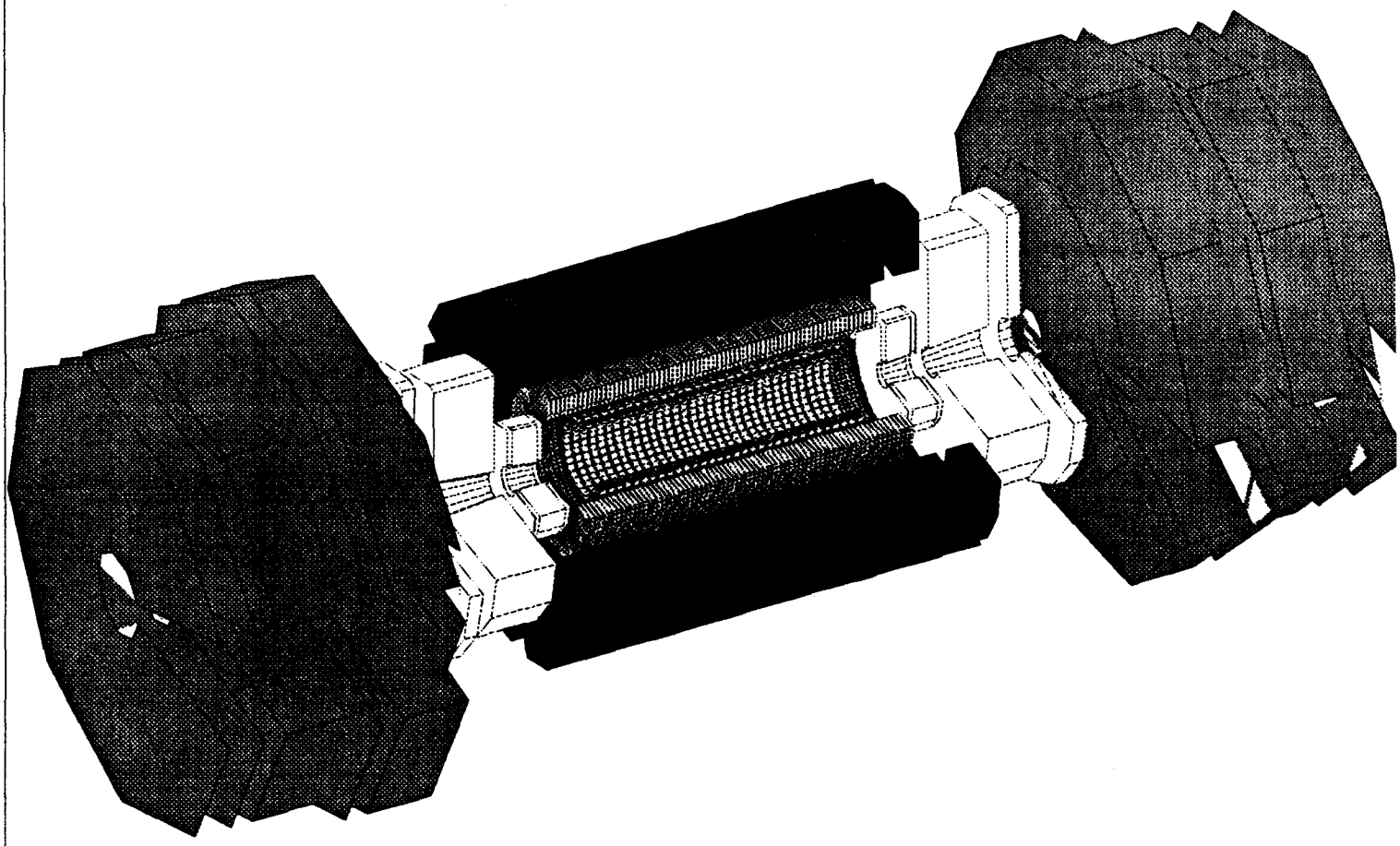
Tracking questions

- The questions to be answered by the tracking package are:
  - Where am I?
  - What's the shortest distance to the boundary (SAFETY)
  - What's the distance to the boundary along the track (SNEXT)?
- The distance along a helix can also also be interesting (not in GEANT now)


JOKE
NILS
SHE1
☆ 2
SHE2
SHE3

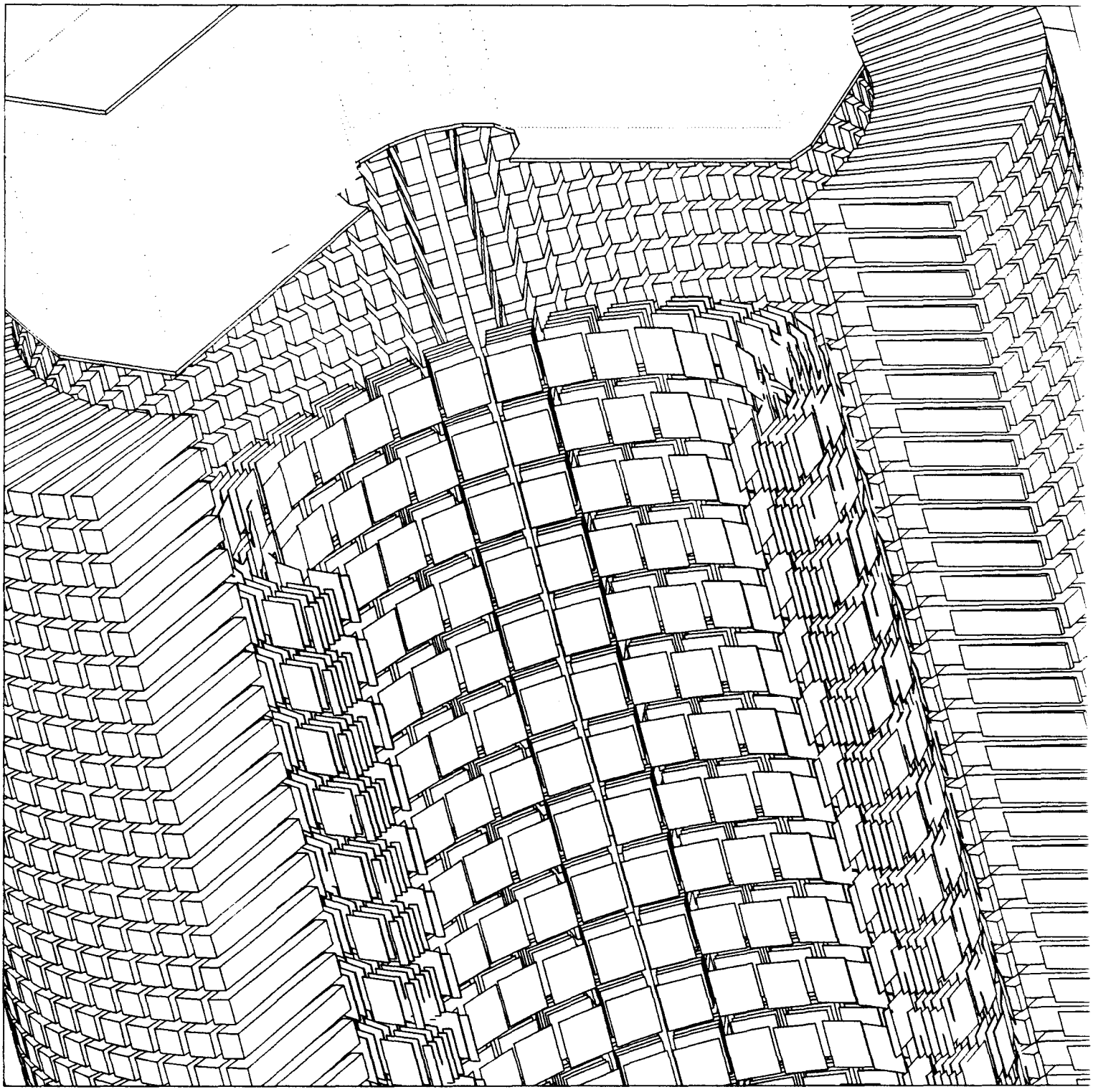

JOKE
CUT
GSUR
GENSUR
NILS
CUT
CUT
CUT
CUT
CUT
GSUR
GENSUR
CUT
CUT
GSUR
GENSUR
CUT
CUT
GSUR
GENSUR
CUT
CUT
GSUR
GENSUR
GSUR
GENSUR
GSUR
GENSUR

EAGLE

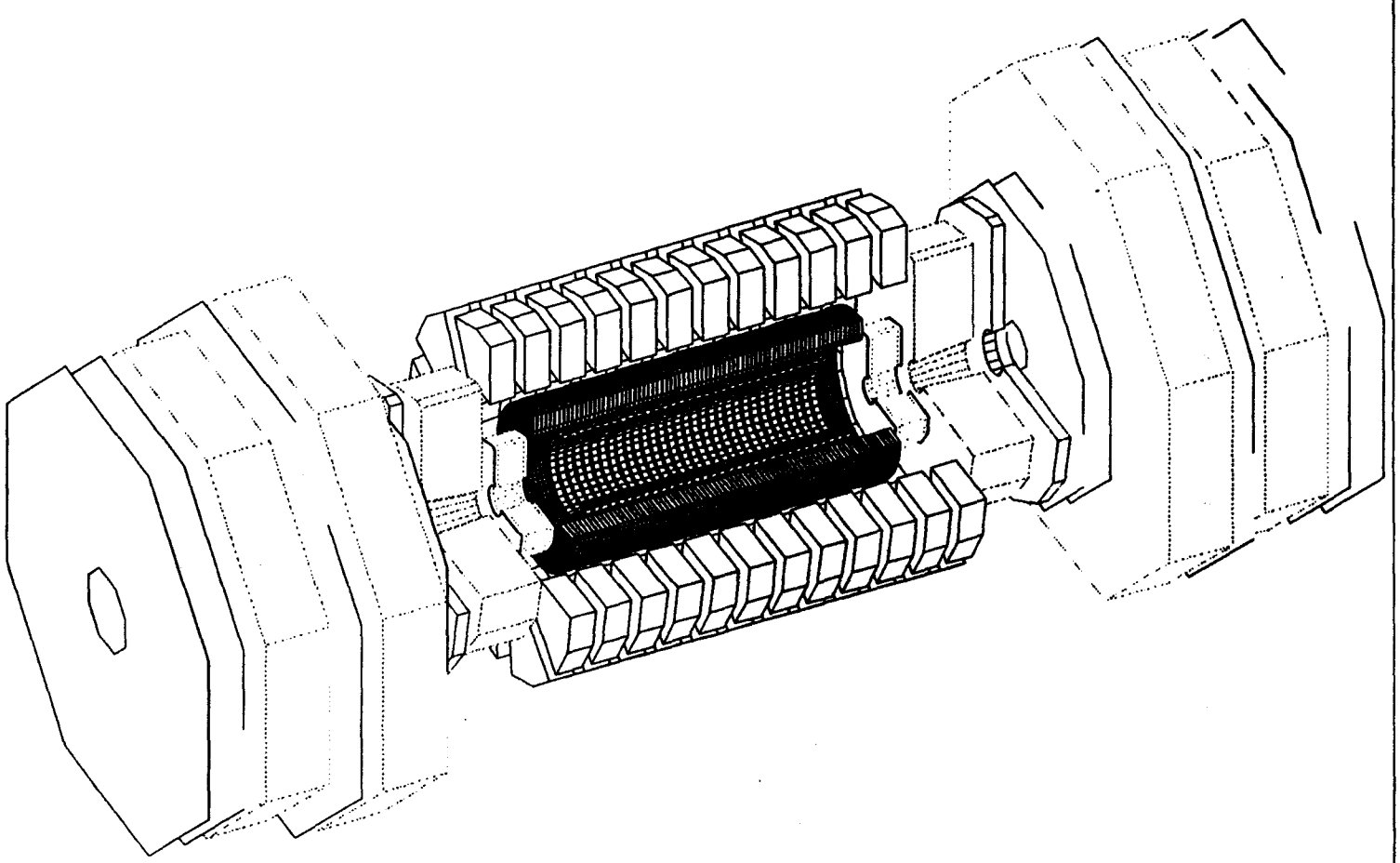


GEANT3.15 Exploded view

CONE
ENDE
EECI
EECO
ENDH

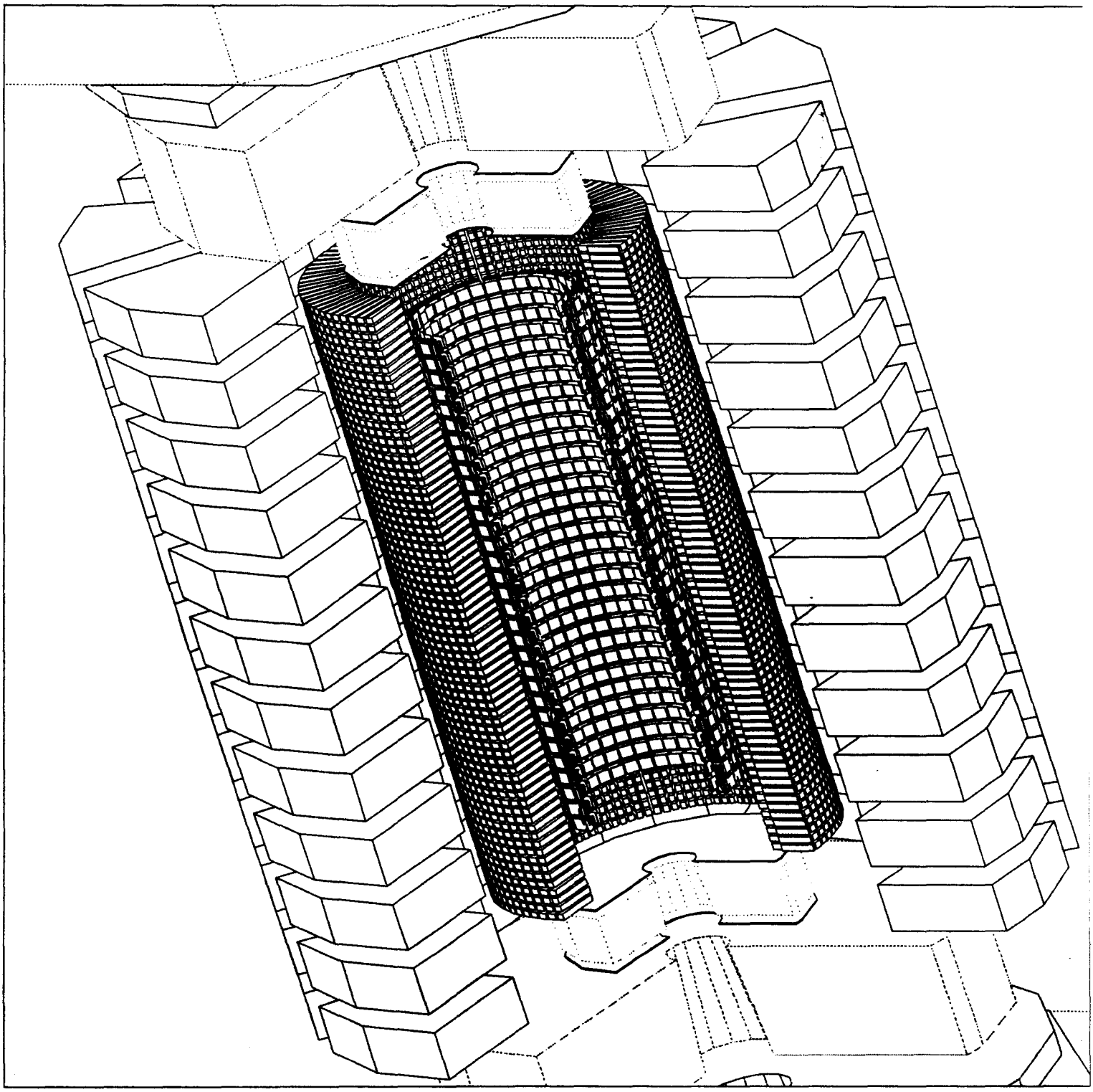


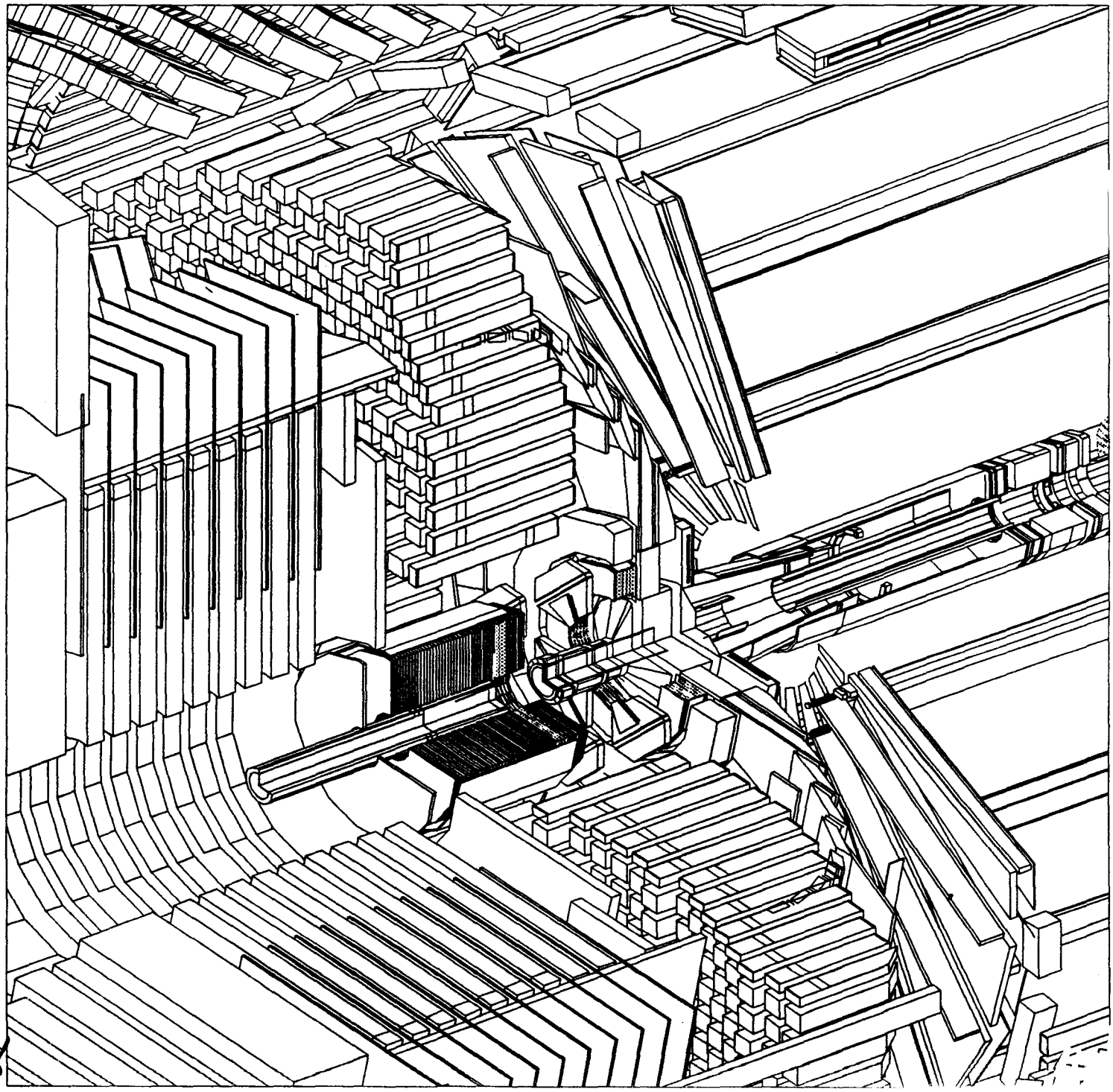
EAGLE

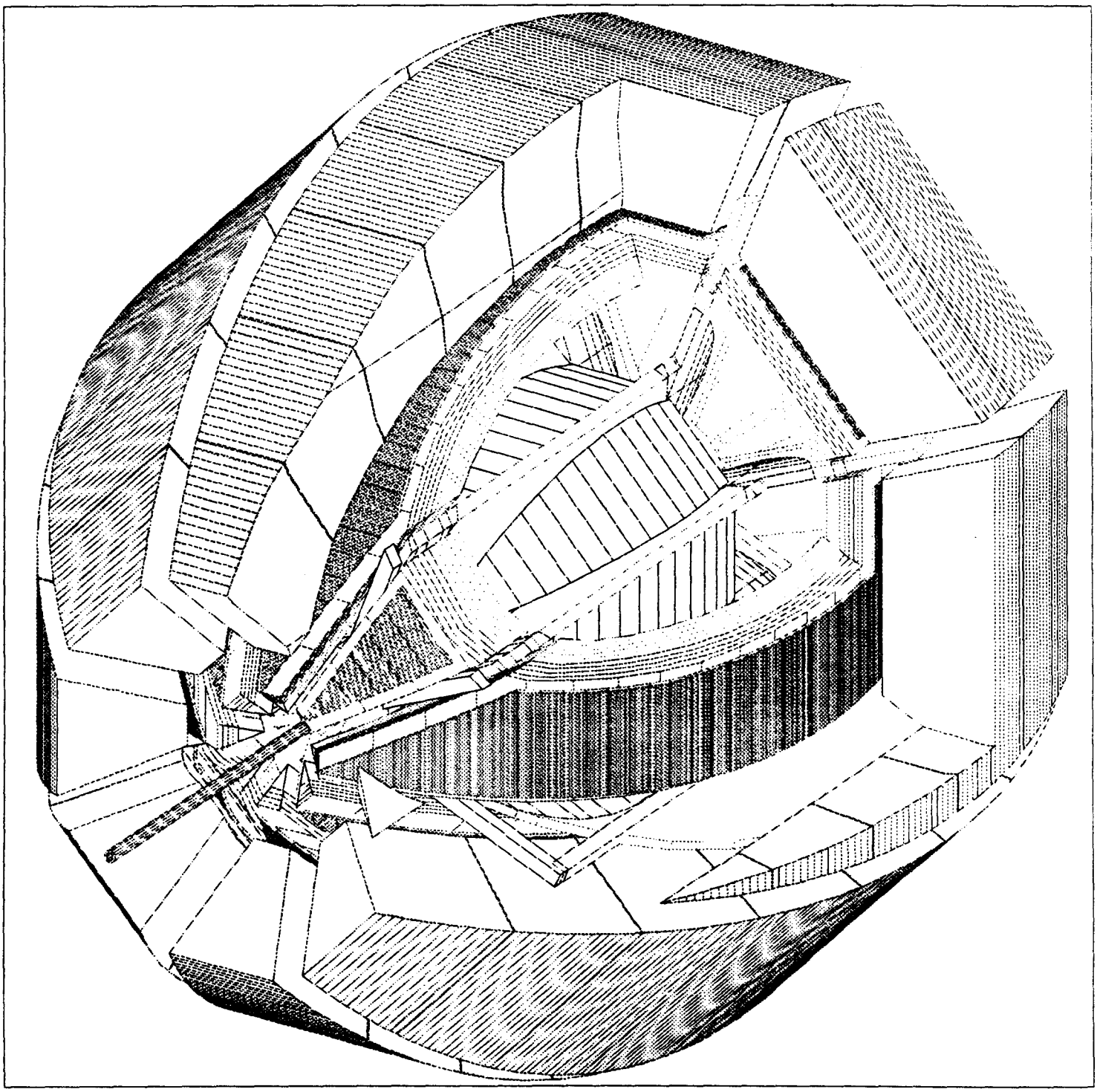


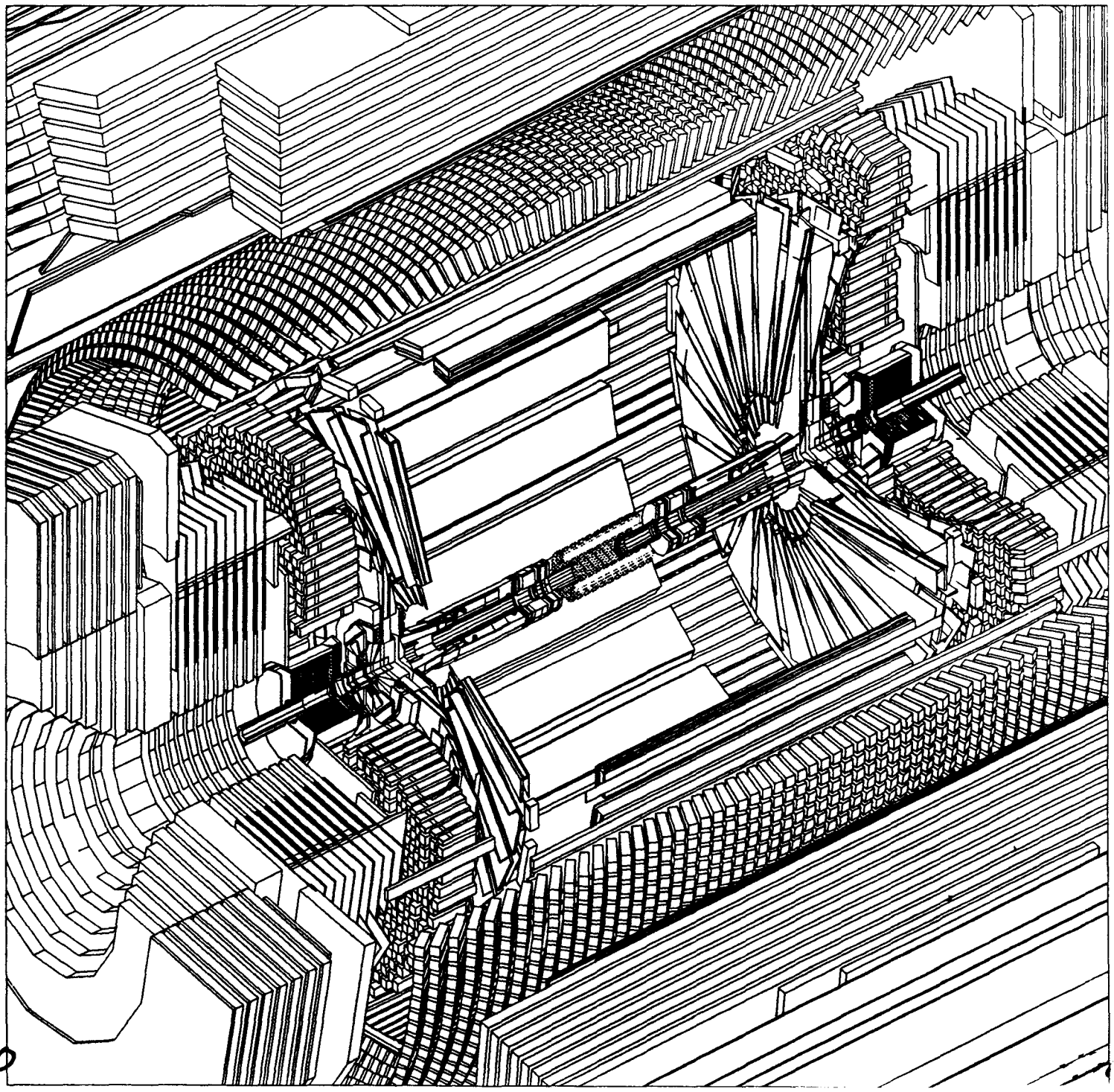
GEANT3.15 Exploded view

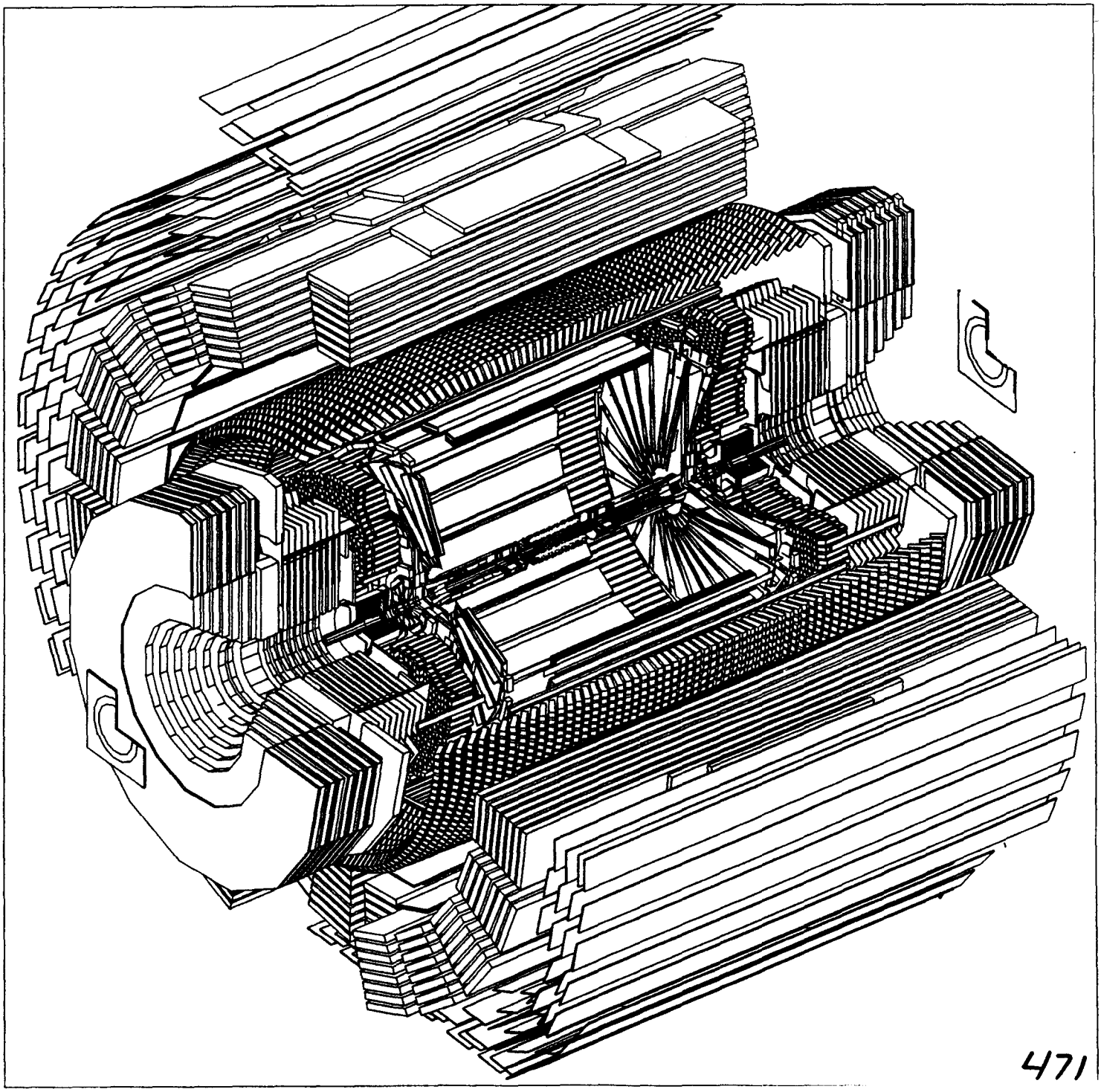
4660

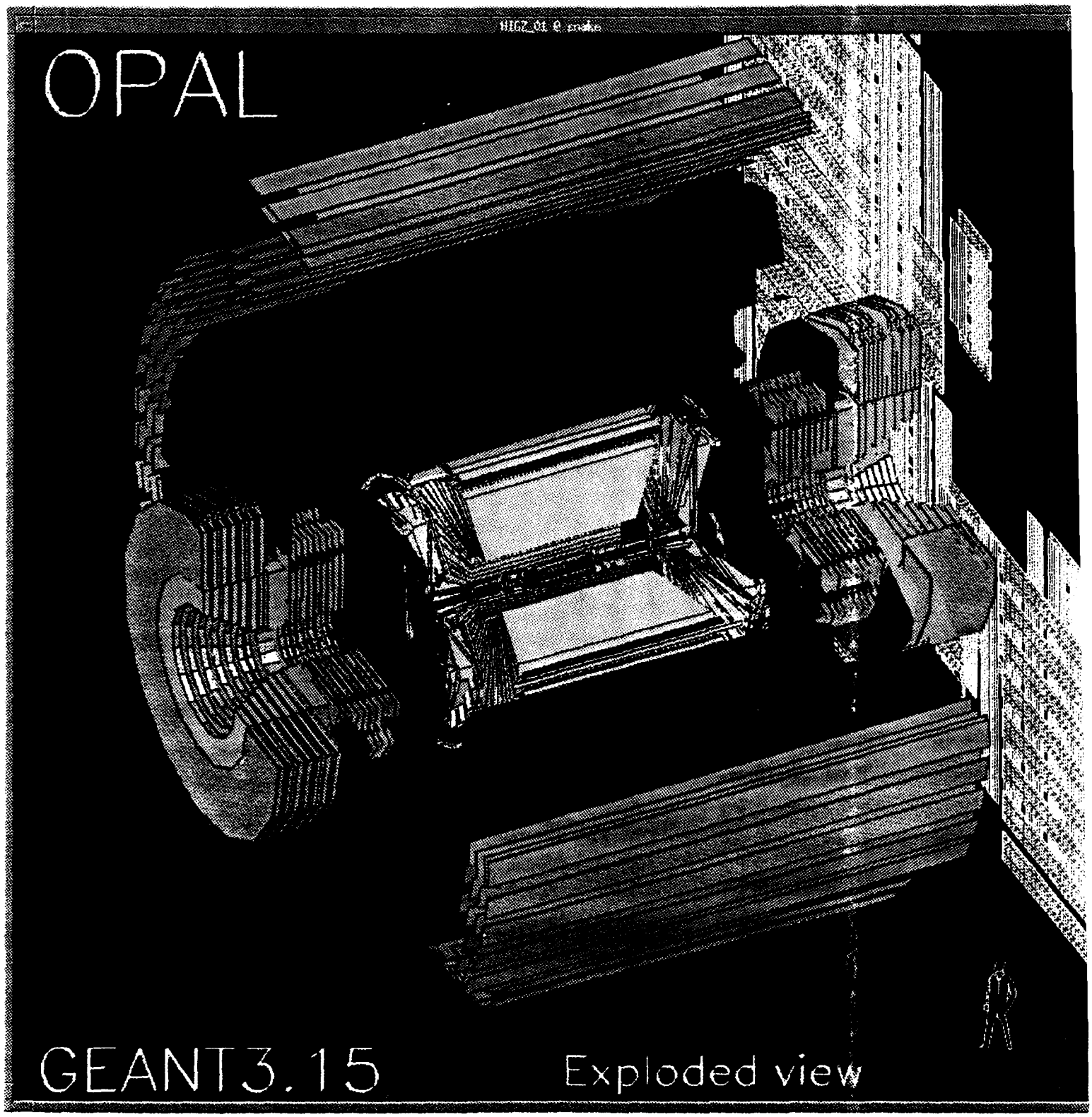






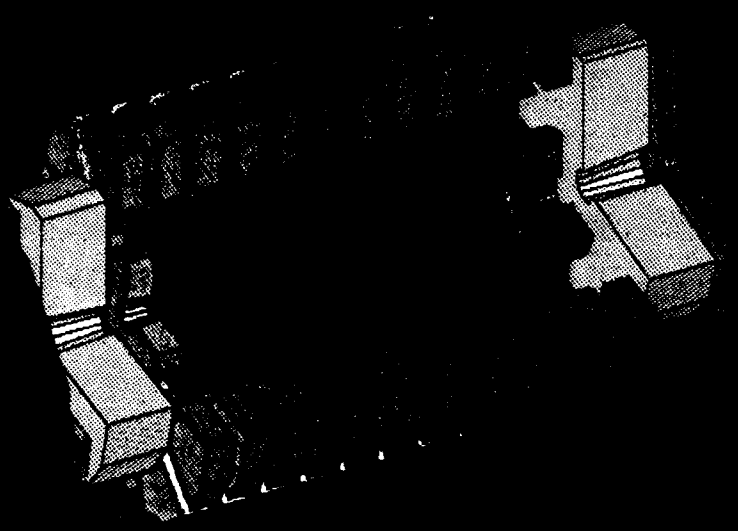




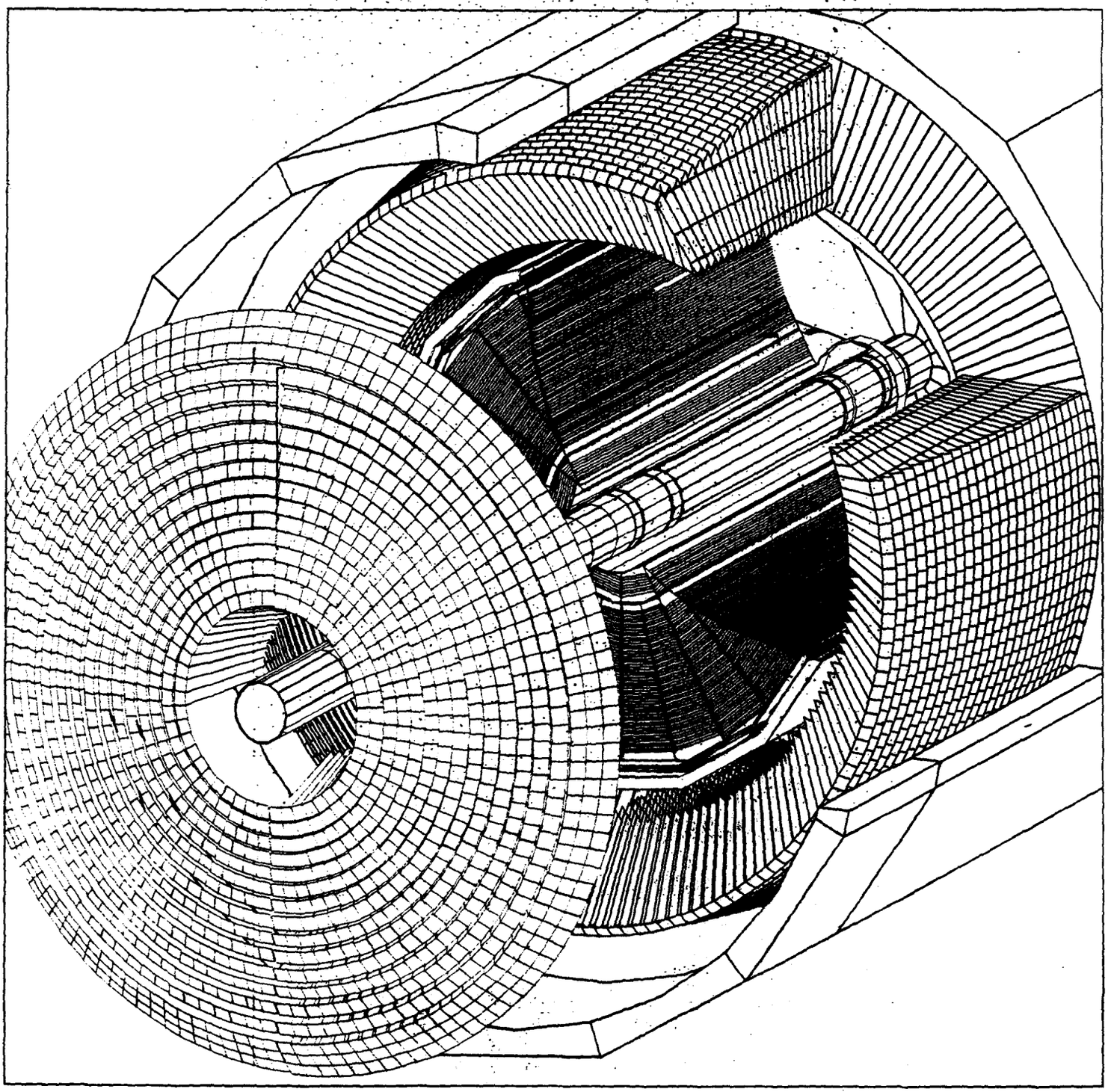

OPAL
GEANT3.15
Exploded view

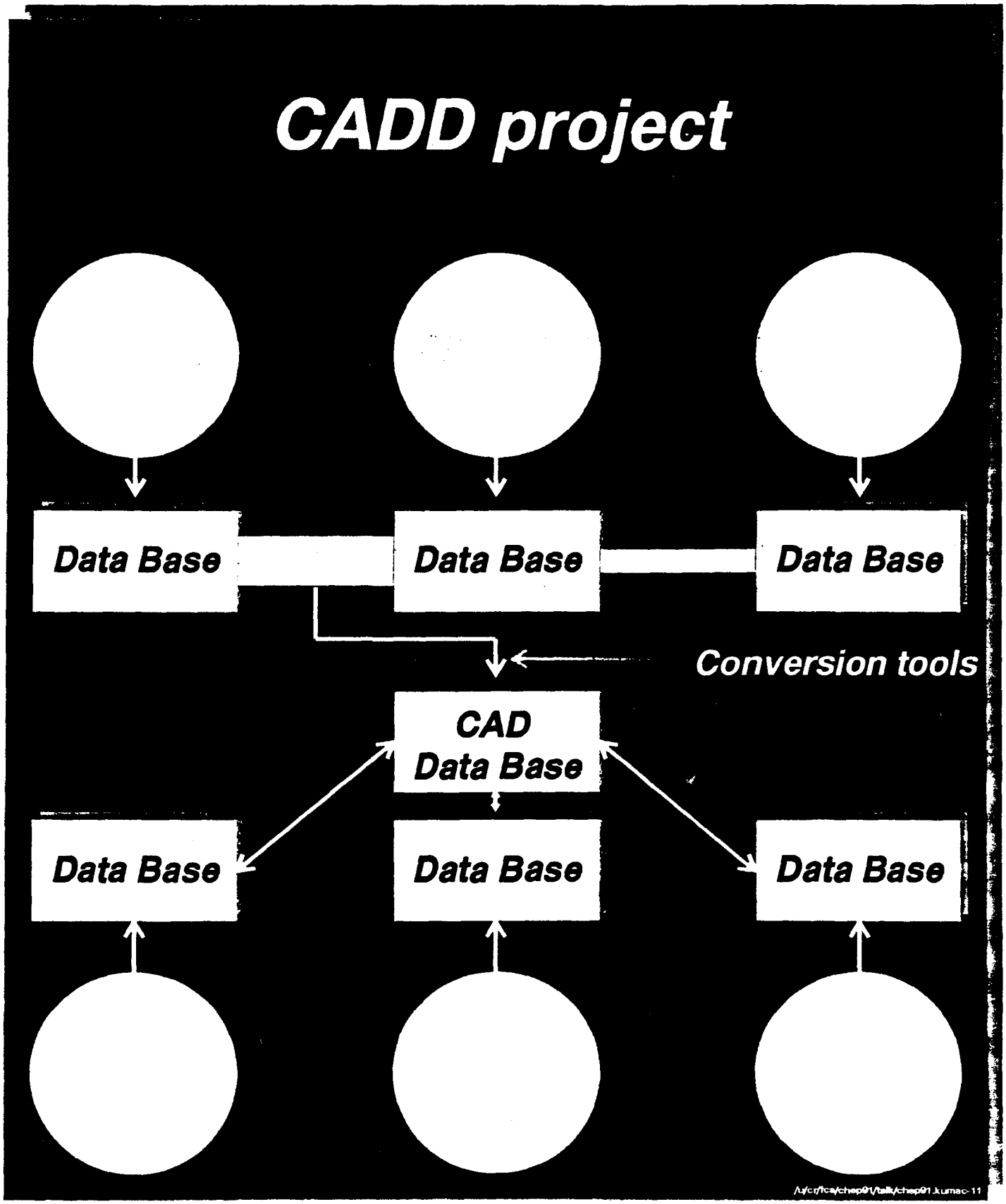
HIGZ_01 @ snake

EAGLE



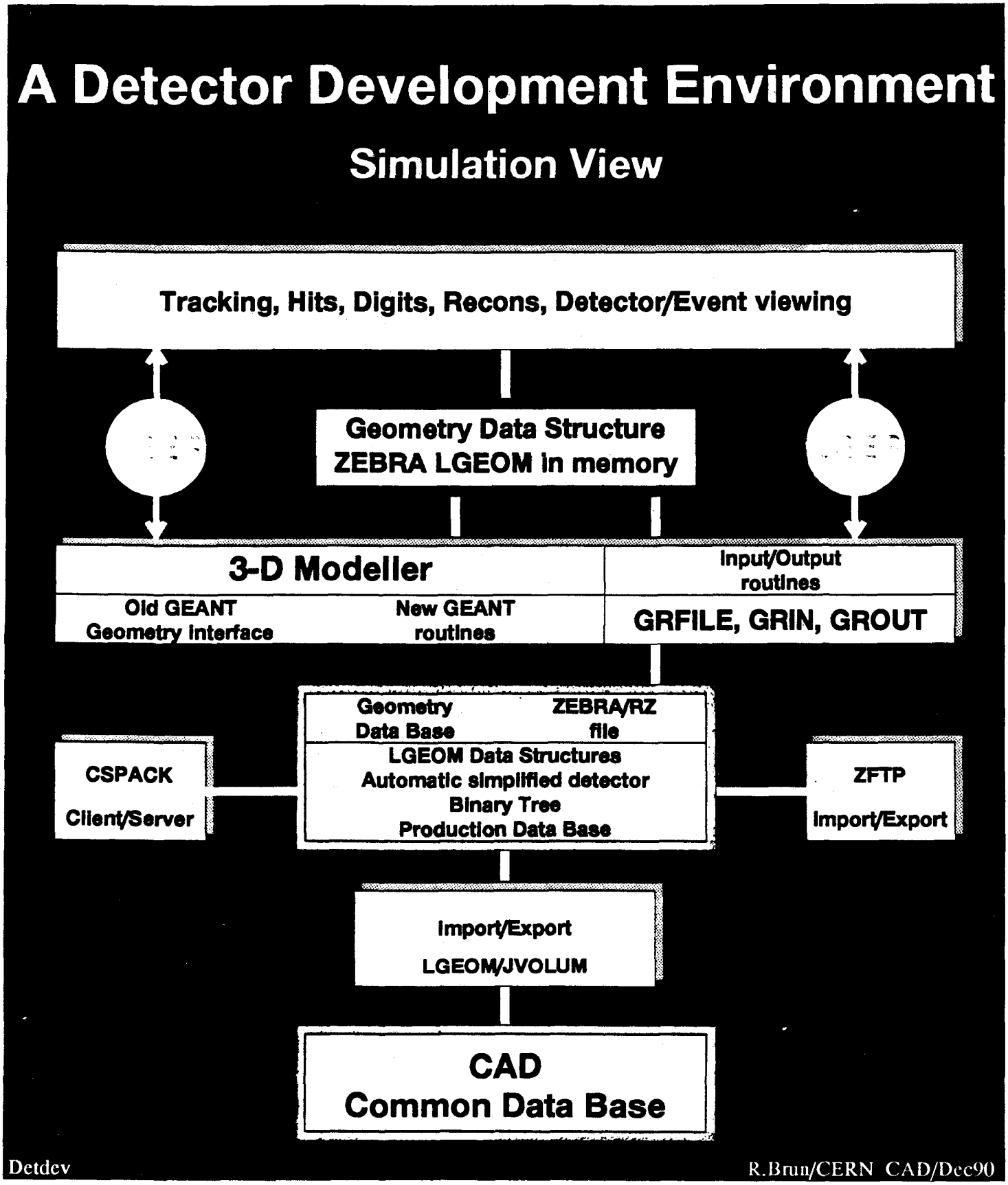
GEANT 3.15




CADD project
Data Base
Data Base
Data Base
Conversion tools
CAD
Data Base
Data Base
Data Base
Data Base

A Detector Development Environment

Simulation View



CAD info exchange

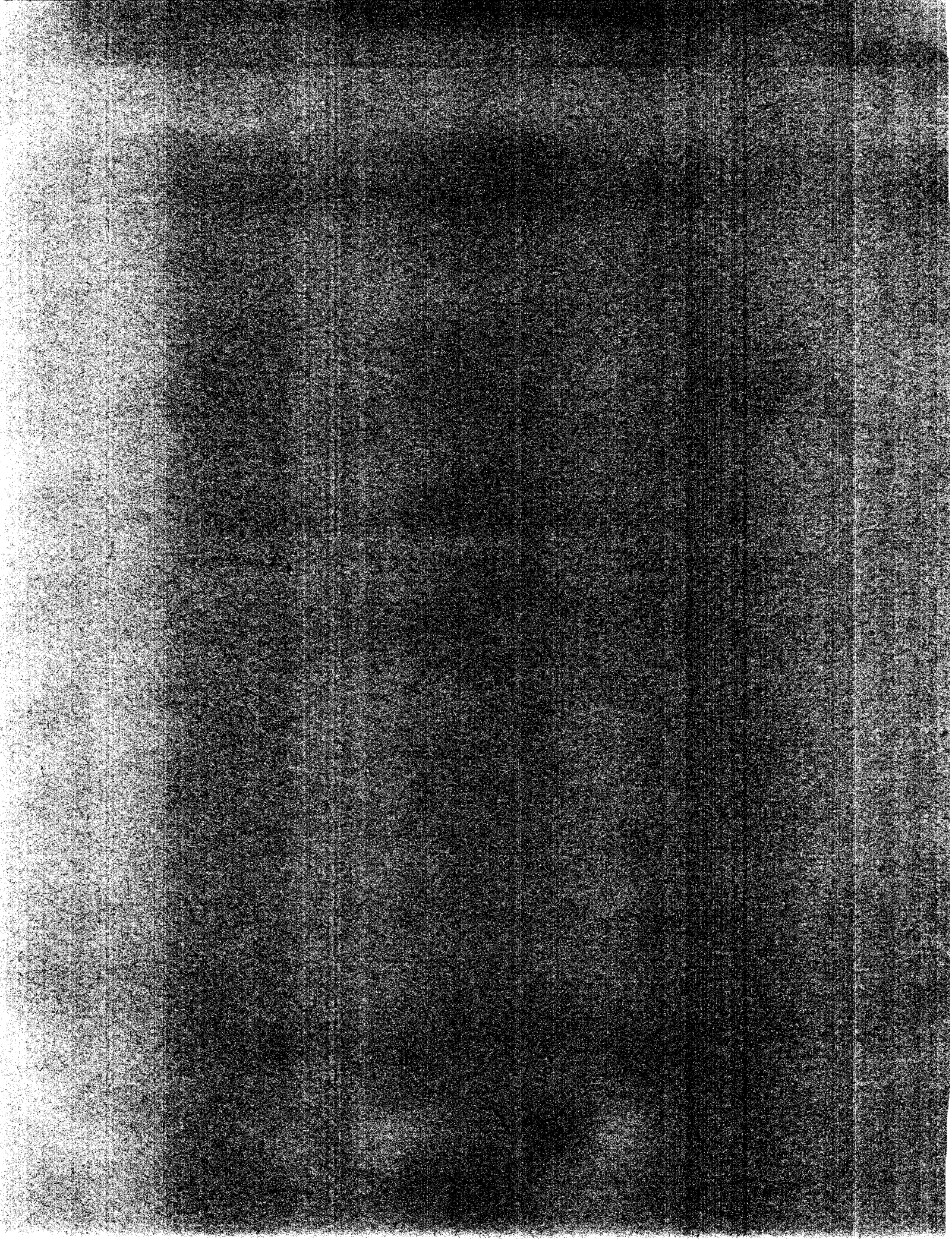
- NFF (Neutral File Formats)
- Very limited use of NFF today
- No international standard
- STEP due to come in 1992-1993
- SET French AFNOR standard
- Initial interface with SET started
- A flat file is really flat
- Syntactical and semantical problems{ the GEANT-CADD meta-objects

CADD project

- Improve communication between physicists and engineers
- Exchange detector design among CAD systems and GEANT
- Neutral File Format
- SET selected for the moment waiting for STEP
- Interface EUCLID-SET-GEANT in preparation
- New geometry will help

"The Possibilities for Using the Intel iPSC Supercomputer for On-line and Off-line Analysis"

P. Keener



Intermediate p_T detectors:

Input rates at 10^{32} luminosity

- 10^7 events/sec at the Level 1 trigger
- 10^5 events/sec at the Level 2 farm

The event size is estimated to be 0.5 Mbyte and the estimated processing time per event is 10 MIP-seconds

The estimated computing power needed in the farms is O(1 TeraOPS)

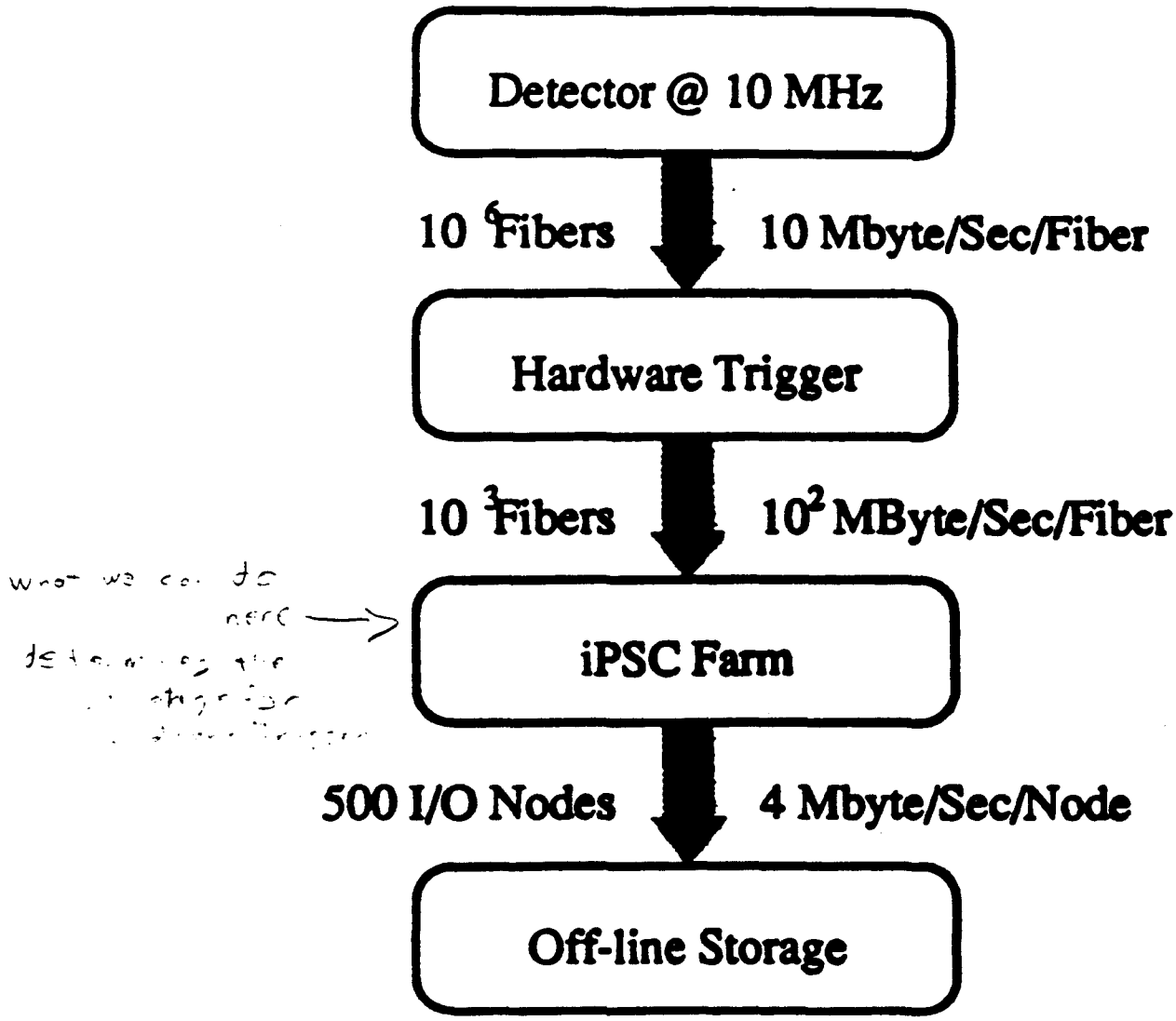


Figure 13: Data flow from detector to off-line storage.

Goals for processor farm R&D

- Determine the technical feasibility of a TeraOPS farm with 1-10,000 nodes by 1997
- Define requirements for interprocessor and processor/detector communications to handle 10-100 Gbyte/sec event rate to the farm
- Determine the trigger algorithms necessary to achieve the rejection rates which are estimated to be necessary
- Investigate the best implementation of algorithms to achieve the optimal physics throughput
  - memory lookups, cluster-finding, etc.
  - massively parallel processing schemes, e.g. neural networks

- Investigate the distribution of high-level processors within an overall trigger scheme

- Event Building - merging $\sim$ 1000 parallel data streams into a 1 Mbyte event at rates of 10 – 100 kHz

- Begin collaborations with industry on R&D for high-performance processor systems and parallel processor software for large-scale farms

- Provide a scalable resource which can be used for SSC physics and detector simulation in the near future and as part of a test-beam DAQ system later

Achievements in 1991

- Major portions of the CERN Program Library including GEANT, as well as ISAJET, PYTHIA, and JETSET have been ported to the Intel iPSC/860 Hypercube. We have provided CERN with these modifications.

- A parallel processor with 4 i860-nodes and 3 I/O-nodes, configured as a hypercube, has been purchased and used extensively.

- Simulations of hypercube I/O properties have been completed using Verilog.

- Preliminary study of random numbers. We note CERN RANF fails Knuth's criteria for the spectral test.

- We are preparing to use the Caltech Consortium MESH machine (512 i860 nodes with 40 MBytes/sec between nodes).

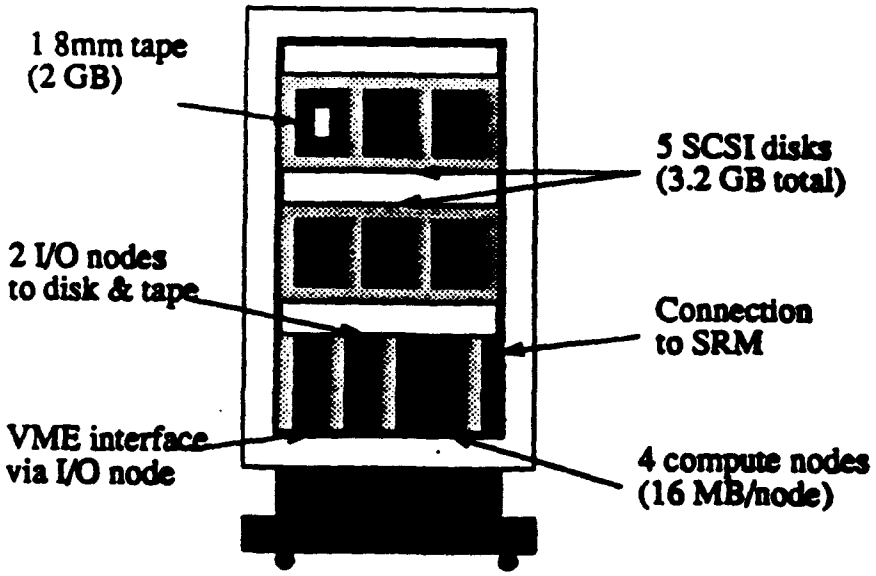


Figure 1: iPSC system configuration

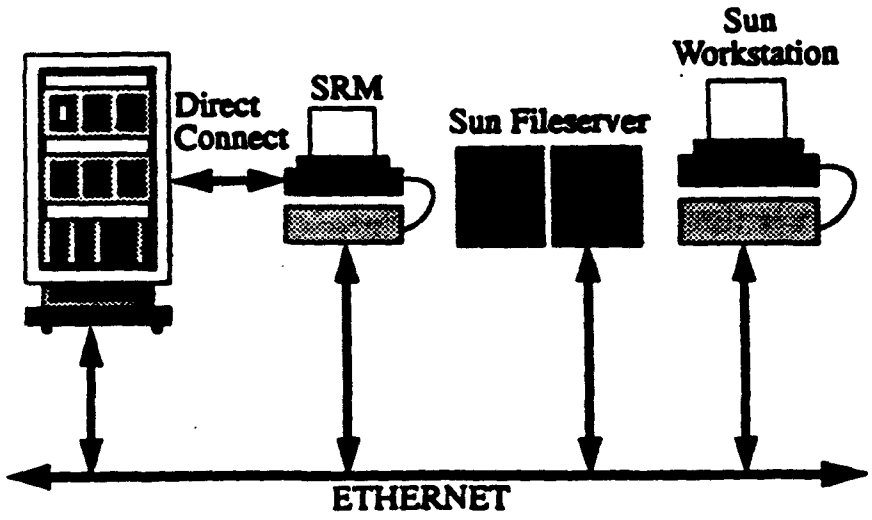


Figure 2: iPSC network configuration

SSCL Hypercube

The SSCL has purchased a 64-node iPSC/860 Hypercube for beam related studies. We have used this machine for the majority of the code ported by L. A. Roberts. We thank George Bourianoff and his group for use of the machine.

Porting Physics codes

Standard difficulties with porting physics codes include

- dependence on CERN libraries - much of the code in these libraries is non-portable. Portability is achieved by identifying a version of the code written for a machine that is similar in operation to that of the target system
- lack of complete standards in FORTRAN, eg. I/O and HEX constants
- CERN libraries for the DECstation still contain some MIPS assembly language routines
- fundamental differences between SYSV and BSD flavors of Unix
- different communication protocols for system services
- lack of sophisticated compilers for new machines

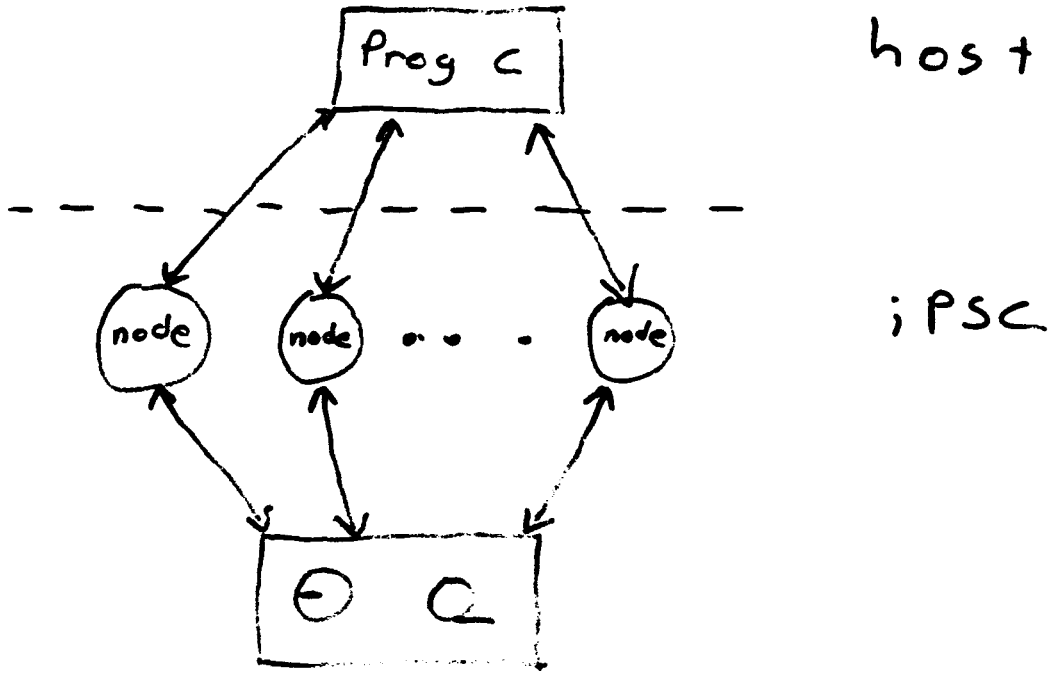
After solving these problems, the CERN libraries, ISAJET, JETSET, PYTHIA and BCD GEANT were successfully run on one node of the SSCL iPSC/860 platform and benchmarked

CPU	GEXAM1 Sec/Event	BCD GEANT Sec/Event	BCD ISAJET Sec/Event
IBM	11.72	236.89	0.05863
Intel	12.33	204.14	0.09836
Sun	11.40	175.11	0.06938
SGI	6.11	139.34	0.04663
HP	4.01	81.18	—

Table 1: Results of several physics benchmarks. The times quoted are CPU seconds

The configurations used for these benchmarks are

- IBM — IBM RS/6000-320, 16MB, AIX 3.1.5, XL FORTRAN 2.1
- Intel — iPSC/860, 8MB, FORTRAN compiler PGFTN Sun4/4.0 Rel 1.3a
- Sun — Sun SPARCstation 2, 16MB, SunOS 4.1.1, SunFORTRAN 1.4
- SGI — Silicon Graphics 4D/35TG, 16MB, IRIX 3.3.2
- HP — HP Apollo 9000/720, 64MB, HP-UX A.B8.05


Prog c
host
node
node
node
iPSC

Large Grain Parallel Code

Program Control

- controlling event generation
- Specify input/output files
- gathering & Presenting intermediate Statistics during the run
- collecting final run data

Input

- static data files
- command decks
- general user input

Output

- event stream
  - need to add info to the event record
- listing file

Random Numbers

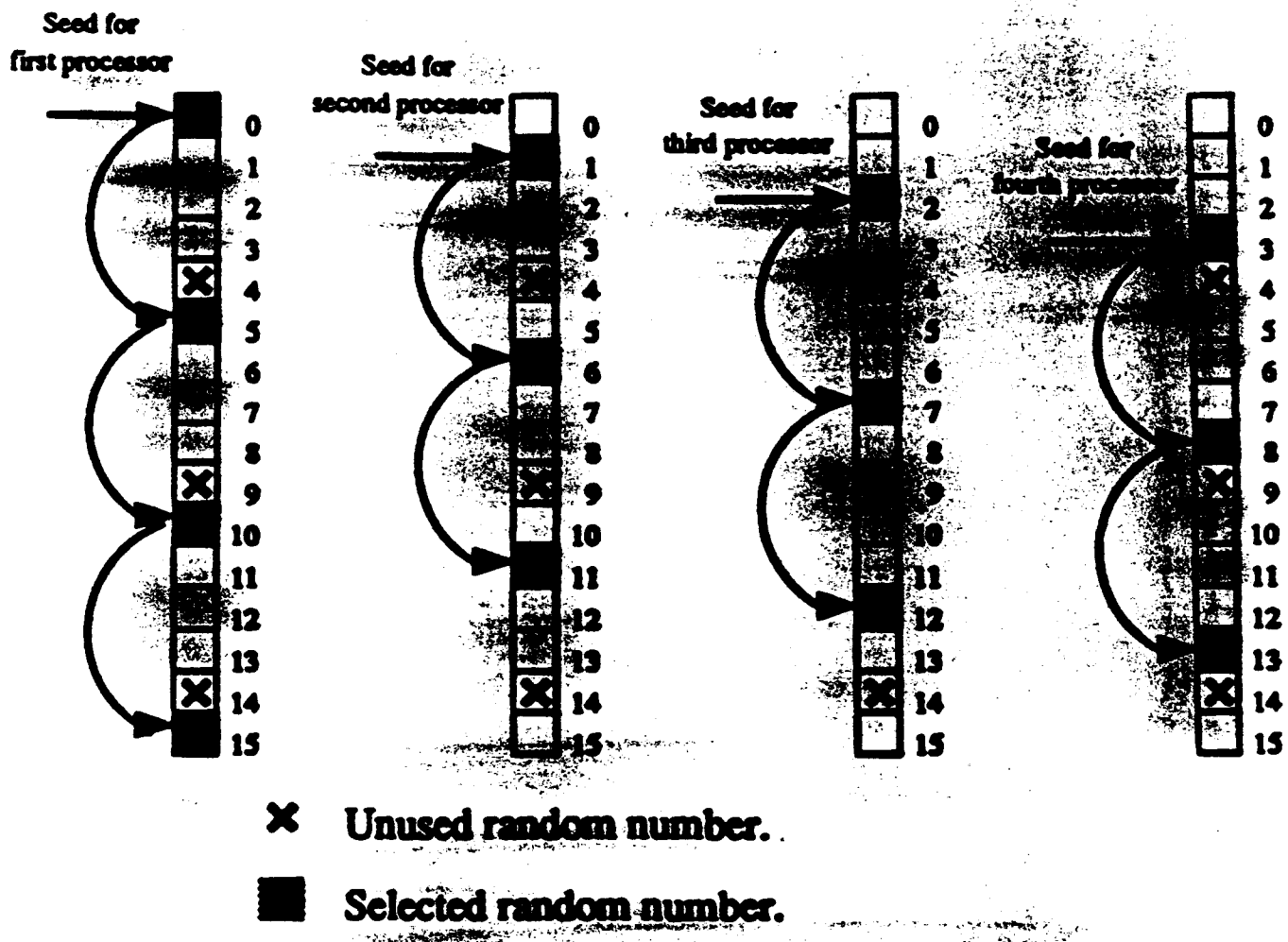


Figure 5: 'Leapfrog' algorithm (adapted from [17]).

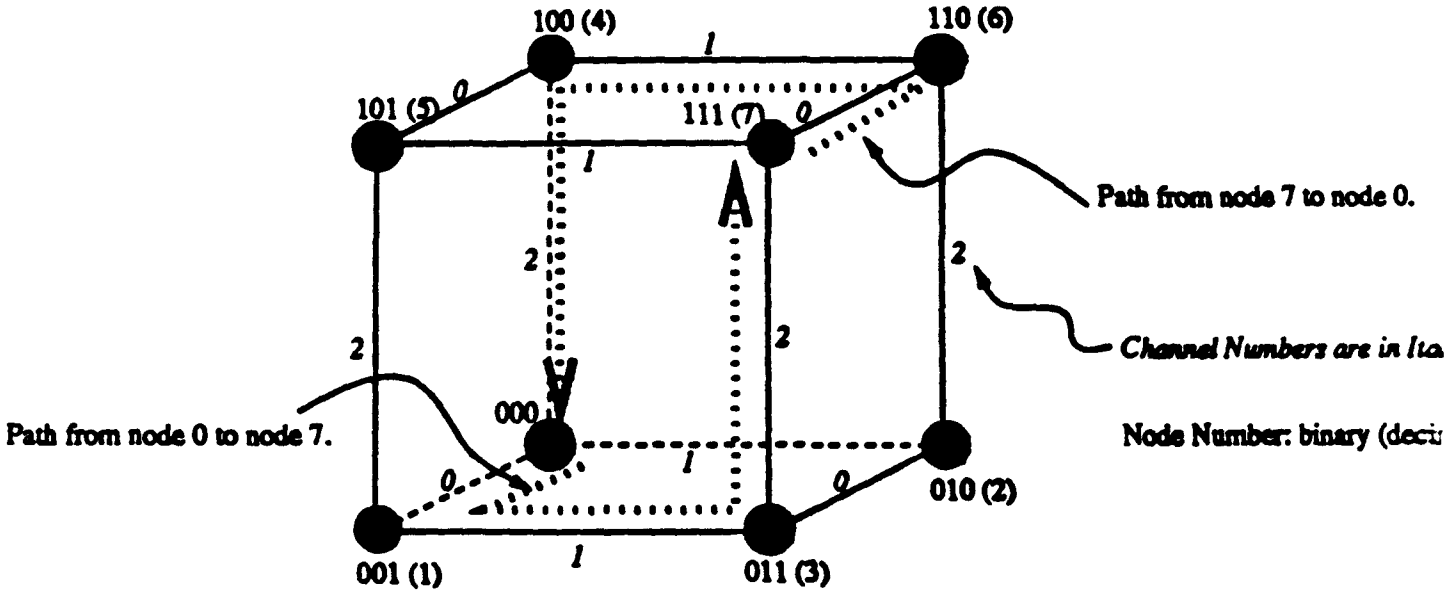


Figure 6: Paths of communication between nodes 0 and 7

6.2.2 Results of Simulation

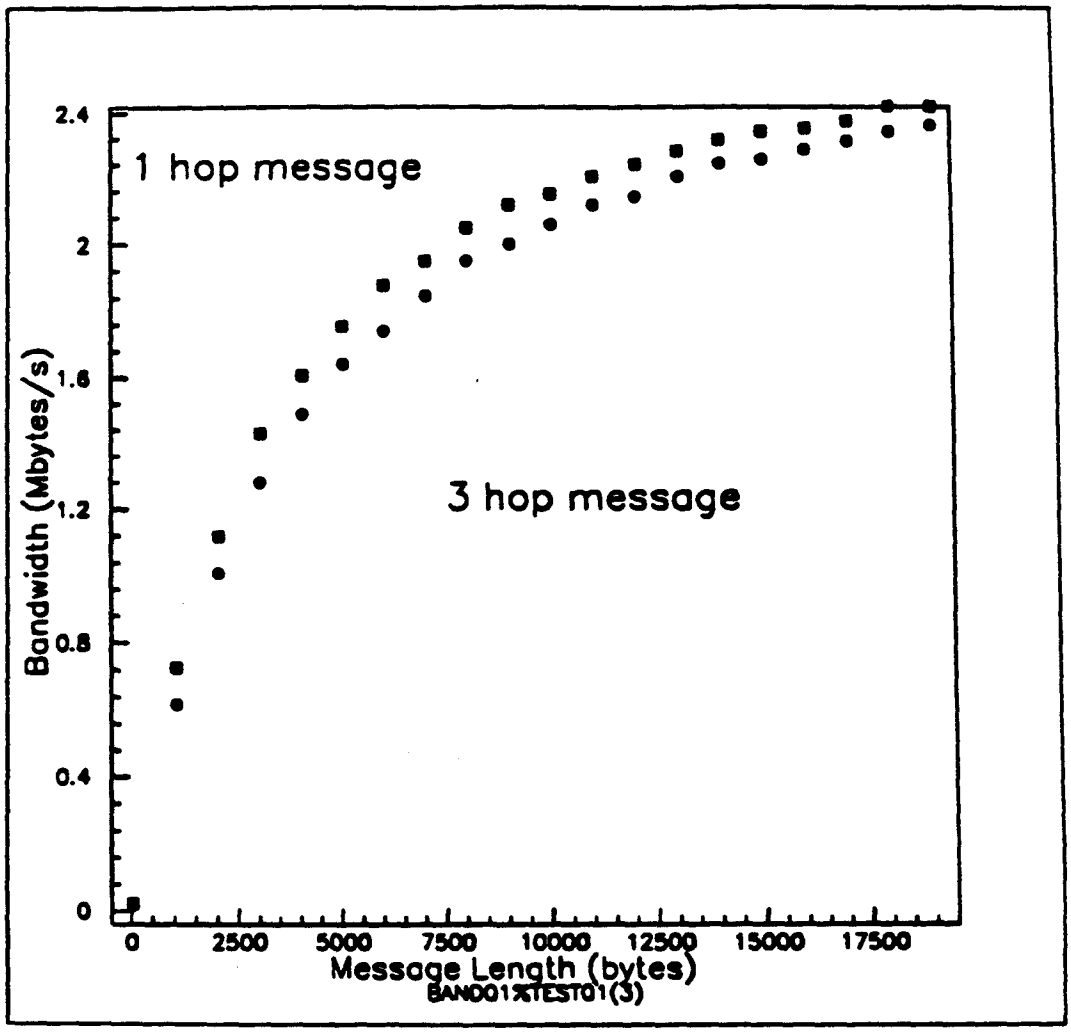


Figure 7: Message Length vs. Bandwidth

4 Node Combinations

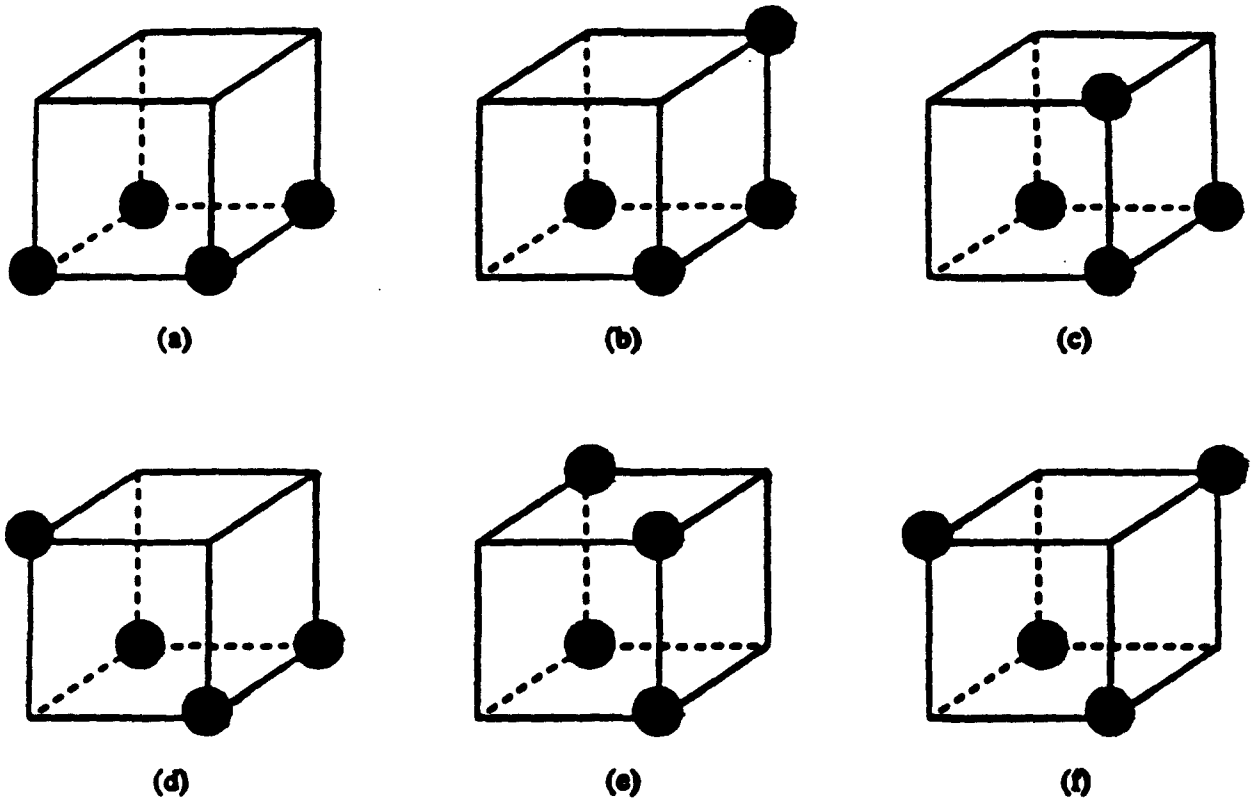


Figure 12: All possible 4 I/O node combinations.

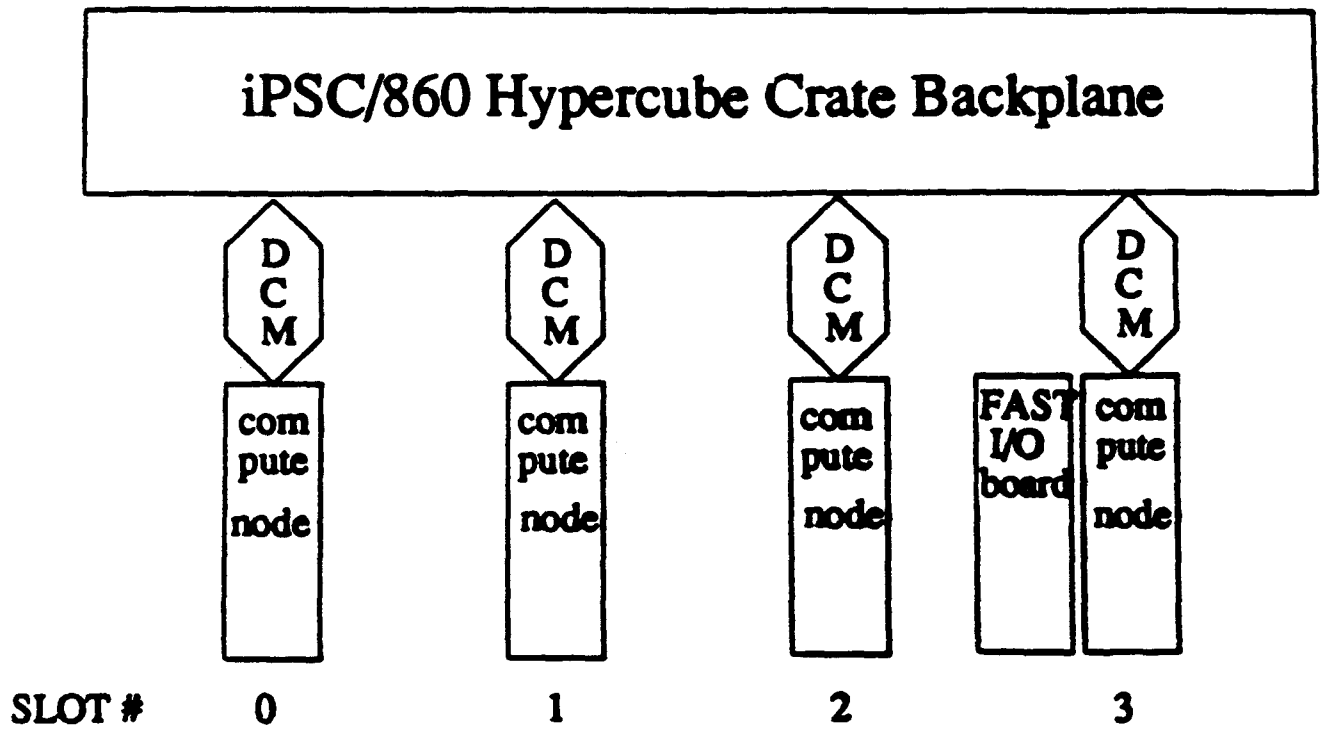


Figure 10: Distributed Model.

HYPER SIM

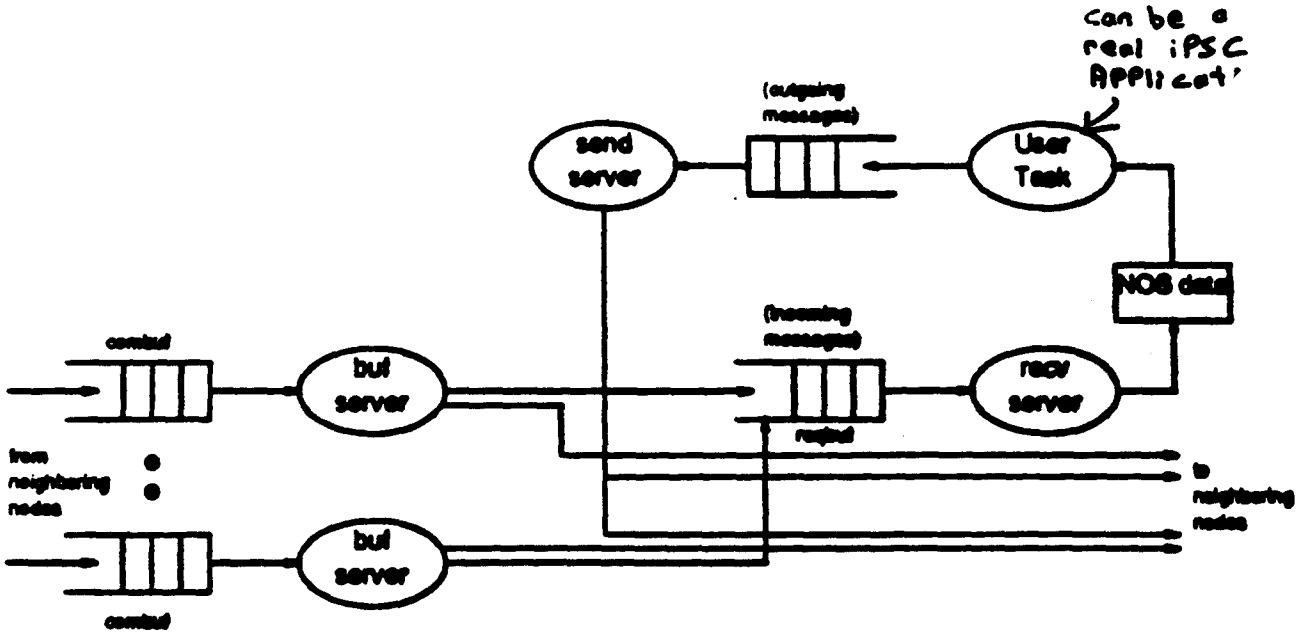


Figure 3: Message Flow Through a Simulated Node

Goals for 1992

- Continue a joint project with Intel to incorporate a FAST I/O Board into the Intel MESH simulation package — Hypersim.
- Demonstrate injection rate of the MESH using the FAST I/O Board.
- Measure time to build events vs. injection-rate on the MESH.
- Measure Level 3 trigger rejection times with concurrent algorithms running on the MESH.
- Continue to develop a version of **ISAJET** and **GEANT** that executes simultaneously on multiple-nodes.

Subsystem Renewal Proposal to SSC Laboratory for R&D of a Parallel Computing Farm

Penn-Princeton-SSCL-Intel Collaboration[1]

K. Anupindi, L. D. Gladney, P. T. Keener,
N. S. Lockyer and M. A. Rezaei
Department of Physics, David Rittenhouse Laboratories
University of Pennsylvania, Philadelphia, PA 10104

J. G. Heinrich and K. T. McDonald
Joseph Henry Laboratories, Princeton University
Princeton, NJ 08544

L. A. Roberts
Superconducting Super Collider Laboratory
Dallas, TX 75237

Justin Rattner
Intel Scientific Computers
Beaverton, OR 97006

[1]Contact Persons: L. D. Gladney and N. S. Lockyer